TEST-TIME RL ALIGNMENT EXPOSES TASK FAMILIARITY ARTIFACTS IN LLM BENCHMARKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct evaluation of LLMs on benchmarks can be misleading because different models may have acquired different degrees of familiarity with the task prior to evaluation. The train-before-test approach controls for task familiarity by giving each model task-relevant training before evaluation, originally through supervised fine-tuning(SFT). However, suitable training data is often hard to come by, and evaluation results vary with the data chosen. In this paper, we propose a two-stage test-time reinforcement learning (RL) alignment method for train-before-test. First, RL with a single sample provides a first alignment of the model to the task format, and second, test-time RL with majority-voting reward aligns the model to the benchmark distribution. Our test-time RL alignment method aligns similarly well as the SFT-based train-before-test, but without requiring a task-specific training set. On a domain-specific benchmark without training data, we show that direct evaluation underestimates base models, which perform substantially better once aligned, yielding a comparison of models under the same task preparation. Moreover, for reasoning tasks, base models reach performance comparable to many of their fine-tuned variants after the same label-free test-time alignment, while the fine-tuned variants themselves exhibit only marginal changes.

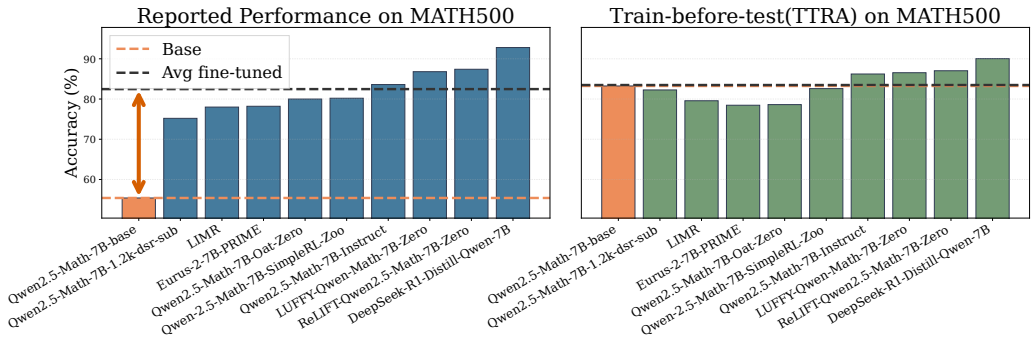

Figure 1: Fine-tuning gains diminish substantially under TTRA-based train-before-test. **Left:** The reported performance reveals substantial accuracy gaps between base models and their fine-tuned variants. **Right:** After applying the same label-free TTRA procedure to all models, the base model improves substantially, while many fine-tuned variants remain close to their reported performance, bringing both to comparable performance levels.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable performance across a diverse range of complex tasks. To measure improvements in model, training, and data, robust evaluation methodologies are critically important. Typically, performance is assessed on benchmarks such as MMLU for broad knowledge (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021) and MATH (Lightman et al., 2024) for mathematical reasoning, (Hager et al., 2024) for clinical decision making, and (Dominguez-Olmedo et al., 2025b) for legal classification tasks, to give some examples. However,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

model comparisons under direct benchmark evaluation may be confounded by differences in how familiar the models are with the task, as noted by Dominguez-Olmedo et al. (2025a).

This evaluation challenge is especially relevant for reasoning models such as OpenAI's o1 (OpenAI et al., 2024) and DeepSeek-R1 (Guo et al., 2025), which have attracted significant attention. Several works report substantial reasoning performance gains from post-training LLMs with supervised fine-tuning (SFT) and reinforcement learning with verifiable rewards (RLVR) (Wu et al., 2026; Liu et al., 2025). Recent work questions whether current fine-tuning methods genuinely incentivize reasoning ability (Yue et al., 2025; Hochlehnert et al., 2025). However, these analyses are often indirect and do not explicitly account for task familiarity.

To enable fair comparisons of models on benchmarks, Dominguez-Olmedo et al. (2025a) proposed the *train-before-test* approach, in which a model is given task-relevant training. Dominguez-Olmedo et al. (2025a); Zhang et al. (2026) demonstrate that SFT on a benchmark-specific training set before evaluation harmonizes model rankings and removes apparent performance gaps between newer and older models.

However, SFT-based train-before-test is limited by its dependency on the training dataset: For reasoning tasks, the dependence on the training dataset introduces substantial instability in evaluation. Applying supervised fine-tuning to the same base model with different reasoning-oriented training datasets can yield substantially different performance, making the resulting model comparisons highly sensitive to the choice of training data.

Moreover, curating suitable validation sets is often difficult. For many widely used reasoning benchmarks, such as AIME, no official training or validation set exists, forcing researchers to curate datasets from other sources. Even if an official training set is provided, it may not be in-distribution. An example is MMLU (Hendrycks et al., 2021), where the provided training data focuses on reading comprehension, while the test set evaluates general knowledge across various domains. Also, for domain-specific benchmarks, such as clinical decision-making (Hager et al., 2024), building a dataset for supervised train-before-test would require expensive annotations from domain experts.

In this paper, we propose a **dataset-free test-time reinforcement learning alignment (TTRA) method for train-before-test**. Our method updates the model, but only on the benchmark's tasks, with minimal supervision. It consists of a one-shot reinforcement learning step based on a single labeled example to provide the model with the necessary initial signal for task format alignment, and a second reinforcement learning step based on majority-voting reward on the benchmark to further adapt the model to the task distribution.

Our findings are as follows:

First, we study TTRA in a similar setup as SFT-based train-before-test (Dominguez-Olmedo et al., 2025a) to validate it. We find that it harmonizes model rankings across related benchmarks similar to SFT-based train-before-test, but with the added advantage of not using training data. We also demonstrate that it is stable on reasoning benchmarks and verify the consistency of the method on held-out data.

Second, we consider reasoning and demonstrate that TTRA reduces the performance differences between base models and their fine-tuned reasoning variants substantially (see Figure 1). After applying the same TTRA procedure, base models reach performance comparable to many RLVR/SFT trained variants, whose own scores change little under TTRA. In contrast, variants distilled from stronger models retain a clear advantage, consistent with Yue et al. (2025). We observe a similar pattern on code-generation benchmarks.

Third, we consider a domain-specific benchmark for clinical decision making, and find that direct evaluation substantially underestimates the performance of base models. Compared with direct evaluation, TTRA unlocks the model's performance and demonstrates that while scores are initially low due to task unfamiliarity, the model achieves accuracy comparable to human doctors after alignment, without requiring a separate training dataset.

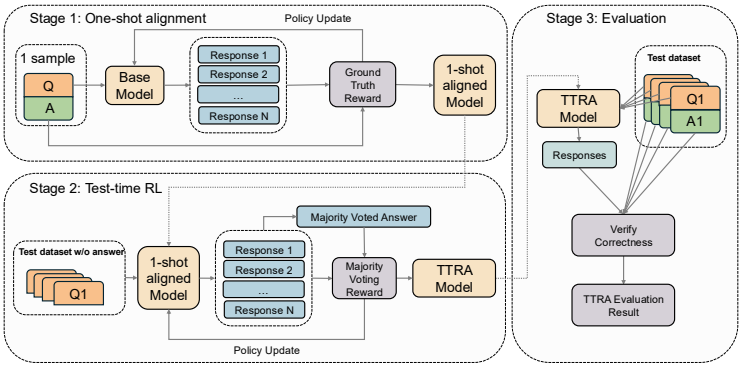

Figure 2: Overview of the TTRA-based train-before-test evaluation pipeline.

2 Test-time reinforcement learning alignment (TTRA) for train-before-test

To overcome the limitations of SFT-based train-before-test, we introduce a dataset-free train-before-test method. Our approach leverages insights from recent self-improvement methods that require no ground-truth labels and have demonstrated that LLMs possess an inherent potential for test-time adaptation. For instance, prior work has explored entropy minimization as a training signal (Cui et al., 2025b; Agarwal et al., 2025; Gao et al., 2025). Furthermore, recent studies have shown that test-time training can be effective for LLMs. Zuo et al. (2025) uses majority voting to generate pseudo-labels for test-time reinforcement learning (TTRL), while Hübotter et al. (2025) proposes test-time curriculum RL to improve performance.

However, unsupervised TTRL can fail if a model lacks a minimal level of task competence, as it may not generate meaningful responses from which to derive a useful reward signal. Conversely, one-shot reinforcement learning (Wang et al., 2025) can provide a crucial initial alignment signal but often causes the model to overfit to the limited example, failing to generalize across the entire test set and thus capping its potential performance. Therefore, we combine these into a two-stage method illustrated in Figure 2 and described below. In this work, we use Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as the policy optimization algorithm.

Stage 1: One-shot task format alignment In the initial stage, we randomly select a single labeled sample from the available data to perform one-shot reinforcement learning. The model learns the fundamental structure and format expectations of the task through this minimal supervision.

Stage 2: Test-time reinforcement learning After the initial alignment, the model possesses basic task understanding, enabling effective test-time adaptation. We employ TTRL using majority voting as the reward signal. This allows the model to further refine its performance across the entire test distribution. The second stage ensures comprehensive alignment with the test dataset while leveraging the model's own generations as learning signals, eliminating the need for additional labeled data.

Our method maintains the benefits of train-before-test while eliminating the dependency on a separate training dataset, requiring only a single labeled sample that is typically obtainable from benchmark.

3 Validation of TTRA-based train-before-test

In this section, we validate the effectiveness of TTRA. First, we study TTRA in a similar setup as the original train-before-test works (Dominguez-Olmedo et al., 2025a; Zhang et al., 2026) considered. We demonstrate that TTRA harmonizes model rankings across related benchmarks, achieving results similar to SFT-based train-before-test, but with the added advantage of not using training data. Furthermore, we show that our method exhibits better stability compared to SFT-based train-before-test on a reasoning benchmark. Finally, we verify the method's consistency on held-out test sets to address potential concerns regarding overfitting to the test set.

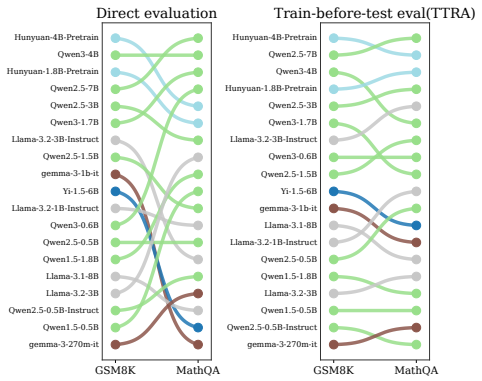

Figure 3: TTRA harmonizes model rankings between GSM8K and MathQA. **Left:** Direct evaluation shows highly discordant model rankings across the two benchmarks. **Right:** After TTRA alignment, the rankings become substantially more consistent and harmonized.

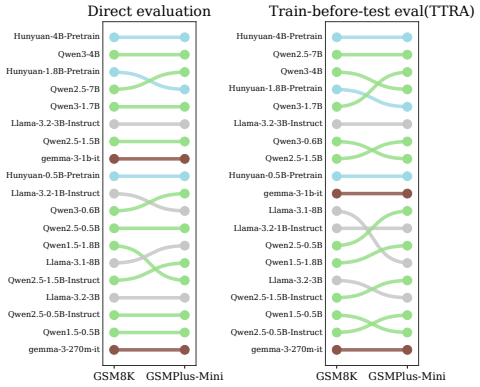

Figure 4: TTRA preserves existing ranking harmony between GSM8K and GSM-plus-mini. **Left:** Direct evaluation shows already consistent rankings for these similar-format tasks. **Right:** TTRA evaluation maintains this high consistency.

3.1 TTRA CAN HARMONIZE LLM RANKINGS

One would expect that relative comparisons among models are relatively consistent for different benchmarks that target a given domain. However, Zhang et al. (2026) revealed that model rankings often vary substantially across related tasks, and demonstrated that SFT-based train-before-test can harmonize these inconsistent rankings.

We demonstrate that TTRA achieves similar harmonization as SFT-based train-before-test, but without requiring training data. We evaluate a range of open-weight LLMs on three mathematical benchmarks: GSM8K (Cobbe et al., 2021), MathQA (Amini et al., 2019), and GSM-plus-mini (Li et al., 2024), with and without TTRA. The full list of evaluated models and training configurations is provided in Appendix A.1.

As Figure 3(left) shows, under direct evaluation, model rankings between GSM8K and MathQA are inconsistent. As Figure 3(right) shows, after TTRA train-before-test, the rankings become substantially more harmonized, with models maintaining more consistent ordinal positions across both benchmarks. Another expected property of a train-before-test method is that, if model rankings are already consistent, TTRA preserves this property. Figure 4 demonstrates that TTRA indeed preserves the already-consistent rankings for models evaluated on the GSM8K and GSM-plus-mini benchmarks.

3.2 STABILITY ANALYSIS OF TTRA AND SFT-BASED TRAIN-BEFORE-TEST

The reliability of SFT-based train-before-test for task alignment depends on the choice of training dataset, and evaluation scores can fluctuate significantly based on the alignment dataset. In this section, we compare the stability of TTRA and SFT-based train-before-test under controlled sources of randomness. For TTRA, we used three different random seeds to select the one-shot alignment sample, utilizing the remaining data for both TTRL and evaluation. To quantify sensitivity of SFT based train-before-test, we fine-tuned the Qwen2.5-Math-1.5B model using a fixed budget of 1,000 samples. We randomly subsampled from AM-DeepSeek-Distilled (Tian et al., 2025), Eurus-2-SFT (Cui et al., 2025a; Yuan et al., 2025), and OpenThoughts-3 (Guha et al., 2026), whereas the S1K-1.1 dataset (Muennighoff et al., 2025) is fully used as it contains exactly 1k samples. Full training details are provided in Appendix A.2.

As illustrated in Figure 5, TTRA is robust, maintaining consistent performance across different random choices of the single example used for the one-shot step. In contrast, SFT-based train-before-test is quite sensitive to the data source used for training, as somewhat expected. On MATH500, accuracy fluctuates by over 16% depending on the dataset used for SFT-based train-before-test.

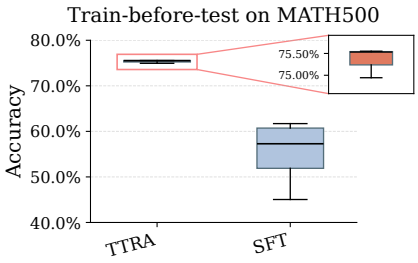

Figure 5: Stability analysis on MATH500. Our dataset-free TTRA method is robust to random choices of one-shot sample selection ($\Delta < 1\%$). In contrast, SFT-based train-before-test is highly sensitive to the training data used.

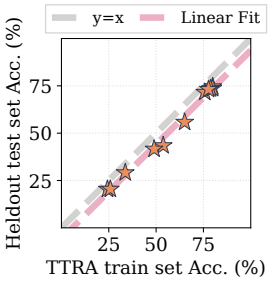

Figure 6: We compare accuracy on a strictly held-out test subset (y-axis) with accuracy on the subset used for TTRA training (x-axis) for each model. Performance is highly consistent across the two subsets, with the linear fit (pink dashed) lying only slightly below the $y = x$ diagonal (grey dashed).

Extended experiments showing similar variation for SFT on AIME and OlympiadBench are in Appendix A.2. This confirms that SFT-based train-before-test is sensitive to the training data, whereas TTRA avoids this issue.

3.3 Consistency study under a separate test set

The TTRL-alignment step only utilizes the question of the benchmark data, but not their labels. In principle, the observed performance gains of our method could therefore arise from robust task alignment (as desired) but also from potential overfitting to the benchmark, despite the fact that TTRA does not use test labels during training. To study whether this occurs, we performed a split-half validation by randomly partitioning the test-data into two equally-sized subsets. The first half was used to update the model via TTRA, while the second served as a strictly held-out test set for evaluation. As shown in Figure 6, evaluations of models on the train and test split are essentially perfectly correlated and are very close to the $y = x$ diagonal. We further find that the performance gap between base and fine-tuned models remains small under cross-benchmark TTRA, where alignment is performed on a different benchmark from the one used for evaluation (Appendix A.4). These experiments support the validity of TTRA-based train-before-test for evaluation and mitigate the overfitting concern.

4 Re-evaluating fine-tuning gains under TTRA-based train-before-test

Recently, many LLM fine-tuning recipes (including methods and data) have been proposed, often reporting substantial performance gains and attributing them to improved reasoning capabilities. It is important to understand whether these gains persist after models receive the same task-specific preparation, particularly when such preparation requires only minimal supervision.

Some prior work has examined whether RLVR enhances math reasoning capabilities. For example, Yue et al. (2025) argues that the gains observed from RLVR may not reflect stronger underlying reasoning, but instead stem from improved sampling behavior, and that RLVR does not introduce new reasoning patterns. Similarly, Hochlehnert et al. (2025) finds that for models already fine-tuned on math reasoning tasks, such as DeepSeek-R1-Distill, additional RLVR yields little benefit, while base models show substantial improvements. However, this comparison may be confounded by differences in task familiarity, as base models may perform poorly due to limited exposure under the reasoning task rather than insufficient reasoning capability.

To address this, we evaluate several base models and their fine-tuned variants on math reasoning benchmarks. Before evaluation, we subject all models to the same TTRA procedure, providing task-specific preparation with minimal supervision. We then examine whether the performance gaps between base and fine-tuned models persist.

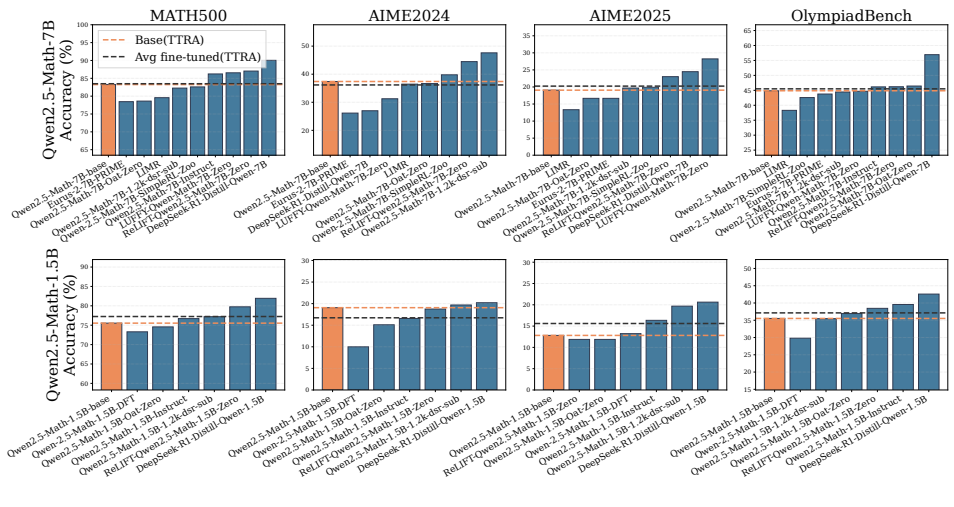

Figure 7: Performance comparison of Qwen2.5-Math base and fine-tuned models under TTRA evaluation. Results are shown for 7B models (top) and 1.5B models (bottom) across four benchmarks. The orange bars denote the base models, while blue bars denote various fine-tuned variants. The orange dashed line marks the base model's performance level, highlighting that the performance advantage of fine-tuned models largely disappears after applying TTRA.

4.1 Exposing spurious fine-tuning gains on math benchmarks

We evaluated Qwen2.5-Math-1.5B/7B base models (Yang et al., 2024b) and a range of variants fine-tuned for reasoning tasks. We include variants derived from Qwen2.5-Math-Instruct (Yang et al., 2024b), ReLIFT (Ma et al., 2026), Oat-Zero (Liu et al., 2025), DeepSeek-R1-Distill (Guo et al., 2025), as well as Qwen2.5-Math-1.2k-dsr-sub, a model trained with a 1.2k DeepScalerR subset (Wang et al., 2025; Luo et al., 2025). At the 1.5B scale, we additionally include a model fine-tuned using the dynamic fine-tuning (DFT) method (Wu et al., 2026). At the 7B scale, we further incorporate several reasoning variants, including PRIME (Cui et al., 2025a; Yuan et al., 2025), SimpleRL-Zoo (Zeng et al., 2025b), LIMR (Li et al., 2025), and LUFFY (Yan et al., 2025).

We evaluate these models on mathematical reasoning benchmarks, including MATH500 (Lightman et al., 2024), AIME 2024 & 2025, and OlympiadBench (He et al., 2024). For MATH500, we randomly select a sample from MATH500 with a difficulty level of 1 for the 1-shot alignment step. For AIME, we perform the initial 1-shot alignment step on a randomly chosen AIME2023 question and the second TTRL-alignment on the combination of AIME2024&2025 since the AIME dataset only contains 30 questions each. Further training details and hyperparameters are in Appendix B.1.

Fine-tuning gains may be confounded by task familiarity The results for Qwen2.5-Math-7B are shown in Figure 1. Figure 1 (left) presents the performance reported in the original model papers, where the scores for the base model and Qwen2.5-Math-7B-Instruct are taken from (Liu et al., 2025). For direct evaluation, the fine-tuned variants appear to be significantly better than base models, with an average performance gain of $\Delta = 27.07\%$. However, after applying TTRA to control for task familiarity the performance gap virtually vanishes (see Figure 1 (right)). This suggests that most of the gain observed under direct evaluation does not persist after TTRA.

In Figure 7, we extend this evaluation to two model scales and four reasoning benchmarks. Consistent with the findings in Figure 1, the average performance gap between base models and fine-tuned variants is small under train-before-test with TTRA. Remarkably, the aligned base model not only matches the average but even outperforms several fine-tuned variants. However, we observe models like DeepSeek-R1-Distill, which maintain a consistent performance advantage even after TTRA alignment. This suggests that gains from distillation with strong teacher models persist even after all models receive the same task-specific preparation. We also evaluated additional Llama models and observed a similar pattern: performance gains diminish substantially after train-before-test. The results are reported in Appendix B.1, Table 4.

After TTRA, base models improve substantially while most fine-tuned variants change only marginally, and the two end up at similar performance levels. This convergence is particularly noteworthy given that TTRA relies on minimal supervision, deriving its training signal primarily from the model's own outputs, whereas the fine-tuned variants were trained with substantially more external supervision. The main exception is DeepSeek-R1-Distill, which is distilled from the much stronger DeepSeek-R1 and retains a clear advantage even after TTRA. Both observations are consistent with the hypothesis of Yue et al. (2025) that RLVR mainly improves the sampling efficiency of reasoning paths already present in the base model, whereas distillation from a stronger teacher can introduce new reasoning patterns.

4.2 Exposing spurious fine-tuning gains on code-generation benchmarks

Next, we apply TTRA to HumanEval (Chen et al., 2021), a code-generation task. In the one-shot training stage, we randomly sample one entry with ground-truth input-output cases to train the model. In the TTRL stage, we use DeepSeek-v4-flash (DeepSeek-AI, 2026) to generate a valid test input for each problem, i.e., only the input arguments to the target function rather than an input-output pair. We then run each model-generated code solution on this same input and collect its output. Majority voting is applied to these outputs to obtain a pseudo-label, which is then used to compute the reward during training. This procedure does not require ground-truth outputs or correctness verification. Detailed steps are provided in Appendix B.2.

We evaluate Qwen2.5-Coder-1.5B/7B (Hui et al., 2024) as well as their instruction-finetuned versions on HumanEval (Chen et al., 2021). For Qwen2.5-Coder-7B, we consider an additional fine-tuned model, TIGER-AceCoder-Qwen2.5-Coder-7B-Ins-RM (Zeng et al., 2025a), hereafter referred to as AceCoder.

As shown in Table 1, instruction-tuned models substantially outperform their base counterparts under direct evaluation, but this gap narrows considerably after TTRA. This suggests that the gains from instruction tuning on code-generation benchmarks are similarly confounded by task familiarity.

Table 1: pass@1 (%) on HumanEval under direct and TTRA evaluation. For Qwen2.5-Coder-7B, "Avg. Gap" is the gap between the average score of two fine-tuned variants and base model.

	Qwen2.5-Coder-1.5B			Qwen2.5-Coder-7B			
	Base	Instruct	Gap	Base	Instruct	AceCoder	Avg. Gap
Direct	3.7±1.4	70.7±1.6	+67.0	40.1±3.4	87.8±1.0	88.5±1.3	+48.1
TTRA	70.2±1.7	73.4±1.9	+3.2	82.9±1.4	88.7±0.9	89.3±0.8	+6.1

5 Train-before-test reveals underestimated base model capabilities

Many important real-world applications of LLMs arise in domain-specific settings, where task definitions and input formats can differ substantially from those in commonly used benchmarks. The community has developed a growing number of domain-specific benchmarks. However, evaluating models directly on such benchmarks can systematically underestimate their capabilities, because the model is unfamiliar with the task.

We hypothesize that when LLMs are evaluated on domain-specific benchmarks without task alignment, their performance can be systematically underestimated, reflecting unfamiliarity with the benchmark rather than a lack of capability. SFT-based train-before-test is often not readily applicable for a lack of training sets, and constructing such sets is expensive and requires expert annotations. In this section, we use TTRA to evaluate several base models on clinical decision-making and legal tasks.

5.1 Case study 1: clinical decision making

The MIMIC-CDM benchmark (Hager et al., 2024) evaluates models on complex clinical cases in which the model must infer an underlying pathology or diagnosis from structured evidence, including

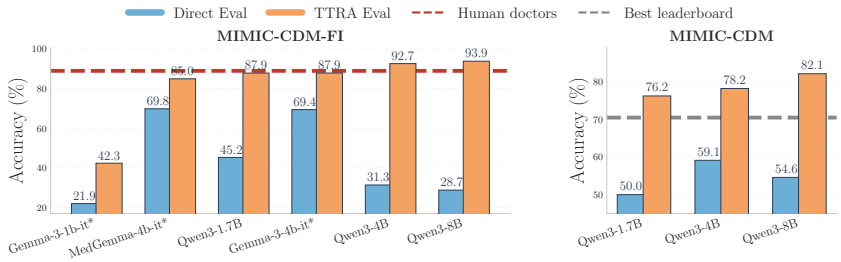

Figure 8: Evaluation of models aligned via TTRA compared to a directly evaluated baseline on MIMIC-CDM-FI (left) and MIMIC-CDM (right). For models marked with (*), base model performance is taken from (Hager et al., 2024). The human doctor baseline (left) and the best leaderboard entry (right, Gemma-3-27b-it) are both reported in (Hager et al., 2024).

patient history, physical examination findings, laboratory test results, and imaging reports. This benchmark has two settings: a full-information setting, where all available clinical information is presented in a single prompt and the model directly predicts the diagnosis, and a conversational setting, where only the patient history is initially revealed and the model must iteratively request additional information (e.g., examination findings, labs, imaging) before making a diagnostic decision.

We consider both the conversational(MIMIC-CDM) and full-information (MIMIC-CDM-FI) settings of the benchmark. To manage context length effectively, we use the Qwen3 tokenizer for token counting. For specific sections, such as imaging reports, that exceed 1,000 tokens in length, we utilize GPT5-nano (Singh et al., 2025) to generate a concise summary of the content before feeding it into the model. All prompts used are in Appendix C.1.

Figure 8 (left) shows the clinical decision-making accuracy of several base models before and after alignment with TTRA. In the full-information setting, the relative differences between models under direct evaluation change substantially after TTRA alignment. Under direct evaluation, the two Gemma 4B models (69.8% and 69.4%) appear strongest, while Qwen3-8B reaches only 28.7%. After TTRA, this ordering reverses: Qwen3-8B becomes the best at 93.9%, consistent with their larger pretraining scale. Under direct evaluation, all models fall well below the human-doctor baseline of Hager et al. (2024); after TTRA, models such as Qwen3-8B surpass it. Direct evaluation thus substantially underestimates how close current LLMs are to human-doctor performance on this task.

Figure 8 (right) shows results in the conversational setting. Under direct evaluation, the ranking does not follow model size: Qwen3-4B outperforms Qwen3-8B, and Qwen3-1.7B is close to Qwen3-8B. After TTRA, the ranking aligns with model size, as would be expected, with Qwen3-8B reaching 82.1%.

Moreover, Hager et al. (2024) evaluate a range of larger models, including Meta-Llama-3-70B-Instruct and DeepSeek-R1-Distill-Llama-70B, and the best model, Gemma-3-27b-it, obtains only 70.5% under direct evaluation. After TTRA, all three of the considerably smaller Qwen3 models surpass this best model. This suggests that leaderboard rankings on this benchmark partly reflect task familiarity. These results indicate that train-before-test alignment can substantially alter the apparent performance of base models in clinical decision-making tasks, revealing capabilities that are underestimated under direct evaluation.

5.2 Case study 2: legal tasks with rubric-based evaluation

We also consider a legal analysis task under rubric-based evaluation, to demonstrate that our approach is also applicable to rubric-based evaluation. We consider LEXAM (Fan et al., 2026), a legal reasoning benchmark based on law school exams. In this experiment, we focus only on its English open-ended questions and use DeepSeek-v4-flash as the judge. The judge is given the question, the model response, and an expert-written reference answer with normative grading guidance. It then scores the response against the rubrics and assigns a correctness score ranging from 0 to 1.

As shown in Table 2, TTRA alters the relative ranking of models (e.g., Qwen2.5-1.5B-Instruct moves above Qwen3-1.7B after alignment). This suggests that task familiarity confounds also open-ended

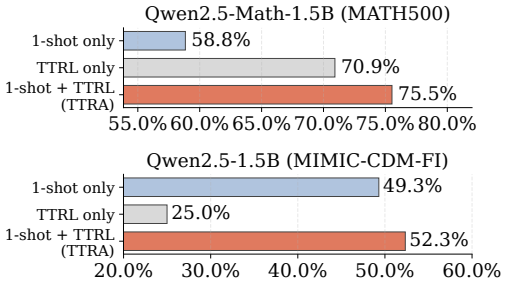

Figure 9: Stage-wise ablation study of TTRA. **Top:** For Qwen2.5-Math-1.5B on the MATH500 benchmark, both 1-shot alignment and TTRL contribute positively, with the hybrid TTRA approach achieving the highest accuracy. **Bottom:** For Qwen2.5-1.5B on the MIMIC-CDM-FI benchmark, removing 1-shot alignment causes TTRL to fail, highlighting the critical role of initial alignment in unfamiliar domains.

Table 2: Rubric-based evaluation on LEXAM English open-ended questions.

Model / Evaluation	Score
Qwen2.5-1.5B-Instruct	
Direct	0.124
TTRA	0.216
Qwen3-1.7B	
Direct	0.164
TTRA	0.208
Qwen3-4B	
Direct	0.266
TTRA	0.344

rubric-based evaluation, and controlling for it via TTRA yields a more faithful relative comparison between models.

6 Discussion and Limitations

In this section, we discuss the stage-wise effectiveness of TTRA, compare the compute costs of different evaluation methods, and discuss limitations of our method.

6.1 Stage-wise ablation study of TTRA

To validate the necessity of the two steps in our two-step framework, we conducted an ablation study to decouple the contributions of 1-shot alignment and TTRL. As shown in Figure 9, the two stages are complementary: 1-shot alignment provides the initial task-format signal needed to make optimization meaningful, while TTRL further adapts the model to the benchmark distribution. On MATH500 with Qwen2.5-Math-1.5B, both stages improve performance, and their combination achieves the best result.

The MIMIC-CDM-FI results further highlight the importance of the initial alignment step. While 1-shot alignment adapts the model to the complex clinical format, TTRL alone fails, reaching only 25.0% accuracy due to severe reward hacking: without the 1-shot signal, optimization collapses to predicting appendicitis in 99.93% of responses. These results suggest that TTRA benefits from both stages, especially in unfamiliar or domain-specific benchmarks where majority-vote rewards may otherwise be unreliable.

6.2 Computational cost

We compare direct evaluation, SFT-based train-before-test, and TTRA on GSM8K using Qwen3-0.6B. Direct evaluation samples 32 responses per question, TTRA uses one GSM8K training example plus test-time RL, and the SFT baseline fine-tunes on MetaMath (Yu et al., 2024) and Orca-Math (Mitra et al., 2024), following prior train-before-test work (Dominguez-Olmedo et al., 2025a). As shown in Appendix D.2 Table 5, SFT costs about $32\times$ direct evaluation, whereas TTRA costs only $6.6\times$ while providing similar alignment benefits.

6.3 Limitation

TTRA requires updating the model for evaluation, so it introduces additional computational overhead compared with direct evaluation, which only performs inference. In addition, our method relies on access to model weights in order to perform policy optimization. As a result, TTRA cannot be directly applied to closed-weight models, limiting its use for evaluating models available only through APIs.

Reproducibility Statement

We describe TTRA in Section 2 and the experimental setups in Sections 3–6. Implementation details, hyperparameters, and prompts for all experiments are provided in Appendices A.1–D.2. We also provide anonymized code in the supplementary material.

AI Use Statement

We used AI assistants to revise and polish the manuscript and to assist with code development and debugging. As part of our experiments, we also used LLMs to generate test inputs for HumanEval, summarize long clinical reports, and serve as the judge for LEXAM (Sections 4 and 5). All AI-assisted content and outputs were reviewed by the authors, who take full responsibility for the paper.

References

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in LLM reasoning. In *Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=UfFTBEsLgI.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 2357–2367. Association for Computational Linguistics, June 2019. doi: 10.18653/v1/N19-1245. URL https://aclanthology.org/N19-1245/.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025a. URL https://arxiv.org/abs/2502.01456.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025b. URL https://arxiv.org/abs/2505.22617.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Ricardo Dominguez-Olmedo, Florian E. Dorner, and Moritz Hardt. Training on the test task confounds evaluation and emergence. In *International Conference on Learning Representations (ICLR)*, 2025a. URL https://openreview.net/forum?id=jOmk0uS1hl.

Ricardo Dominguez-Olmedo, Vedant Nanda, Rediet Abebe, Stefan Bechtold, Christoph Engel, Jens Frankenreiter, Krishna P. Gummadi, Moritz Hardt, and Michael Livermore. Lawma: The power of specialization for legal annotation. In *International Conference on Learning Representations (ICLR)*, 2025b. URL https://openreview.net/forum?id=7El7K1DoyX.

Yu Fan, Jingwei Ni, Jakob Merane, Yang Tian, Yoan Hermstrüwer, Yinya Huang, Mubashara Akhtar, Etienne Salimbeni, Florian Geering, Oliver Dreyer, Daniel Brunner, Markus Leippold, Mrinmaya Sachan, Alexander Stremitzer, Christoph Engel, Elliott Ash, and Joel Niklaus. LEXam: Benchmarking legal reasoning on 340 law exams. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=xNhbMyXsJn.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

Zitian Gao, Lynx Chen, Haoming Luo, Joey Zhou, and Bryan Dai. One-shot entropy minimization, 2025. URL https://arxiv.org/abs/2505.20282.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit

Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Etash Kumar Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Rea Sprague, Ashima Suvarna, Benjamin Feuer, Leon Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su, Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah M Pratt, Vivek Ramanujan, Jon Saad-Falcon, Stutee Acharya, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alex Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for

reasoning models. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=7xjoTuaNmN.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Paul Hager, Friederike Jungmann, Robbie Holland, Kunal Bhagat, Inga Hubrecht, Manuel Knauer, Jakob Vielhauer, Marcus Makowski, Rickmer Braren, Georgios Kaissis, and Daniel Rueckert. Evaluation and mitigation of the limitations of large language models in clinical decision-making. *Nature Medicine*, 30(9):2613–2622, September 2024. ISSN 1546-170X. doi: 10.1038/s41591-024-03097-1. URL https://www.nature.com/articles/s41591-024-03097-1.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 3828–3850. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A Sober Look at Progress in Language Model Reasoning: Pitfalls and Paths to Reproducibility. In *Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=90UrTTxp5O.

Jonas Hübotter, Leander Diaz-Bone, Ido Hakimi, Andreas Krause, and Moritz Hardt. Learning on the job: Test-time curricula for targeted reinforcement learning, 2025. URL https://arxiv.org/abs/2510.04786.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, Yunlong Feng, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5-coder technical report, 2024. URL https://arxiv.org/abs/2409.12186.

Hynek Kydlíček. Math-Verify: Math Verification Library. URL https://github.com/huggingface/math-verify.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. GSM-plus: A comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2961–2984. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.163. URL https://aclanthology.org/2024.acl-long.163/.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling, 2025. URL https://arxiv.org/abs/2502.11886.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-Like Training: A Critical Perspective. In *Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Hao Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, Wentao Zhang, and Bin CUI. Learning What Reinforcement Learning Can't: Interleaved Online Fine-Tuning for Hardest Questions. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=LzCBLrNoyM.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-Math: Unlocking the potential of SLMs in Grade School Math, 2024. URL https://arxiv.org/abs/2402.14830.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 20275–20321. Association for Computational Linguistics, November 2025. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang,

Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. OpenAI o1 System Card, 2024. URL https://arxiv.org/abs/2412.16720.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report, 2025. URL https://arxiv.org/abs/2412.15115.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A Flexible and Efficient RLHF Framework. In *European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297. Association for Computing Machinery, 2025. ISBN 9798400711961. doi: 10.1145/3689031.3696075.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, Alex Makelov, Alex Neitz, Alex Wei, Alexandra Barr, Alexandre Kirchmeyer, Alexey Ivanov, Alexi Christakis, Alistair Gillespie, Allison Tam, Ally Bennett, Alvin Wan, Alyssa Huang, Amy McDonald Sandjideh, Amy Yang, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrei Gheorghe, Andres Garcia Garcia, Andrew Braunstein, Andrew Liu, Andrew Schmidt, Andrey Mereskin, Andrey Mishchenko, Andy Applebaum, Andy Rogerson, Ann Rajan, Annie Wei, Anoop Kotha, Anubha Srivastava, Anushree Agrawal, Arun Vijayvergiya, Ashley Tyra, Ashvin Nair, Avi Nayak, Ben Eggers, Bessie Ji, Beth Hoover, Bill Chen, Blair Chen, Boaz Barak,

Borys Minaiev, Botao Hao, Bowen Baker, Brad Lightcap, Brandon McKinzie, Brandon Wang, Brendan Quinn, Brian Fioca, Brian Hsu, Brian Yang, Brian Yu, Brian Zhang, Brittany Brenner, Callie Riggins Zetino, Cameron Raymond, Camillo Lugaresi, Carolina Paz, Cary Hudson, Cedric Whitney, Chak Li, Charles Chen, Charlotte Cole, Chelsea Voss, Chen Ding, Chen Shen, Chengdu Huang, Chris Colby, Chris Hallacy, Chris Koch, Chris Lu, Christina Kaplan, Christina Kim, CJ Minott-Henriques, Cliff Frey, Cody Yu, Coley Czarnecki, Colin Reid, Colin Wei, Cory Decareaux, Cristina Scheau, Cyril Zhang, Cyrus Forbes, Da Tang, Dakota Goldberg, Dan Roberts, Dana Palmie, Daniel Kappler, Daniel Levine, Daniel Wright, Dave Leo, David Lin, David Robinson, Declan Grabb, Derek Chen, Derek Lim, Derek Salama, Dibya Bhattacharjee, Dimitris Tsipras, Dinghua Li, Dingli Yu, DJ Strouse, Drew Williams, Dylan Hunn, Ed Bayes, Edwin Arbus, Ekin Akyurek, Elaine Ya Le, Elana Widmann, Eli Yani, Elizabeth Proehl, Enis Sert, Enoch Cheung, Eri Schwartz, Eric Han, Eric Jiang, Eric Mitchell, Eric Sigler, Eric Wallace, Erik Ritter, Erin Kavanaugh, Evan Mays, Evgenii Nikishin, Fangyuan Li, Felipe Petroski Such, Filipe de Avila Belbute Peres, Filippo Raso, Florent Bekerman, Foivos Tsimpourlas, Fotis Chantzis, Francis Song, Francis Zhang, Gaby Raila, Garrett McGrath, Gary Briggs, Gary Yang, Giambattista Parascandolo, Gildas Chabot, Grace Kim, Grace Zhao, Gregory Valiant, Guillaume Leclerc, Hadi Salman, Hanson Wang, Hao Sheng, Haoming Jiang, Haoyu Wang, Haozhun Jin, Harshit Sikchi, Heather Schmidt, Henry Aspegren, Honglin Chen, Huida Qiu, Hunter Lightman, Ian Covert, Ian Kivlichan, Ian Silber, Ian Sohl, Ibrahim Hammoud, Ignasi Clavera, Ikai Lan, Ilge Akkaya, Ilya Kostrikov, Irina Kofman, Isak Etinger, Ishaan Singal, Jackie Hehir, Jacob Huh, Jacqueline Pan, Jake Wilczynski, Jakub Pachocki, James Lee, James Quinn, Jamie Kiros, Janvi Kalra, Jasmyn Samaroo, Jason Wang, Jason Wolfe, Jay Chen, Jay Wang, Jean Harb, Jeffrey Han, Jeffrey Wang, Jennifer Zhao, Jeremy Chen, Jerene Yang, Jerry Tworek, Jesse Chand, Jessica Landon, Jessica Liang, Ji Lin, Jiancheng Liu, Jianfeng Wang, Jie Tang, Jihan Yin, Joanne Jang, Joel Morris, Joey Flynn, Johannes Ferstad, Johannes Heidecke, John Fishbein, John Hallman, Jonah Grant, Jonathan Chien, Jonathan Gordon, Jongsoo Park, Jordan Liss, Jos Kraaijeveld, Joseph Guay, Joseph Mo, Josh Lawson, Josh McGrath, Joshua Vendrow, Joy Jiao, Julian Lee, Julie Steele, Julie Wang, Junhua Mao, Kai Chen, Kai Hayashi, Kai Xiao, Kamyar Salahi, Kan Wu, Karan Sekhri, Karan Sharma, Karan Singhal, Karen Li, Kenny Nguyen, Keren Gu-Lemberg, Kevin King, Kevin Liu, Kevin Stone, Kevin Yu, Kristen Ying, Kristian Georgiev, Kristie Lim, Kushal Tirumala, Kyle Miller, Lama Ahmad, Larry Lv, Laura Clare, Laurance Fauconnet, Lauren Itow, Lauren Yang, Laurentia Romaniuk, Leah Anise, Lee Byron, Leher Pathak, Leon Maksin, Leyan Lo, Leyton Ho, Li Jing, Liang Wu, Liang Xiong, Lien Mamitsuka, Lin Yang, Lindsay McCallum, Lindsey Held, Liz Bourgeois, Logan Engstrom, Lorenz Kuhn, Louis Feuvrier, Lu Zhang, Lucas Switzer, Lukas Kondraciuk, Lukasz Kaiser, Manas Joglekar, Mandeep Singh, Mandip Shah, Manuka Stratta, Marcus Williams, Mark Chen, Mark Sun, Marselus Cayton, Martin Li, Marvin Zhang, Marwan Aljubeh, Matt Nichols, Matthew Haines, Max Schwarzer, Mayank Gupta, Meghan Shah, Melody Huang, Meng Dong, Mengqing Wang, Mia Glaese, Micah Carroll, Michael Lampe, Michael Malek, Michael Sharman, Michael Zhang, Michele Wang, Michelle Pokrass, Mihai Florian, Mikhail Pavlov, Miles Wang, Ming Chen, Mingxuan Wang, Minnia Feng, Mo Bavarian, Molly Lin, Moose Abdool, Mostafa Rohaninejad, Nacho Soto, Natalie Staudacher, Natan LaFontaine, Nathan Marwell, Nelson Liu, Nick Preston, Nick Turley, Nicklas Ansman, Nicole Blades, Nikil Pancha, Nikita Mikhaylin, Niko Felix, Nikunj Handa, Nishant Rai, Nitish Keskar, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Oona Gleeson, Pamela Mishkin, Patryk Lesiewicz, Paul Baltescu, Pavel Belov, Peter Zhokhov, Philip Pronin, Phillip Guo, Phoebe Thacker, Qi Liu, Qiming Yuan, Qinghua Liu, Rachel Dias, Rachel Puckett, Rahul Arora, Ravi Teja Mullapudi, Raz Gaon, Reah Miyara, Rennie Song, Rishabh Aggarwal, RJ Marsan, Robel Yemiru, Robert Xiong, Rohan Kshirsagar, Rohan Nuttall, Roman Tsiupa, Ronen Eldan, Rose Wang, Roshan James, Roy Ziv, Rui Shu, Ruslan Nigmatullin, Saachi Jain, Saam Talaie, Sam Altman, Sam Arnesen, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Sarah Yoo, Savannah Heon, Scott Ethersmith, Sean Grove, Sean Taylor, Sebastien Bubeck, Sever Banesiu, Shaokyi Amdo, Shengjia Zhao, Sherwin Wu, Shibani Santurkar, Shiyu Zhao, Shraman Ray Chaudhuri, Shreyas Krishnaswamy, Shuaiqi, Xia, Shuyang Cheng, Shyamal Anadkat, Simón Posada Fishman, Simon Tobin, Siyuan Fu, Somay Jain, Song Mei, Sonya Egoian, Spencer Kim, Spug Golden, SQ Mah, Steph Lin, Stephen Imm, Steve Sharpe, Steve Yadlowsky, Sulman Choudhry, Sungwon Eum, Suvansh Sanjeev, Tabarak Khan, Tal Stramer, Tao Wang, Tao Xin, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Degry, Thomas Shadwell, Tianfu Fu, Tianshi Gao, Timur Garipov, Tina Sriskandarajah, Toki Sherbakov, Tomer Kaftan, Tomo Hiratsuka, Tongzhou Wang, Tony Song, Tony Zhao, Troy Peterson, Val Kharitonov, Victoria Chernova, Vineet Kosaraju, Vishal Kuo, Vitchyr Pong, Vivek

Verma, Vlad Petrov, Wanning Jiang, Weixing Zhang, Wenda Zhou, Wenlei Xie, Wenting Zhan, Wes McCabe, Will DePue, Will Ellsworth, Wulfie Bain, Wyatt Thompson, Xiangning Chen, Xiangyu Qi, Xin Xiang, Xinwei Shi, Yann Dubois, Yaodong Yu, Yara Khakbaz, Yifan Wu, Yilei Qian, Yin Tat Lee, Yinbo Chen, Yizhen Zhang, Yizhong Xiong, Yonglong Tian, Young Cha, Yu Bai, Yu Yang, Yuan Yuan, Yuanzhi Li, Yufeng Zhang, Yuguang Yang, Yujia Jin, Yun Jiang, Yunyun Wang, Yushi Wang, Yutian Liu, Zach Stubenvoll, Zehao Dou, Zheng Wu, and Zhigang Wang. Openai gpt-5 system card, 2025. URL https://arxiv.org/abs/2601.03267.

Xiaoyu Tian, Sitong Zhao, Haotian Wang, Shuaiting Chen, Yiping Peng, Yunjie Ji, Han Zhao, and Xiangang Li. DeepDistill: Enhancing LLM Reasoning Capabilities via Large-Scale Difficulty-Graded Data Training, 2025. URL https://arxiv.org/abs/2504.17565.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and yelong shen. Reinforcement Learning for Reasoning in Large Language Models with One Training Example. In *Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=IBrRNLr6JA.

Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. On the Generalization of SFT: A Reinforcement Learning Perspective with Reward Rectification. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=Lv7PjbcaMi.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to Reason under Off-Policy Guidance. In *Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=vO8LLoNWWk.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 Technical Report, 2024a. URL https://arxiv.org/abs/2407.10671.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024b. URL https://arxiv.org/abs/2409.12122.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=N8N0hgNDRt.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free Process Rewards without Process Labels. In *International Conference on Machine Learning (ICML)*, 2025. URL https://openreview.net/forum?id=8ThnPFhGm8.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? In *Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=4OsgYD7em5.

Huaye Zeng, Dongfu Jiang, Haozhe Wang, Ping Nie, Xiaotong Chen, and Wenhu Chen. Acecoder: Acing coder rl via automated test-case synthesis. In *Annual Meeting of the Association for Computational Linguistics (ACL))*, pp. 12023–12040, 2025a.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun MA, and Junxian He. SimpleRL-Zoo: Investigating and Taming Zero Reinforcement Learning for Open Base Models in the Wild. In *Conference on Language Modeling (COLM)*, 2025b. URL https://openreview.net/forum?id=vSMCBUgrQj.

Guanhua Zhang, Ricardo Dominguez-Olmedo, and Moritz Hardt. Train-before-Test Harmonizes Language Model Rankings. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=ORv3SAzus1.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-Time Reinforcement Learning. In *Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=VuVhgEiu20.

A Details for Section 3: TTRA validation and stability

A.1 Implementation details and additional results for Section 3.1

Training details. Our implementation is based on the verl framework (Sheng et al., 2025). We use GRPO (Shao et al., 2024) for training. All models are trained using the AdamW optimizer with a cosine learning rate scheduler and a linear warmup over 10% of the total steps. During training, responses are sampled with a temperature of 0.6, using 32 rollouts and a maximum sequence length of 1024 tokens. The batch size is set to 1 for 1-shot RL and 4 for TTRL alignment. For GSM8K and GSM-plus-mini, models are trained with a learning rate of $1e{-6}$ for 100 steps, followed by $5e{-7}$ for 300 steps. For MathQA, models are trained with a learning rate of $1e{-6}$ for 100 steps, followed by $5e{-8}$ for 300 steps. Since MathQA is a multiple-choice dataset, we want to prevent the model from collapsing to predicting a single option. To this end, we rotate the answer choices and construct multiple rotated variants for the 1-shot training sample. All models are trained on H100 GPUs.

For data selection, we randomly select one example from the GSM8K training split and use the test split for TTRL and evaluation. For both MathQA and GSM-plus-mini, we randomly select one example from the test set and use the remaining examples for TTRL and evaluation.

Verification and evaluation. For verification, we use math-verify (Kydlíček) on GSM8K and GSM-plus-mini. For MathQA, we use exact string matching on either the final boxed answer or, if no boxed answer is present, the first line of the model output. This accounts for models that directly output the answer without intermediate reasoning. During evaluation, we generate 32 responses per prompt with a temperature of 0.6, a top-p of 0.95, and a maximum length of 1024 tokens.

We use the following prompt for GSM8K and GSM-plus-mini:

Prompt for GSM8K and GSM-plus-mini

```
Q: {Question}
Let's think step by step and output the final answer
within \boxed{}.
```

We use the following prompt for MathQA:

Prompt for MathQA

```
Let's think step by step and output the final answer
with only the correct option letter enclosed in
boxed{}.

Question:
{Question}

Options:
A. ...
...

Answer:
```

We provide visual illustrations of the rank harmonization effect discussed in Section 3.1. Figure 3 depicts the ranking dynamics between GSM8K and MathQA. Under direct evaluation, the rankings exhibit significant discordance with numerous crossing lines. However, applying TTRA effectively aligns the tasks, substantially harmonizing the rankings. Furthermore, Figure 11 demonstrates the non-disruptive nature of our method on the closely related GSM8K and GSM-plus-mini benchmarks. For these naturally aligned tasks, TTRA preserves the existing high consistency, confirming that our framework corrects format-induced discrepancies where they exist without negatively impacting performance or rankings on already concordant tasks.

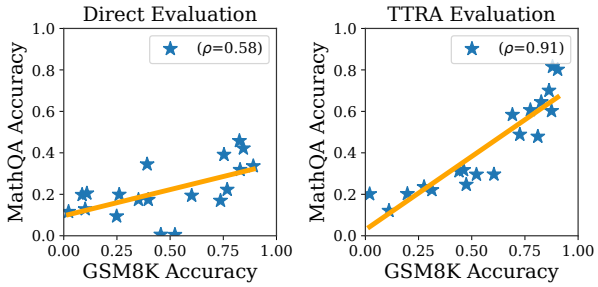

Figure 10: TTRA increases the ranking correlation between GSM8K and MathQA.**Left:** Direct evaluation shows a moderate correlation ($\rho = 0.58$). **Right:** After TTRA alignment, the correlation significantly increases ($\rho = 0.91$).

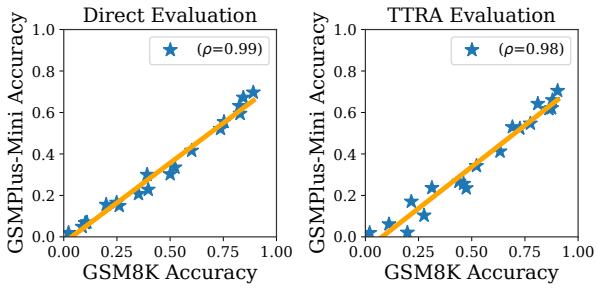

Figure 11: TTRA maintains a high correlation for already harmonized tasks. **Left:** Direct evaluation of the GSM8K and GSM-plus-mini benchmarks shows a high correlation. **Right:** TTRA alignment preserves this high correlation.

A.2 Implementation details and additional results for Section 3.2

For SFT training, we follow the configuration in Muennighoff et al. (2025). We train the models for 5 epochs with a learning rate of 1e-5, a linear warmup over 5% of the training steps, $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a weight decay of 1e-4. For TTRA training, we use the same setup as described in Appendix B.1, but with a different random seed for selecting the one-shot alignment sample.

We also provide some addition results of SFT sensitivity analysis discussed in Section 3.2. Figure 12 illustrates the performance of Qwen2.5-Math-1.5B fine-tuned on four distinct datasets (AM, s1k-1.1, Eurus-2, and OpenThoughts) across three benchmarks: MATH500, AIME2024, and OlympiadBench. We observe substantial performance fluctuations across all evaluated tasks, further confirms that SFT evaluation scores are heavily depends on the specific choice of training data.

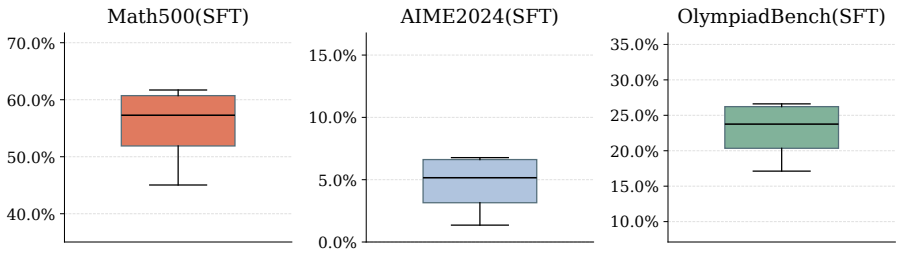

Figure 12: Sensitivity of SFT to training data selection. We evaluate the performance of the base model fine-tuned on various open-source datasets with a fixed data budget (1k samples). The results reveal significant performance variance across different data sources.

A.3 Implementation details for Section 3.3

For the consistency analysis, we sample 250 examples from the MATH500 benchmark for TTRA training and reserve the remaining 250 for held-out evaluation. We follow the same training and evaluation setup as described in Appendix B.1, but use a smaller learning rate of $1e{-6}$ during the TTRL alignment stage due to the smaller training set.

For this experiment, we evaluated Qwen2.5-Math-1.5B, along with its official Instruct variant (Yang et al., 2024b) and several fine-tuned versions, including ReLIFT (Ma et al., 2026), Oat-Zero (Liu et al., 2025), and a model trained on a 1.2k subset of DeepScalerR (Wang et al., 2025; Luo et al., 2025). In addition, we included some other Qwen models: Qwen2.5-0.5B, Qwen2.5-1.5B (Qwen et al., 2025), Qwen2-1.5B, as well as Qwen2-Math-1.5B (Yang et al., 2024a). Finally, we also evaluated models from a different architecture family: Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct (Grattafiori et al., 2024).

A.4 Consistency study across benchmarks

TTRA adapts a model to the question distribution of the benchmark on which it is applied. The split-half study in Section 3.3 holds this distribution fixed and varies only whether individual questions are seen during TTRA. Here, we additionally vary the benchmark itself. If the small post-TTRA gap between base and fine-tuned models depended on adaptation to the evaluated benchmark, applying TTRA on a different benchmark should leave a substantially larger gap.

We apply TTRA to Qwen2.5-Math-1.5B and Qwen2.5-Math-1.5B-Instruct on either MATH500 or OlympiadBench, following the setup in Appendix B.1, and evaluate the resulting models on both benchmarks without further updates. Although both benchmarks assess mathematical reasoning, they differ in difficulty and problem composition, so alignment on one benchmark is expected to transfer less effectively to the other than alignment directly on the evaluation benchmark.

As shown in Table 3, absolute accuracy is lower when TTRA is performed on the other benchmark, but the base–Instruct gap remains small. Under cross-benchmark TTRA, the gap is $+3.0$ points on MATH500 and $+4.4$ points on OlympiadBench, compared with -1.0 and $+4.0$ points under in-benchmark TTRA. Under direct evaluation, the corresponding gaps are $+41.2$ and $+17.4$ points (Liu et al., 2025). Thus, the small post-TTRA gap does not appear to depend strongly on adapting to the evaluation benchmark itself.

Table 3: Accuracy (%) of Qwen2.5-Math-1.5B Base and Instruct when TTRA is run on one benchmark and evaluated on another.

TTRA on	Evaluated on	Base	Instruct	Gap
MATH500	MATH500	75.50	74.54	-0.96
	OlympiadBench	34.63	39.04	$+4.41$
OlympiadBench	MATH500	70.01	72.99	$+2.98$
	OlympiadBench	35.53	39.57	$+4.04$

B Details for Section 4

B.1 Math reasoning evaluation setup

In this section, we provide details of the experimental setup and additional results for the reasoning tasks discussed in Section 4.

Training details All models are trained with the AdamW optimizer, a cosine learning rate schedule, and a linear warmup over 10% of the total training steps. For both one-shot alignment and TTRL, we use GRPO (Shao et al., 2024) for reinforcement learning, sampling 32 responses per question at a temperature of 0.6. We configured the 1-shot alignment stage with a learning rate of 5e-7 for 100 epochs, generating 32 rollouts per step with a maximum length of 3072 tokens for MATH500 and AIME2024 & 2025, and maximum length of 2500 tokens for the OlympiadBench. In the subsequent

TTRL stage, we trained for 300 steps with a learning rate of 5e-6 and a batch size of 4. We maintained a rollout size of 32, which was utilized for both optimization and majority voting. During the training, we used the math-verify (Kydlíček) for verifying the correctness and majority voting.

Verification and evaluation. We also use the math-verify for verification. During the evaluation, we used the temperature 0.6 and top-p of 0.95, generate 32 responses with maximum length of 3072 tokens for generation, and report the average score.

Following is the prompt we used:

Prompt for Section 4

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{Question}
Let's think step by step and output the final answer
within \boxed{}.  <|im_end|>
<|im_start|>assistant
```

Compute cost. All experiments were run on NVIDIA H100 GPUs. On MATH500, training the 1.5B model on a single H100 took roughly 0.4 hours for one-shot alignment and 1.7 hours for TTRL. The 7B model trained on 4 H100s took about 0.9 hours for one-shot alignment and 2.1 hours for TTRL.

Additional results In addition to the Qwen models, we also evaluate Llama models. We use the same user content as the prompt, but without the chat template. The results are summarized in Table 4. Compared to direct evaluation, the performance gaps between base models and their fine-tuned variants are substantially reduced after applying TTRA, further supporting the conclusions drawn in Section 4.

Table 4: Comparison between direct evaluation and evaluation using train-before-test with TTRA. Scores marked with * are reported from the original papers. Under direct evaluation, fine-tuned variants show large improvements; after applying TTRA, these gains are substantially smaller.

Llama-3.1-8B				
	MATH500		GSM8K	
Models	Direct	TTRA	Direct	TTRA
Base	13.6*	23.2	39.7*	75.7
SimpleRL-Zoo (Zeng et al., 2025b)	23.0*	27.6	79.2*	81.3
	+9.4	+4.4	+39.5	+5.6
Llama-3.2-3B-NuminaQA				
	MATH500		OlympiadBench	
Models	Direct	TTRA	Direct	TTRA
Base	0.6*	40.5	0.1*	9.6
Oat-Zero-3B (Liu et al., 2025)	50.0*	46.3	14.7*	12.5
	+49.4	+5.8	+14.6	+2.9

B.2 Code generation evaluation setup

For the one-shot alignment stage, we randomly sample a single labeled problem with ground-truth input-output cases. The model receives a reward of 1 only if its generated code passes all ground-truth test cases, and 0 otherwise.

For the TTRL stage, we use the HumanEval test set without its ground-truth test cases. Since HumanEval does not provide test cases that can be directly used as a reward signal at test time, we prompt DeepSeek-v4-flash to generate a single valid test input (i.e., only the input arguments to the target function, not an input-output pair) for each problem. For each question, we then execute all 32 model-generated code solutions on this shared input and collect their outputs. Majority voting over these outputs yields a pseudo-label, which is used as the reward signal during TTRL. This procedure requires neither ground-truth outputs nor external correctness verification. The prompt used to elicit test inputs from DeepSeek-v4-flash is shown below.

Prompt for input case generation

```
System: You are an AI assistant that helps people
with python coding tasks.

User: You are an expert at generating valid test
inputs for Python functions.
Given the following Python function, generate exactly
1 valid set of input arguments.
Function name: {fn_name}
{instruction}
Return ONLY a JSON object in this exact format:
{
  "fn_name": "{fn_name}",
  "inputs": [["arg1", "arg2", ...]]
}
Rules:
- "inputs" is a list containing exactly 1 test case.
- Each test case is a list of strings, one per function
argument, in order.
- Each string must be a valid Python literal (e.g.
"[1, 2, 3]" for a list, "0.5" for a float, "\"hello\""
for a string, "True" for a boolean).
- Do NOT write assert statements, expected outputs, or
any explanations.
- Return only the JSON object, nothing else.
```

Training details. All models are trained with the AdamW optimizer, a cosine learning rate scheduler, and a linear warmup over 10% of the total training steps. We use GRPO (Shao et al., 2024) for both stages, sampling 32 responses per question at a temperature of 0.6. For the one-shot alignment stage, we train for 100 steps with a learning rate of $1e{-}6$ and a batch size of 1. For the TTRL stage, we train for 100 steps with a learning rate of $5e{-}7$ and a batch size of 4, using 32 generations per prompt for both optimization and majority voting. We use the `prime_code` verifier to check correctness during training.

C Implementation details for Section 5: domain-specific benchmarks

C.1 Case study 1: clinical decision making

For the MIMIC-CDM-FI dataset, we randomly sampled 1 case from each pathology for the 1-shot alignment, which is 4 samples in total. And the remaining for the evaluation.

Training details. We follow the training setup described in Appendix B.1, with the following differences: we set the maximum input length to 4000 tokens and the maximum output length to 1024 tokens. For one-shot alignment, we use a learning rate of $1e{-}7$ for 100 steps. For TTRL alignment, we use the same learning rate for 300 steps for Qwen3 models, and $5e{-}8$ for Gemma models.

Verification and evaluation. For verification, we extract the answer from the final boxed content and perform exact string matching. Consistent with our previous settings, we generate 32 responses per prompt using a temperature of 0.6, a top-p of 0.95, and a maximum sequence length of 1024 tokens. For accuracy calculation, we first compute the accuracy for each pathology and then average the results across the 4 pathologies.

The following prompt is used for Qwen3 models. For Gemma models, we use the same prompt without the chat template.

Prompt for Section 5

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
You are an expert Clinical Decision Support System.
Your goal is to analyze patient data against a
provided set of diagnostic criteria to identify the
most likely pathology.
{Diagnostic Criteria}
{Patient History}
{Physical Examination}
{Laboratory Tests}
{Imaging Reports}
Based on the above information, identify the most
likely pathology affecting the patient.  Please
explain your reasoning, and then provide the final
answer with only the correct pathology enclosed in
\boxed{}.<|im_end|>
<|im_start|>assistant
```

The prompt used to summarize the long report is shown below:

Prompt for report summary

```
You are a medical artificial intelligence assistant.
Your goal is to effectively, efficiently and
accurately reduce text without inventing information.
You want to return verbatim observations that are
abnormal and of interest to a possible diagnosis of
the patient.  Normal observations can be combined.  Do
not invent information.  Use medical abbreviations
when possible to save characters.  Put the most
important information first.  Please summarize the
following result:
{Report}
Summary:
```

Compute cost. Training was conducted on NVIDIA H100 GPUs. On MIMIC-CDM-FI, Qwen3-1.7B (1 H100) took about 0.7 hours for one-shot alignment and 3.6 hours for TTRL. Qwen3-4B (2 H100s) took about 0.9 and 4.4 hours, respectively. Qwen3-8B (4 H100s) took about 0.9 and 3.7 hours, respectively.

Conversational setup. In the conversational setting (Hager et al., 2024), the model is given only the patient history together with a list of diagnostic criteria for the four candidate pathologies, and must iteratively request additional clinical information before issuing a diagnosis. We implement this as a multi-turn ReAct-style loop using the verl-agent framework (Feng et al., 2025). At each turn, the model outputs a single action enclosed in `\boxed{...}` which is either (i) the name of one diagnostic test to request, drawn from a fixed list of 12 tools (Physical Examination, Complete Blood Count, Basic / Comprehensive Metabolic Panel, Renal / Liver Function Panel, Urinalysis, Electrolyte

Panel, CT, MRI, Radiograph, Ultrasound), or (ii) one of the four pathologies as a final diagnosis. After a tool action, the corresponding test result is appended to the conversation as the next user message. The episode terminates when the model outputs a pathology or when the maximum of 6 turns is reached. The initial prompt for each case is shown below.

Prompt for MIMIC-CDM (conversational)

```
User: You are an expert Clinical Decision Support
System. Your goal is to analyze patient data against
a provided set of diagnostic criteria to identify the
most likely pathology.
To aid you in your diagnosis, you can request the
results for the following tests for the patient:
- Physical Examination
- Complete Blood Count
- Basic Metabolic Panel
- Comprehensive Metabolic Panel
- Renal Function Panel
- Liver Function Panel
- Urinalysis
- Electrolyte Panel
- CT
- MRI
- Radiograph
- Ultrasound
You can only request one test at a time. After
receiving the results, you should analyze them
and decide on the next best step, which could be
requesting another test or making a diagnosis.
{Diagnostic Criteria}
{Patient History}
Based on the information above, either make a
diagnosis if the evidence is sufficient, or if the
evidence is insufficient, select one tool from the
list above to obtain additional information. Do
not explain your reasoning. Provide only your final
answer as a single item, either the chosen tool or
the predicted pathology, enclosed in \boxed{} with no
additional text.
```

Training and evaluation. We train multi-turn rollouts using the verl-agent (Feng et al., 2025) implementation of GRPO with a per-episode turn budget of 6 and a maximum of 512 tokens per turn. The reward is 1 if the final diagnosis matches the ground-truth pathology (or the majority-voted pseudo-label, in the TTRL stage) and 0 otherwise. The one-shot stage uses a learning rate of $5\mathrm{e}{-7}$ and the TTRL stage $5\mathrm{e}{-8}$; both are trained for 100 steps. Evaluation accuracy is computed in the same way as in the full-information setting.

C.2 Case study 2: legal tasks with rubric-based evaluation

We use the LEXam (Fan et al., 2026) English open-ended subset and follow the two-stage TTRA pipeline. For the one-shot stage, we use a single labeled example with an expert-written reference answer. We ask DeepSeek-v4-flash to compare the model response against the reference and assign a continuous correctness score in $[0, 1]$ using the same rubric-based judge prompt as Fan et al. (2026); this score is used directly as the training reward.

For the TTRL stage, neither ground-truth answers nor expert references are available. For each question, we sample N responses $\{\hat{y}_k\}_{k=1}^{N}$ from the model and select a pseudo-reference answer y^*

by embedding-similarity-based majority voting:

$$y^* = \arg \max_{y \in \{\hat{y}_k\}_{k=1}^N} \sum_{j=1}^{N} V(y, \hat{y}_j),$$

where $V(\cdot, \cdot)$ denotes the cosine similarity between Qwen3-Embedding-0.6B embeddings of the two responses. We then ask DeepSeek-v4-flash to score each of the N responses against y^* using the same rubric-based prompt, and use the score as the TTRL reward.

Training details. All models are trained with the AdamW optimizer, a cosine learning rate scheduler, and a linear warmup over 10% of the total training steps. We use GRPO (Shao et al., 2024) for both stages, with a maximum response length of 2048 tokens. For the one-shot alignment stage, we train for 150 steps with a learning rate of $1e{-6}$, a batch size of 1, and 16 responses per prompt at temperature 0.6. For the TTRL stage, we train for 150 steps with a learning rate of $5e{-7}$ and a batch size of 1, sampling 16 responses per prompt at temperature 0.6 for optimization and using $N = 32$ votes per prompt for the embedding-based majority voting.

Verification and evaluation. For evaluation, we generate 32 responses per prompt with a temperature of 0.6 and a maximum length of 2048 tokens, score each response with DeepSeek-v4-flash against the held-out expert reference answer using the same rubric-based judge prompt as Fan et al. (2026), and report the average score.

Compute cost. On NVIDIA H100 GPUs, both the one-shot and TTRL stages took roughly 5 hours each for Qwen2.5-1.5B-Instruct and Qwen3-1.7B on a single H100, and similar per-stage times for Qwen3-4B on 2 H100s.

D Additional details and results for Section 6: discussions and limitations

D.1 Implementation details for Section 6.1

In Section 6.1, we conduct a stage-wise ablation study of TTRA. For training on MATH500, we adopt the same settings as described in Appendix B.1. For the MIMIC-CDM-FI benchmark, we follow a similar setup to that in Appendix C.1. However, since the models exhibit weaker capability on this benchmark, we use a learning rate of $5e{-6}$ for one-shot alignment and $5e{-8}$ for TTRL alignment.

D.2 Implementation details results for Section 6.2

For Section 6.2, all training and inference are conducted on NVIDIA H100 GPUs. The TTRA training and evaluation follow the same setup as described in Appendix A.1. For SFT training, we adopt a training procedure similar to that of Dominguez-Olmedo et al. (2025a). The model is trained using the AdamW optimizer with a cosine learning rate scheduler and a warmup ratio of 1%, a learning rate of $2e{-6}$, and a weight decay of 0.1. The batch size is set to 64. Training data consist of a combination of MetaMath (Yu et al., 2024) and Orca-Math (Mitra et al., 2024).

Table 5: Evaluation time and relative computational cost of direct evaluation and train-before-test methods (SFT and TTRA) using Qwen3-0.6B on GSM8K.

Method	**Time**	**Relative time**
Direct Evaluation	19 min	1.0
Train-before-test (SFT)		32×
SFT	10 h	
Inference	19 min	
Train-before-test (TTRA)		6.6×
One-shot RL	15 min	
TTRL	92 min	
Inference	19 min	