From Language to Logic: Unlocking General Reasoning by Training on Natural Language Inference

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) has improved LLM reasoning in domains such as math and code, but the gains can be domain-specific and do not always transfer. We revisit Natural Language Inference (NLI) and observe an **NLI paradox**: despite massive pretraining, general-purpose LLMs can trail a strong DeBERTa-style NLI encoder by **about 7 points on average** and show weak separation between "simple" and "hard" NLI instances., signaling a failure to master fundamental logical relations. To bridge this gap, we recast NLI as a verifiable, generative reinforcement learning (RL) task. By optimizing Qwen-2.5-instruct models with Group Relative Policy Optimization (GRPO) and a logic-centric reward, we force the internalization of relational logical primitives. Our approach resolves the NLI Paradox, achieving a **12% performance leap** and surpassing DeBERTa baselines. Most notably, pure NLI training exhibits powerful **cross-domain transfer**: without any domain-specific data, it yields average gains of **+3.6%** in math, **+2.4%** in code, and **+1.3%** on general reasoning benchmarks, with a remarkable **+26.5%** boost on MATH500 (3B). Furthermore, NLI-trained models generate **7.4% fewer tokens**, demonstrating a more efficient and compact reasoning style. Our results suggest that NLI-based RL strengthens universal meta-reasoning skills—such as consistency checking and evidence integration—enabling broader cognitive transfer than traditional math-centric post-training.

1 Introduction

Building Large Language Models (LLMs) with robust, cross-domain reasoning capabilities remains a fundamental challenge (Huang and Chang, 2023). Recent breakthroughs in Reinforcement Learning with Verifiable Rewards (RLVR) (Wen et al., 2025; DeepSeek-AI et al., 2025) have demonstrated that optimizing models on math and code data can unlock emergent reasoning behaviors like self-correction and significantly boost problem-solving accuracy. However, these gains are often *domain-specific*. Extensive training on math benchmarks frequently fails to transfer to—and can even degrade performance on—general logical reasoning tasks (Wang et al., 2024; Li et al., 2025). This suggests a critical bottleneck: instead of acquiring a universal reasoning engine, models may be overfitting to domain-specific patterns, utilizing answer-centric shortcuts rather than mastering fundamental deductive principles.

We revisit Natural Language Inference (NLI)—a foundational task for truth-conditional reasoning—and observe what we term the **NLI paradox** . As illustrated in Figure 1, this paradox manifests in two surprising ways. First, there is a **performance gap**: despite massive pretraining and instruction tuning, general LLMs consistently trail specialized, much smaller NLI encoders (like DeBERTa) (He et al., 2021) by approximately **7 points on average** (Fig. 1, B). Second, there is a **discriminative collapse**: while specialized models show a clear performance hierarchy between "easy" (SNLI/MNLI) (Bowman et al., 2015; Williams et al., 2018) and "hard" (ANLI) instances (Nie et al., 2020), general LLMs often exhibit a compressed range, failing to dominate easy cases while sometimes performing anomalously high on hard ones (Fig. 1, A). This suggests that their reasoning may rely more on surface-level pattern matching than on a systematic logical kernel. If NLI captures the atomic units of linguistic logic, this weakness implies that scale alone does not guarantee robust inference, motivating our use of NLI as a direct, *verifiable training signal*.

As shown in Figure 1(C), we propose a **Logic-First** training paradigm. We recast NLI as a **verifiable, generative reinforcement learning task**, utilizing Group Relative Policy Optimization (GRPO) to optimize models on NLI datasets. Unlike math

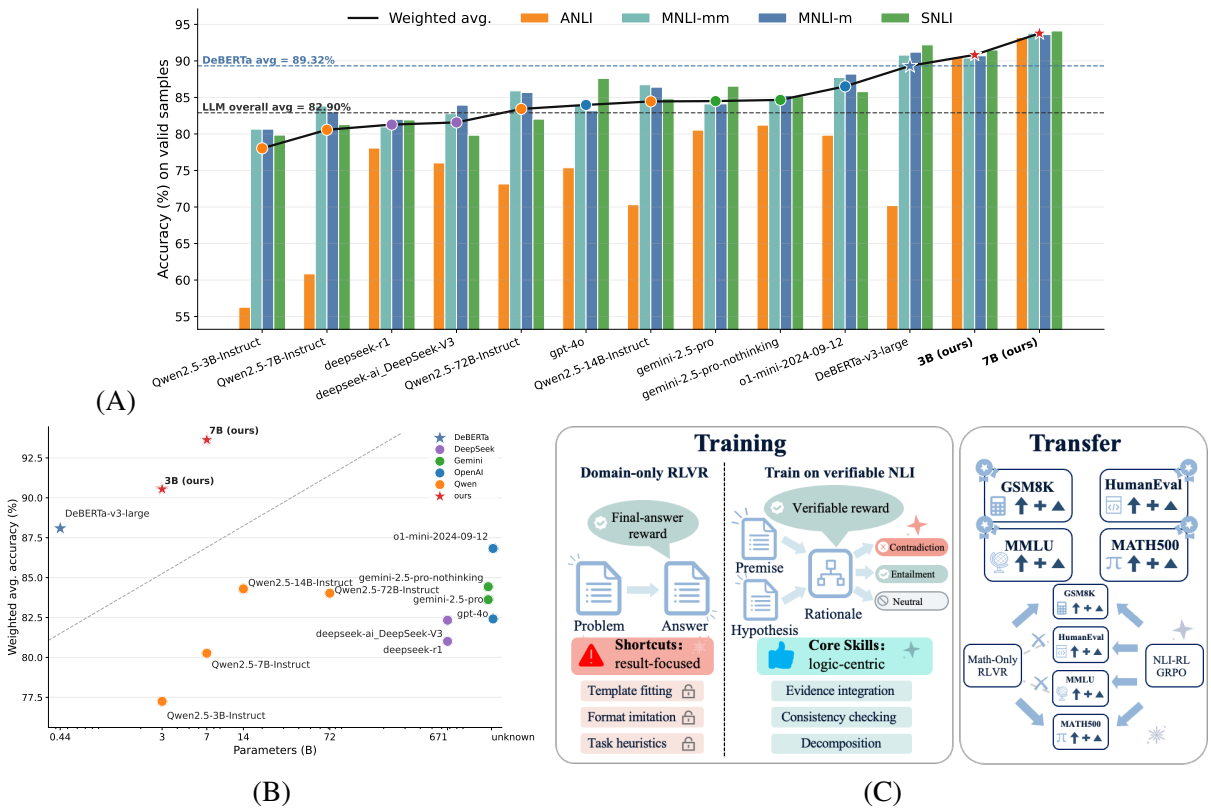

Figure 1: **Overview of the NLI paradox and our logic-first solution. (A) Paradox across difficulty.** Despite massive pretraining, general-purpose LLMs can underperform a specialized NLI encoder (DeBERTa) on the weighted average, and show an implausible pattern where hard NLI (ANLI) is relatively competitive but easy NLI (SNLI/MNLI) remains far behind the encoder, indicating weak discriminative reasoning. **(B) Paradox under scaling.** NLI ability is *not* positively correlated with parameter count: increasing model size does not reliably improve NLI. Our logic-first post-training produces a phase transition, improving NLI by nearly **12%** on average for Qwen-2.5-3B/7B-instruct (Team, 2024) and surpassing the DeBERTa baseline at matched scale. **(C) Overview (Logic-first RLVR via verifiable NLI).** We train on verifiable NLI with a structured rationale+label objective to strengthen reusable logic primitives, which leads to effective cross-domain transfer as quantified in later sections.

problems that focus on deterministic numeric targets, NLI provides a **relational training signal**, forcing the model to internalize the underlying logical consistency between a premise and a hypothesis. We posit that this relational grounding strengthens **logical primitives**—reusable cognitive skills such as evidence integration and consistency checking—that naturally radiate to other reasoning-intensive domains.

Our experiments on Qwen-2.5-instruct (3B/7B) (Team, 2024) demonstrates the efficacy of our method, as evidenced by three core contributions:

1. **Resolution of the NLI Paradox (The Leap):** Our Logic-First training leads to a phase transition in NLI capability. As shown in Figure 1(A), our models achieve a massive performance jump, improving by nearly **12%** on average and surpassing the strong DeBERTa baseline under comparable parameter counts. This confirms that the logic kernel can be effectively repaired.

2. **Broad Cross-Domain Transfer:** This strengthened logic kernel transfers widely. In our best pure-NLI setting on Qwen-2.5-7B, we observe average gains of **+3.5** on math, **+2.4** on code, and **+1.3** on general reasoning benchmarks. For smaller models (3B), gains on challenging math (MATH500) (Lightman et al., 2023a) reach up to **+26.5** points.

3. **Efficiency and Compactness:** Perhaps most strikingly, NLI training fundamentally alters *how* the model thinks. As visualized in Section 4 (Figure 2), comparing our NLI-trained models against Math-trained models reveals that NLI training significantly reduces token usage (by ∼7.4% on correct solutions). Unlike math training, which can encourage ver-

bose chain-of-thought patterns, logic training seemingly promotes a more concise, efficient reasoning style without sacrificing accuracy.

Overall, our results suggest a **Logic-First** training signal—verifiable NLI reasoning—can serve as a practical catalyst for broad, efficient, and transferable reasoning capabilities.

2 Related Work

2.1 RL Post-Training with Verifiable Rewards

Reinforcement learning (RL) has become a key paradigm for post-training LLMs beyond supervised fine-tuning, including RLHF-style objectives (Ouyang et al., 2022) and policy-optimization methods such as PPO (Schulman et al., 2017). A particularly effective setting is RLVR, where rewards come from automatic checkers (e.g., exact-match in math or execution-based verification in code). Recent systems report strong gains in mathematics and coding by incentivizing longer deliberation and self-correction (Guo et al., 2025; Yeo et al., 2025; Lightman et al., 2023b). However, verifiable signals can be *domain-shaped*: optimizing against narrow verifiers may encourage shortcuts (templates, formatting tricks, heuristics) that do not reliably transfer. We address this by using NLI—a verifiable but *relational* signal—as the sole RL objective to promote reusable logical primitives.

2.2 Natural Language Inference and Robust Textual Entailment

Natural Language Inference (NLI), also known as Recognizing Textual Entailment (RTE) (Poliak, 2020), benchmarks truth-conditional reasoning between a *premise* and a *hypothesis*. Datasets such as SNLI/MultiNLI (Bowman et al., 2015; Williams et al., 2018) and harder challenge sets like ANLI (Nie et al., 2020) are widely used to probe robustness, including failures due to spurious heuristics (McCoy et al., 2019). While encoder models (e.g., BERT/DeBERTa) remain strong on NLI (Devlin et al., 2019; He et al., 2021), generative LLMs can be brittle on hard entailment, motivating NLI as not only an evaluation probe but also a training signal. Prior work uses NLI supervision for transferable representations (Chen et al., 2024), and rationale-augmented resources (e.g., e-SNLI) provide explicit explanations (Camburu et al., 2018). In contrast, we cast NLI as a *verifiable generative RL* task, optimizing structured reasoning plus a final label.

2.3 Reasoning Transfer, Process Supervision, and Relational Inductive Bias

Which training signals yield transferable reasoning beyond the source domain remains open. Chain-of-thought prompting/supervision emphasizes intermediate reasoning (Wei et al., 2023), while analyses caution that gains may reflect recitation or shortcut learning (Yan et al., 2025). From an inductive-bias perspective, supervision that emphasizes *relations* (e.g., consistency) can be more reusable than task-specific final answers (Conneau et al., 2017). Our approach complements process supervision and verifier-based training: by using verifiable NLI decisions as a relational signal, we encourage evidence integration, consistency checking, and decomposition, and evaluate transfer to math, code, and general reasoning under scale-matched settings.

3 Methodology

We introduce a framework that reshapes NLI into a verifiable reinforcement learning task, with the goal of inducing transferable reasoning abilities. Our method focuses on two core designs: (1) a generative reformulation of NLI that exposes reasoning as an explicit action space, and (2) a logic-centric reward that directly supervises relational consistency rather than domain-specific answers. We train the model using GRPO (Shao et al., 2024). Additionally, we analyze the nature of the NLI training signal in Section 3.3, contrasting it with mathematical reasoning to clarify why NLI-based supervision promotes cross-domain generalization.

3.1 Task Formulation: Generative NLI

Let $\mathcal{D}_{\text{NLI}} = \{(P, H, L^*)\}$ denote an NLI dataset, where P is the premise, H is the hypothesis, and $L^* \in \mathcal{L} = \{\text{Entailment, Contradiction, Neutral}\}$ represents the ground-truth logical relation. Traditionally, NLI is modeled as a discriminative task $\mathcal{P}(L|P, H)$ to predict a label given the input pair.

To induce explicit reasoning, we reformulate this as a sequential generation task. Let $x = (P, H)$ be the input. The model π_θ must generate a reasoning chain $C = (c_1, c_2, \ldots, c_k)$ followed by a final answer $A \in \mathcal{L}$. The joint probability is defined as:

$$\mathcal{P}_\theta(C, A|x) = \prod_{t=1}^{|C|} \mathcal{P}_\theta(c_t|x, c_{<t}) \cdot \mathcal{P}_\theta(A|x, C) \quad (1)$$

This formulation forces the model to externalize the logical steps linking P and H before committing

to a label, thereby transforming latent reasoning into an observable and rewardable action space.

3.2 Logic-centric Reward Engineering

The reward function $R(o)$ is the guiding signal for logic acquisition. Since NLI has a ground truth label L^*, we can employ a hard correctness reward augmented by structural constraints. The total reward for an output $o = (C, A)$ is:

$$R(o) = \lambda_{\text{acc}} \cdot \mathbb{I}(A = L^*) + \lambda_{\text{format}} \cdot S(C) \quad (2)$$

where λ_{acc} and λ_{format} are scaling coefficients that balance task accuracy and structural constraints. The components are defined as follows:

- **Correctness Reward** $\mathbb{I}(A = L^*)$: An indicator function that yields 1 if the model's final answer A matches the ground-truth label L^*, and 0 otherwise.
- **Structure Reward** $S(C)$: A validity check ensuring the output follows the XML format: `<reasoning>...</reasoning> <answer>...</answer>`.

By optimizing this reward using GRPO, the model learns that the only reliable way to maximize reward is to generate valid, structured, and sufficiently detailed logical derivations.

3.3 Nature of the NLI Training Signal

To clarify why our framework fosters genuine reasoning, we contrast the NLI training signal with that of mathematical tasks. In mathematical tasks, the reward is **answer-centric**, focusing on matching a deterministic answer a from $\mathcal{D}_{\text{Math}} = \{(q, a)\}$. In contrast, our NLI reward is **relational**: $\mathbb{I}(A = L^*)$ depends on verifying semantic consistency between P and H in $\mathcal{D}_{\text{NLI}}$. This relational nature forces the model to optimize for logical recovery of entailment rather than pattern matching. While answer-centric signals risk rote memorization, the NLI signal compels learning the underlying logic, moving from outcome recall to acquiring transferable reasoning skills.

Our hypothesis is that this framework not only boosts genuine reasoning but also transfers to other inference domains. Experimental validation of these gains is detailed in Section 4 .

4 Experiments

We evaluate whether verifiable NLI training improves *reasoning transfer* beyond the source domain. To keep the section reviewer-friendly, we structure the experiments around a small set of questions: (RQ1) Is there an "NLI paradox" for modern LLMs? (RQ2) Does NLI-RL transfer to math/code/general reasoning? (RQ3) How does it compare to domain-specific math training? (RQ4) What factors drive transfer (e.g., input length), and is it logic rather than format?

4.1 Experimental Setup

Base Models: Without loss of generality, we conduct experiments using `Qwen-2.5-3B-Instruct` and `Qwen-2.5-7B-Instruct` (Team, 2024), chosen for their strong baseline performance and instruction-following capabilities.

Training Datasets:

- **NLI-Reasoning (Ours):** A curated dataset of 1000k samples merged from **SNLI**, **MNLI**, and **ANLI**. (Bowman et al., 2015; Williams et al., 2018; Nie et al., 2020)
- **Mathematical Datasets:**
  - *Simple Math:* `PRM800k`, representing standard arithmetic and solution-level supervision.(Lightman et al., 2023a)
  - *Hard Math (DAPO):* A dataset focusing on complex, competition-level mathematics (used for comparison where applicable).(Yu et al., 2025)
  - *GSM8K-Train:* The standard training set of GSM8K.(Cobbe et al., 2021)

Evaluation Benchmarks:

- **Mathematics: GSM8K** (Grade School Math) (Cobbe et al., 2021) and **MATH500** (Subsampled Challenging Math) (Lightman et al., 2023a).
- **Coding: HumanEval** (Chen et al., 2021) and **MBPP** (Sanitized) (Austin et al., 2021).
- **General Reasoning: MMLU** (Massive Multitask) (Hendrycks et al., 2021b,a) and **DROP** (Discrete Reasoning Over Paragraphs) (Dua et al., 2019).

4.2 Implementation Details (Reproducibility)

Full training/evaluation details (GRPO hyperparameters, prompts, and reward specification) are provided in the appendix (Appendix A, §A.1–§A.2). Here we focus on the core empirical findings.

Model Scale	Training Data	Math		Code		Reasoning		Avg.
		GSM8K	MATH500	HumanEval	MBPP	MMLU	DROP	Imp.
3B	Base	77.7	30.0	75.0	57.8	64.8	52.2	–
	Trained on domain data (Math & code)							
	Qwen2.5-Coder-instruct	80.7	–	**84.1**	**73.6**	56.5	–	–
	DAPO	83.6	55.4	73.2	58.2	66.5	48.7	–
	Pure NLI Variants (Ours)							
	ANLI	82.1$_{\uparrow 4.4}$	56.5$_{\uparrow 26.5}$	75.6$_{\uparrow 0.6}$	59.2$_{\uparrow 1.4}$	**66.9**$_{\uparrow 2.1}$	54.7$_{\uparrow 2.5}$	**$\uparrow$6.3**
	NLI_ALL	**85.8**$_{\uparrow 8.1}$	54.6$_{\uparrow 24.6}$	75.2$_{\uparrow 0.2}$	58.8$_{\uparrow 1.0}$	66.8$_{\uparrow 2.0}$	53.5$_{\uparrow 1.3}$	$\uparrow$6.2
	Mixed training (Ours)							
	GSM8K+NLI	83.7$_{\uparrow 6.0}$	56.1$_{\uparrow 26.1}$	74.4$_{\downarrow 0.6}$	59.4$_{\uparrow 1.6}$	66.4$_{\uparrow 1.6}$	**56.1**$_{\uparrow 3.9}$	**$\uparrow$6.4**
	DAPO+NLI	84.7$_{\uparrow 7.0}$	**57.0**$_{\uparrow 27.0}$	74.4$_{\downarrow 0.6}$	57.0$_{\downarrow 0.8}$	66.4$_{\uparrow 1.6}$	49.9$_{\downarrow 2.3}$	$\uparrow$5.3
7B	Base	86.1	63.2	82.3	63.2	74.4	65.5	–
	Trained on domain data (Math & code)							
	Qwen2.5-Coder-instruct	86.7	–	**88.4**	**83.5**	68.7	–	–
	DAPO	89.8	64.8	81.7	63.6	75.4	62.3	–
	Pure NLI Variants (Ours)							
	ANLI	90.2$_{\uparrow 4.1}$	65.4$_{\uparrow 2.2}$	83.5$_{\uparrow 1.2}$	65.2$_{\uparrow 2.0}$	74.9$_{\uparrow 0.5}$	66.5$_{\uparrow 1.0}$	$\uparrow$1.8
	NLI_ALL	90.1$_{\uparrow 4.0}$	66.4$_{\uparrow 3.2}$	85.4$_{\uparrow 3.1}$	64.8$_{\uparrow 1.6}$	75.1$_{\uparrow 0.7}$	**67.4**$_{\uparrow 1.9}$	**$\uparrow$2.4**
	Mixed training (Ours)							
	GSM8K+NLI	**90.6**$_{\uparrow 4.5}$	65.2$_{\uparrow 2.0}$	85.4$_{\uparrow 3.1}$	65.0$_{\uparrow 1.8}$	74.9$_{\uparrow 0.5}$	66.4$_{\uparrow 0.9}$	$\uparrow$2.1
	DAPO+NLI	90.3$_{\uparrow 4.2}$	**67.0**$_{\uparrow 3.8}$	84.2$_{\uparrow 1.9}$	65.4$_{\uparrow 2.2}$	**75.6**$_{\uparrow 0.9}$	64.1$_{\downarrow 1.4}$	$\uparrow$1.9

Table 1: **Main results under a unified evaluation format.** All models share the same backbone (`Qwen-2.5-3B-Instruct` or `Qwen-2.5-7B-Instruct`), enabling a fair comparison across different training data and paradigms. Results are reported separately for the 3B and 7B model scales. The "Avg. Imp." column summarizes the average performance change relative to the corresponding Base model, highlighting the strong cross-domain generalization of NLI-based training. For each metric within the same model scale, the **best** result is shown in **bold**, and the second best is underlined. Performance improvements and degradations relative to the Base model are indicated by green upward arrows ($\uparrow$) and red downward arrows ($\downarrow$), respectively.

4.3 RQ1: The NLI Paradox

We first quantify the **NLI paradox** introduced in Section 1. Figure 1 (B) highlights that modern LLMs can remain surprisingly weak on NLI: (i) **NLI ability does not scale monotonically with parameter count**—larger, stronger LLMs are not consistently better at NLI; and (ii) even for high-parameter, high-performing LLMs, there remains a **large gap** to specialized NLI encoders (e.g., DeBERTa). In the top plot, "weighted avg" denotes the **dataset-count-weighted** average over the evaluated NLI datasets (i.e., each dataset receives equal weight, rather than weighting by the number of instances).

Simple vs. Hard NLI. In our setting, "hard NLI" refers to adversarial or challenge-style NLI instances (e.g., ANLI-style), and "simple NLI" refers to standard NLI instances (e.g., SNLI/MNLI-style). We keep the label space and evaluation protocol identical across subsets; only the instance distribution differs. (Full dataset construction details and subset definitions will be provided in the final version.)

Discriminative collapse (easy–hard separation). Figure 1 (A) further shows that general-purpose LLMs can exhibit a compressed separation between easy (SNLI/MNLI) and hard (ANLI) NLI subsets, compared to specialized NLI encoders. Our NLI-trained models restore a clearer easy–hard ordering while also closing the overall NLI performance gap.

4.4 Main Results: Generalization of NLI Reasoning

We organize our evaluation into three categories commonly used to probe LLM reasoning: **Math** (GSM8K, MATH500), **Code** (HumanEval, MBPP), and **General Reasoning** (MMLU, DROP).

Table 1 presents the results for our primary NLI training strategies. Across the *reported* benchmarks, we observe a consistent trend: training on

Model Scale	Training Data	Math		Code		Reasoning		Avg.
		GSM8K	MATH500	HumanEval	MBPP	MMLU	DROP	Imp.
3B Models	Base	77.7	30.0	75.0	57.8	64.8	52.2	–
	Pure NLI Variants							
	MNLI	76.7$_{\downarrow 1.0}$	40.8$_{\uparrow 10.8}$	**78.7**$_{\uparrow 3.7}$	58.0$_{\uparrow 0.2}$	65.8$_{\uparrow 1.0}$	53.6$_{\uparrow 1.4}$	$\uparrow$2.7
	SNLI	74.3$_{\downarrow 3.4}$	48.6$_{\uparrow 18.6}$	75.6$_{\uparrow 0.6}$	58.8$_{\uparrow 1.0}$	66.3$_{\uparrow 1.5}$	53.5$_{\uparrow 1.3}$	$\uparrow$3.3
	ANLI	82.1$_{\uparrow 4.4}$	<u>56.5</u>$_{\uparrow 26.5}$	75.6$_{\uparrow 0.6}$	<u>59.2</u>$_{\uparrow 1.4}$	**66.9**$_{\uparrow 2.1}$	<u>54.7</u>$_{\uparrow 2.5}$	$\uparrow$6.3
	NLI_ALL	**85.8**$_{\uparrow 8.1}$	54.6$_{\uparrow 24.6}$	75.2$_{\uparrow 0.2}$	58.8$_{\uparrow 1.0}$	<u>66.8</u>$_{\uparrow 2.0}$	53.5$_{\uparrow 1.3}$	$\uparrow$6.1
	NLI_SHORT	74.1$_{\downarrow 3.6}$	37.9$_{\uparrow 7.9}$	75.6$_{\uparrow 0.6}$	58.0$_{\uparrow 0.2}$	66.1$_{\uparrow 1.3}$	52.5$_{\uparrow 0.3}$	$\uparrow$1.0
	NLI_LONG	<u>84.7</u>$_{\uparrow 7.0}$	45.9$_{\uparrow 15.9}$	<u>76.2</u>$_{\uparrow 1.2}$	**59.4**$_{\uparrow 1.6}$	66.2$_{\uparrow 1.4}$	53.6$_{\uparrow 1.4}$	$\uparrow$4.8
	Avg. (NLI variants)	79.6$_{\uparrow 1.9}$	47.4$_{\uparrow 17.4}$	76.1$_{\uparrow 1.1}$	58.7$_{\uparrow 0.9}$	66.4$_{\uparrow 1.6}$	53.6$_{\uparrow 1.4}$	$\uparrow$4.0
	Math & Mixed Baselines							
	Simple Math (PRM800k)	–	–	–	–	–	–	–
	Hard Math (DAPO)	83.6$_{\uparrow 5.9}$	55.4$_{\uparrow 25.4}$	73.2$_{\downarrow 1.8}$	58.2$_{\uparrow 0.4}$	66.5$_{\uparrow 1.7}$	48.7$_{\downarrow 3.5}$	$\uparrow$4.7
	GSM8K_NLI	83.7$_{\uparrow 6.0}$	56.1$_{\uparrow 26.1}$	74.4$_{\downarrow 0.6}$	**59.4**$_{\uparrow 1.6}$	66.4$_{\uparrow 1.6}$	**56.1**$_{\uparrow 3.9}$	$\uparrow$6.4
	DAPO_NLI	<u>84.7</u>$_{\uparrow 7.0}$	**57.0**$_{\uparrow 27.0}$	74.4$_{\downarrow 0.6}$	57.0$_{\downarrow 0.8}$	66.4$_{\uparrow 1.6}$	49.9$_{\downarrow 2.3}$	$\uparrow$5.3
7B Models	Base	86.1	63.2	82.3	63.2	74.4	65.5	–
	Pure NLI Variants							
	MNLI	89.8$_{\uparrow 3.7}$	64.0$_{\uparrow 0.8}$	84.2$_{\uparrow 1.9}$	65.2$_{\uparrow 2.0}$	74.8$_{\uparrow 0.4}$	65.5$_{0.0}$	$\uparrow$1.5
	SNLI	90.1$_{\uparrow 4.0}$	64.8$_{\uparrow 1.6}$	84.8$_{\uparrow 2.5}$	<u>65.6</u>$_{\uparrow 2.4}$	74.8$_{\uparrow 0.4}$	64.3$_{\downarrow 1.2}$	$\uparrow$1.6
	ANLI	90.2$_{\uparrow 4.1}$	65.4$_{\uparrow 2.2}$	83.5$_{\uparrow 1.2}$	65.2$_{\uparrow 2.0}$	74.9$_{\uparrow 0.5}$	66.5$_{\uparrow 1.0}$	$\uparrow$1.8
	NLI_ALL	90.1$_{\uparrow 4.0}$	66.4$_{\uparrow 3.2}$	<u>85.4</u>$_{\uparrow 3.1}$	64.8$_{\uparrow 1.6}$	75.1$_{\uparrow 0.7}$	67.4$_{\uparrow 1.9}$	$\uparrow$2.4
	NLI_SHORT	90.1$_{\uparrow 4.0}$	66.2$_{\uparrow 3.0}$	<u>85.4</u>$_{\uparrow 3.1}$	**66.4**$_{\uparrow 3.2}$	75.1$_{\uparrow 0.7}$	**69.0**$_{\uparrow 3.5}$	$\uparrow$2.9
	NLI_LONG	**90.6**$_{\uparrow 4.5}$	<u>66.8</u>$_{\uparrow 3.6}$	**86.0**$_{\uparrow 3.7}$	64.4$_{\uparrow 1.2}$	75.0$_{\uparrow 0.6}$	<u>67.8</u>$_{\uparrow 2.3}$	$\uparrow$2.7
	Avg. (NLI variants)	90.2$_{\uparrow 4.1}$	65.6$_{\uparrow 2.4}$	84.9$_{\uparrow 2.6}$	65.3$_{\uparrow 2.1}$	74.9$_{\uparrow 0.5}$	66.8$_{\uparrow 1.3}$	$\uparrow$2.1
	Math & Mixed Baselines							
	Simple Math (PRM800k)	88.5$_{\uparrow 2.4}$	62.4$_{\downarrow 0.8}$	83.2$_{\uparrow 0.9}$	64.2$_{\uparrow 1.0}$	74.6$_{\uparrow 0.2}$	64.6$_{\downarrow 0.9}$	$\uparrow$0.5
	Hard Math (DAPO)	89.8$_{\uparrow 3.7}$	64.8$_{\uparrow 1.6}$	81.7$_{\downarrow 0.6}$	63.6$_{\uparrow 0.4}$	<u>75.4</u>$_{\uparrow 1.1}$	62.3$_{\downarrow 3.2}$	$\uparrow$0.5
	GSM8K_NLI	**90.6**$_{\uparrow 4.5}$	65.2$_{\uparrow 2.0}$	<u>85.4</u>$_{\uparrow 3.1}$	65.0$_{\uparrow 1.8}$	74.9$_{\uparrow 0.5}$	66.4$_{\uparrow 0.9}$	$\uparrow$2.1
	DAPO_NLI	<u>90.3</u>$_{\uparrow 4.2}$	**67.0**$_{\uparrow 3.8}$	84.2$_{\uparrow 1.9}$	65.4$_{\uparrow 2.2}$	**75.6**$_{\uparrow 0.9}$	64.1$_{\downarrow 1.4}$	$\uparrow$1.9

Table 2: **Comprehensive results (ablations + mixed baselines)** This table consolidates (i) all pure-NLI variants (including the **NLI_SHORT/LONG** length split), (ii) domain-specific baselines, and (iii) mixed training, for both 3B and 7B scales. Within each model scale and each metric column, the **best** result is shown in **bold**, and the <u>second best</u> is underlined. Changes relative to the corresponding Base model are shown as subscript-style green upward arrows (↑) for improvements and red downward arrows (↓) for degradations.

pure NLI logic (e.g., **ANLI** and **NLI_ALL**) improves performance compared to the base model. We further study length-based variants in Table 2: **NLI_SHORT/LONG** are defined by the **original NLI input length** (Premise+Hypothesis) using a mean-length split.

Cross-domain transfer vs. specialization. Table 1 shows a clear trade-off between *domain specialization* and *cross-domain transfer*. Specialized models can be strongest on their home domain (e.g., Qwen2.5-Coder-7B (Hui et al., 2024) reaches 88.4/83.5 on HumanEval/MBPP), but their out-of-domain reasoning can lag (e.g., 68.7 on MMLU). In contrast, pure NLI training yields broad improvements: **NLI_ALL** reaches 74.8 MMLU at 7B and lifts MMLU from 56.5 (Coder-3B) to 66.8 at 3B, while also achieving large gains on MATH-500 (up to **+26.5** at 3B).

Mixed training mitigates cross-domain degradation. Domain-only training can narrow improvements and may reduce cross-domain robustness (e.g., 7B DAPO lowers DROP from 65.5 to 62.3). Adding NLI to domain data preserves in-domain gains while mitigating this degradation: **DAPO+NLI** attains the best MATH-500 score (67.0) and improves MMLU to 75.6, and **GSM8K+NLI** attains the best GSM8K score (90.6) while maintaining strong MMLU/DROP. This supports the view that verifiable NLI provides a transferable logic signal that complements domain rewards rather than amplifying domain-specific shortcuts.

4.5 Comparison with Domain-Specific Training

To verify whether the gains are unique to NLI logic or simply a result of post-training, we compare NLI training against math-focused training (`PRM800k`, `DAPO`) and mixed strategies in Table 2.

Accuracy: specialization vs. transfer. Table 2 shows that math-only training can yield *narrow* improvements and even regress out-of-domain metrics. For example, at 7B, `DAPO` improves MATH500 (+1.6) and MMLU (+1.1) but *drops* DROP by 3.2 points (65.5→62.3) and slightly reduces HumanEval (82.3→81.7). In contrast, pure NLI training achieves broader gains: **NLI_ALL** improves MATH500 (+3.2) while also improving HumanEval (+3.1) and DROP (+1.9), consistent with stronger cross-domain transfer.

Simple domain data can underperform pure NLI. Importantly, a *simple* domain dataset is not guaranteed to deliver strong gains even on the target domain. For instance, at 7B, `PRM800k` yields only a small overall improvement (+0.5 on Avg. Imp.) and even *reduces* MATH500 (63.2→62.4, −0.8). Meanwhile, pure NLI variants are consistently stronger: **NLI_ALL** improves both GSM8K (86.1→90.1, +4.0) and MATH500 (63.2→66.4, +3.2), and also boosts code and DROP, highlighting that the logic-centric signal can be more effective than "easy" domain-only supervision.

Mixed training: strong in-domain without catastrophic drift. Mixed strategies can partially combine the benefits: at 7B, **DAPO_NLI** achieves the best MATH500 (67.0) while keeping MMLU high (75.6). While some metrics can still be task-dependent (e.g., DROP is slightly lower than the base for `DAPO_NLI`), overall the pattern supports our claim that NLI provides a transferable logic signal that complements domain rewards rather than amplifying domain-specific shortcuts.

Efficiency: Response Token Usage. Beyond accuracy, as shown in Figure 2, we also analyze the *cost* of reasoning in terms of **generated response tokens**. We compare pure NLI post-training against math-only post-training using the same decoding and a matched-correctness subset (details in Appendix A, §A.6). Under this protocol, pure NLI training yields **7.4% fewer** response tokens on average than math-only post-training, indicating a more compact reasoning style at matched correctness.

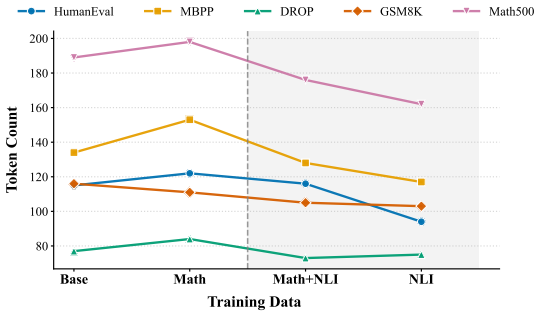

Figure 2: **Efficiency of Logic-First Training.** Average **completion token** counts on a matched-correctness subset for five benchmarks (GSM8K, MATH500, HumanEval, MBPP, DROP). The x-axis compares four post-training settings (Base, Math-only, Math+NLI, Pure NLI); the dashed vertical line separates domain-only from adding NLI. Protocol (matched correctness, decoding, and tokenization) is in Appendix A, §A.6.

For reference, we also include the best *pure NLI* setting from Table 1 for each model scale, highlighting that mixed training does not necessarily outperform the strongest NLI-only configuration.

4.6 Qualitative Analysis: Why NLI Transfers

Why NLI beats simple math. We find that many errors in math word problems stem from the *text-to-constraints* step (parsing the statement into actionable constraints), rather than arithmetic. NLI training directly exercises semantic parsing and consistency checking, which can repair this bottleneck and transfer beyond the NLI domain.

Why code can improve. We hypothesize a lightweight "logic–code correspondence": entailment/contradiction/conditionals in NLI resemble assertion/negation/branching in programs, so strengthening verifiable conditional reasoning may benefit code generation.

Case study. Figure 3 shows a representative DROP instance where the base model subtracts the wrong quantities (predicting 36), and math-only training can still perform arithmetic over irrelevant numbers (predicting 1369). In contrast, both **Math+NLI** and **Pure NLI** identify the correct quantities and relation and produce the correct answer (214).

4.7 Ablation Studies

1. The Role of Reasoning Length: We categorize NLI training data by the **original input**

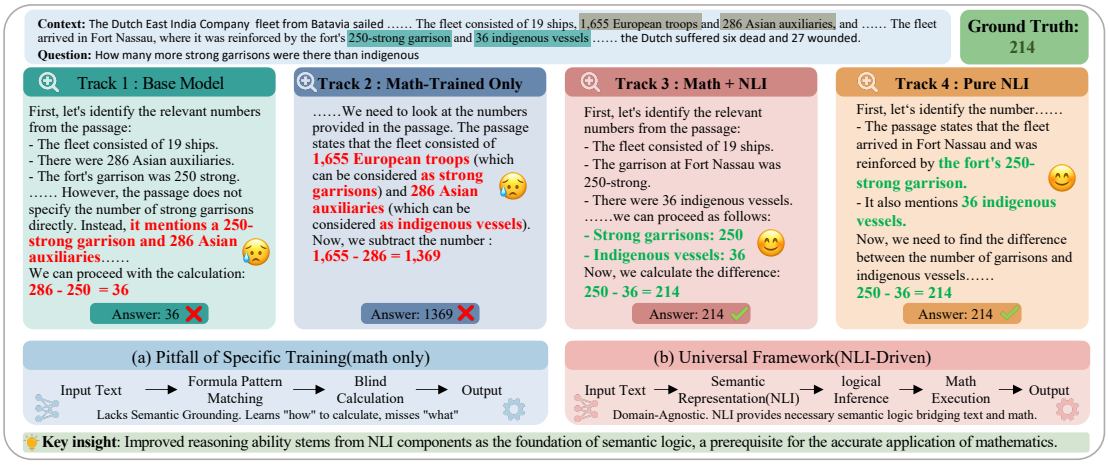

Figure 3: **Case study.** Representative examples comparing Base, math-only post-training, and NLI-based post-training. NLI training tends to improve text-to-constraints decomposition and spec-consistent control flow.

Model Scale	Training Data / Condition	Math		Code		Reasoning		Avg.
		GSM8k	Math500	HumanEval	MBPP	MMLU	DROP	Imp.
7B	Base	86.1	63.2	82.3	63.2	74.4	65.5	–
	NLI_ALL (Correct)	**90.1**$_{\uparrow 4.0}$	**66.4**$_{\uparrow 3.2}$	**85.4**$_{\uparrow 3.1}$	**64.8**$_{\uparrow 1.6}$	**74.8**$_{\uparrow 0.4}$	**67.4**$_{\uparrow 1.9}$	**↑2.2**
	Random Labels	86.3$_{\uparrow 0.2}$	62.4$_{\downarrow 0.8}$	83.5$_{\uparrow 1.2}$	64.2$_{\uparrow 1.0}$	74.4$_{0.0}$	66.2$_{\uparrow 0.7}$	↑0.4
	Swap (Inverted)	86.7$_{\uparrow 0.6}$	63.0$_{\downarrow 0.2}$	82.7$_{\uparrow 0.4}$	63.4$_{\uparrow 0.2}$	74.5$_{\uparrow 0.1}$	66.3$_{\uparrow 0.8}$	↑0.3

Table 3: **Sanity checks on label semantics.** Results are reported for **Qwen-2.5-7B-Instruct** under identical training settings. Within each metric column, the **best** result is shown in **bold**, and the second best is underlined. Performance changes relative to the Base model are indicated by subscript-style green upward arrows (↑) for improvements and red downward arrows (↓) for degradations. Correct-label NLI training yields substantially larger gains than control conditions with randomized or inverted labels.

length (Premise+Hypothesis). Specifically, we compute the token length of each input and split the dataset by the mean length (μ): **Long** ($> \mu$) and **Short** ($\leq \mu$). Table 2 reports the corresponding **NLI_SHORT** and **NLI_LONG** variants alongside corpus ablations and baselines. Overall, longer NLI inputs tend to yield larger gains on math (especially MATH500) and HumanEval, while the effect can be task-dependent (e.g., some code/general reasoning metrics).

2. Logic vs. Format: Table 3 compares correct-label NLI training against control variants with randomized or inverted labels (keeping the same output format). Random or inverted labels yield much smaller gains than correct labels, suggesting that improvements come from learning label semantics (logic) rather than memorizing a format.

5 Conclusion

We present a **logic-first** route from language to general reasoning via **verifiable NLI training**. By casting NLI as a generative RL objective, we observe a clear **NLI leap** that resolves the NLI paradox and transfers to mathematics, code generation, and general reasoning under scale-matched, comparable training budgets, often more robustly than math-only post-training. We argue that NLI is an effective training signal because it exercises reusable inferential primitives; empirically, closing the NLI gap and restoring the easy–hard separation (Figure 1) correlate with more compact solutions at matched correctness (Figure 2) and improved text-to-constraints reasoning in our case study (Figure 3). We hope this work encourages further study of verifiable, language-level supervision as a practical path to cross-domain reasoning transfer.

Limitations

While NLI training significantly boosts general reasoning and foundational math skills, we acknowledge certain limitations. First, for highly specialized, high-difficulty domain tasks—such as

Olympiad-level mathematics or code generation involving obscure libraries—pure NLI training may not fully replace domain-specific knowledge injection. NLI processprovides the logical skeleton, but expert-level tasks often require a deep reservoir of domain-specific facts. Second, our experiments are primarily conducted on the Qwen-2.5 series of mid-sized models (3B and 7B). While we hypothesize that our findings hold for larger scales, comprehensive validation on 70B+ parameter models remains future work. Finally, although we utilized long Chain-of-Thought NLI data, automatically generating high-quality, high-complexity NLI samples remains a challenge. Future research could explore using the model itself to generate adversarial examples for self-play reinforcement learning.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *Preprint*, arXiv:1508.05326.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Preprint*, arXiv:1812.01193.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA. Association for Computational Linguistics.

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, and 39 others. 2021. Evaluating large language models trained on code.

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. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 179 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv*, abs/2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *Preprint*, arXiv:1903.00161.

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, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey. *Preprint*, arXiv:2212.10403.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Yu Li, Zhuoshi Pan, Honglin Lin, Mengyuan Sun, Conghui He, and Lijun Wu. 2025. Can one domain help others? a data-centric study on multi-domain reasoning via reinforcement learning. *Preprint*, arXiv:2507.17512.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023a. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023b. Let's verify step by step. *Preprint*, arXiv:2305.20050.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *Preprint*, arXiv:1902.01007.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. *Preprint*, arXiv:1910.14599.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Adam Poliak. 2020. A survey on recognizing textual entailment as an nlp evaluation. *Preprint*, arXiv:2010.03061.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S Dhillon, and Sanjiv Kumar. 2024. Two-stage llm fine-tuning with less specialization and more generalization. *Preprint*, arXiv:2211.00635.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. 2025. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *Preprint*, arXiv:2506.14245.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. *Preprint*, arXiv:1704.05426.

Kai Yan, Yufei Xu, Zhengyin Du, Xuesong Yao, Zheyu Wang, Xiaowen Guo, and Jiecao Chen. 2025. Recitation over reasoning: How cutting-edge language models can fail on elementary school-level reasoning problems? *Preprint*, arXiv:2504.00509.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. *Preprint*, arXiv:2502.03373.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *ArXiv*, abs/2503.14476.

A Appendix

A.1 Training Implementation Details

Hyperparameters. We train all models using Group Relative Policy Optimization (GRPO). We use the same configuration for both Qwen-2.5-3B-Instruct and Qwen-2.5-7B-Instruct to ensure a fair comparison. The reference model is the initial SFT model (Qwen-2.5-Instruct) frozen during training.

GRPO Hyperparameters (Appendix)

Hyperparameter	Value
Group Size (G)	16
KL Coefficient (β)	0.04
Learning Rate	1e-6
LR Scheduler	Cosine
Global Batch Size	128
Max Sequence Length	2048
Reward Function	Binary (Correctness) + Format
Training Steps	500
Optimizer	AdamW

Training Setup. All models were trained on a single node with 8 NVIDIA A100 GPUs. We use data-parallel training with synchronized updates across all devices.

Reward Function. Our reward function consists of two components: 1. **Correctness Reward (r_{acc}):** A binary reward of +1 if the predicted NLI label (Entailment, Neutral, or Contradiction) exactly matches the ground truth, and

0 otherwise. 2. **Format Reward (r_{fmt}):** A small penalty or bonus to enforce the XML structure `<reasoning>...</reasoning><answer>...</answer>`. In our experiments, we found that Qwen-2.5 models adhere to format well, so this term has a low weight (0.1).

A.2 Prompt Templates

NLI Training Prompt. During training, we wrap each NLI instance in the following template to encourage Chain-of-Thought reasoning. The model is trained to generate the content within the XML tags.

NLI Training Prompt

System: You are a helpful logical reasoning assistant. Given a premise and a hypothesis, determine the logical relationship between them (Entailment, Contradiction, or Neutral). First, analyze the semantic and logical details step-by-step in a `reasoning` block. Then, output the final label in an `answer` block.
User: Premise: {premise} Hypothesis: {hypothesis} Please analyze the relationship.
Assistant: <reasoning> {Model generates CoT here} </reasoning> <answer> {Label} </answer>

Example: Converting raw NLI into a trainable prompt. Below is a representative example illustrating how we reformat an original NLI pair into a unified prompt-and-response format to enable large-scale training (e.g., our 1000k NLI pool).

Trainable NLI Prompt Example

```
Premise: She had thrown away her cloak and
tied her hair back into a topknot to keep
it out of the way.
Hypothesis: She tied her hair up with a
ribbon
You are an NLI classifier. Decide whether
the hypothesis is entailment, neutral, or
contradiction with respect to the premise.
You FIRST think about the reasoning process
as an internal monologue and then provide
the final answer. The reasoning process
MUST BE enclosed within <think> </think>
tags.
The final answer MUST BE put in \boxed{}.
Required format:
<think> ... </think>
\boxed{entailment|neutral|contradiction}
```

Evaluation Prompts. For downstream benchmarks (Math, Code, General Reasoning), we use standard zero-shot prompts. For GSM8K and MATH500, we simply ask the model to "Solve the following problem step by step" to trigger its learned reasoning capabilities.

A.3 Dataset Construction Details

Data Sources. Our training dataset is constructed from three primary NLI corpora:

- **SNLI (Stanford NLI):** Sourced from image captions, representing grounded, everyday reasoning.
- **MNLI (Multi-Genre NLI):** Covering various genres like fiction, government reports, and telephone speech.
- **ANLI (Adversarial NLI):** Hard examples constructed to fool simpler models, providing a stronger signal for robust logic.

Rationale Augmentation. Since original NLI datasets only contain labels, we augment them with reasoning paths using GPT-4o. We construct a few-shot prompt with high-quality logical explanations and query GPT-4o to generate rationales for 100k sampled instances.

Quality Filtering. To ensure the "Reasoning Kernel" is clean, we apply a consistency filter: we verify if the rationale generated by the teacher model leads to the correct ground-truth label. If the teacher's reasoning is ambiguous or concludes with a wrong label, the sample is discarded. This process yields our final **NLI-Reasoning** dataset.

A.4 Additional Qualitative Examples

We provide additional qualitative examples in the released artifacts. The main paper includes the primary case study figure in Section 4.

A.5 Additional Analysis Notes

Relational vs. answer-centric supervision (interpretive). Under verifiable RL, NLI provides a *relational* signal (correctness depends on the premise–hypothesis relation), whereas many math settings provide an *answer-centric* signal (correctness depends on an exact final answer). A plausible hypothesis is that NLI-RL induces a stronger relational inductive bias that is reusable for constraint propagation and consistency checking; we leave mechanistic measurements (e.g., representation similarity or probing) to future work.

Why longer NLI inputs can help (interpretive). Longer NLI inputs tend to contain more nested conditionals and long-range dependencies. Training on such instances may encourage maintaining logical consistency over longer contexts, which can

Model	ANLI	MNLI-mm	MNLI-m	SNLI	Weighted Avg.
Qwen2.5-3B-Instruct	56.3	80.7	80.7	79.8	78.03
Qwen2.5-7B-Instruct	60.8	83.8	83.0	81.3	80.57
deepseek-r1	78.1	81.0	82.0	81.9	81.28
deepseek-ai_DeepSeek-V3	76.0	82.8	83.9	79.8	81.58
Qwen2.5-72B-Instruct	73.2	85.9	85.7	82.0	83.42
gpt-4o	75.4	83.9	83.2	87.6	83.98
Qwen2.5-14B-Instruct	70.3	86.7	86.4	84.8	84.45
gemini-2.5-pro	80.5	84.1	84.1	86.5	84.50
gemini-2.5-pro-nothinking	81.2	84.6	85.3	85.2	84.66
o1-mini-2024-09-12	79.8	87.7	88.2	85.8	86.52
DeBERTa-v3-large	70.2	90.8	91.2	92.2	89.32
3B (ours)	**90.5**	**90.4**	**90.7**	**91.5**	**90.83**
7B (ours)	**93.2**	**93.8**	**93.6**	**94.1**	**93.77**

Table 4: **Figure 1 (Top) underlying scores.** Filtered accuracy (%) on valid samples across NLI datasets. Models are sorted by the weighted average (weighted by evaluation set sample counts). Ours are shown in bold.

transfer to multi-step word problems and longer code-generation tasks. We treat this as an explanation rather than a mechanistic claim.

A.6 Token Counting Protocol for Figure 2

Figure 2 compares response length (generated tokens) between pure NLI post-training and math-only post-training under a matched protocol:

- **Tasks:** We aggregate four downstream benchmarks used throughout the paper: GSM8K, MATH500, HumanEval, and MBPP.
- **Matched correctness subset:** For each benchmark, we keep only those evaluation instances where *both* compared models are correct under the same evaluation metric (accuracy for GSM8K/MATH500; pass@1 for HumanEval/MBPP). This controls for the confound that incorrect solutions can be arbitrarily short or long.
- **Decoding:** We use the same decoding settings for both models, including the same max generation length and stopping criteria.
- **Token counting:** We count the number of generated tokens in the model *completion* (excluding the prompt), using the tokenizer corresponding to the evaluated base model.

A.7 Additional Tables

All comprehensive results are included in the main paper (Table 2). We omit duplicate result tables in the appendix to save space, but provide supplementary tables that support figures in the main text. Figure 1 (A and B)'s scores are in Table 4

A.8 Code and Data Availability

Our code and reconstructed training data will be released upon paper acceptance.

A.9 AI Assistance Disclosure

We used an AI assistant only to help with code-writing and English polishing. All core experiments and data processing were performed by the authors.