Correct Reasoning Paths Visit Shared Decision Pivots

Dongkyu Cho[1,*] **Amy B. Z. Zhang**[2], **Bilel Fehri**[2], **Sheng Wang**[3]
Rumi Chunara[1], **Hengrui Cai**[4], **Rui Song**[2†]
[1] New York University [2] Amazon [3] University of Washington [4] UC Irvine
dongkyu.cho@nyu.edu, {amybzz, bffehri}@amazon.com, swang@cs.washington.edu, rumi.chunara@nyu.edu, hengrc1@uci.edu, ruisong@amazon.com

Abstract

Chain-of-thought (CoT) reasoning exposes the intermediate thinking process of large language models (LLMs), yet verifying those traces at scale remains unsolved. In response, we introduce the idea of decision pivots—minimal, verifiable checkpoints that any correct reasoning path must visit. We hypothesize that correct reasoning, though stylistically diverse, converge on the same pivot set; incorrect ones violate at least one pivot. Leveraging this property, we propose a self-training pipeline that (i) samples diverse reasoning paths and mines shared decision pivots, (ii) compresses each trace into pivot-focused short-path reasoning using an auxiliary verifier, and (iii) post-trains the model using its self-generated outputs. The proposed method aligns reasoning without ground truth reasoning data or external metrics. Experiments on standard benchmarks such as LogiQA, MedQA, and MATH500 show the effectiveness of our method.

1 Introduction

Large language models (LLMs) can now solve diverse problems when prompted to articulate their *chain-of-thought* (CoT) reasoning [38]. Yet these reasoning paths are often redundant, self-contradictory, or logically unsound, even when they culminate in the right answer. Current training pipelines such as reinforcement learning from human feedback (RLHF) [28] evaluate entire responses, providing only coarse signals that do little to supervise the underlying reasoning process. As a result, we still lack a scalable way to verify and train high-quality explanations without resorting to expensive human annotation. We claim that many reasoning tasks are governed by decision pivots: minimal, verifiable checkpoints (e.g., keyword, key logic) that every sound explanation must traverse. Echoing the proverb "All roads lead to Rome", correct CoTs succeed for the same reason: they visit the shared pivot set; whereas incorrect paths fail in many different ways, omitting or contradicting at least one pivot. Exploiting this observation, we propose a self-training method, ROMA (Reasoning Optimization via Multi-path Aggregation), that aligns a model's reasoning without utilizing human-annotated rationales or handcrafted metrics.

Our method operates in three stages. First, we draw K diverse CoT reasoning paths for each input sample, bootstrapping the pivots shared by the successful reasoning paths [41]. Second, we merge multiple reasoning paths into a *short-path reasoning* (SPR) that focuses on the pivots and removes redundant steps. Finally, we post-train the model with RLHF [31], treating SPRs as preferred positives and individual reasoning paths as negatives. The entire loop is model-driven and repeatable at scale [32]. Experiments on LogiQA[22], MedQA[16], and MATH500[13] yield simultaneous gains in solving downstream task accuracy and in reasoning quality, while confirming the method's generality across commonsense, expert (e.g., medical), and math domains.

[*]Paper written during Dongkyu's internship at Amazon

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

Contributions.

- We formalize *decision pivots* and show that correct reasoning concentrates on a compact pivot set.
- We propose ROMA, a bootstrap→rewrite framework that aggregates K paths and, via a fine-tuned verifier, distills a compact, pivot-consistent short-path reasoning.
- We build a self-training pipeline that leverages the short-path data (e.g., with DPO) to improve both task accuracy and explanation quality across benchmarks (LogiQA, MedQA, MATH500).
- We introduce a decision pivot-based reasoning metric and show it complements general CoT metrics (e.g., ROSCOE) for preference-data selection.

2 Preliminaries & Positioning

Post-training. Post-training adapts a pre-trained model to produce human-aligned outputs. Typically, this involves supervised fine-tuning (SFT) on curated human-aligned data, followed by reinforcement learning (RL) [28] on preference-labeled examples. A key challenge is constructing reliable preference data. One line of work derives labels from explicit metrics [9, 10, 30] or from model-based feedback [3, 17, 19, 23] (LLM-as-judge). RLVR (reinforcement learning with verifiable rewards) removes human preference signals entirely by using programmatic verifiers to assign binary rewards—effective in domains like math or coding, where correctness is objectively checkable [6]. Extensions seek to broaden this to more open ended reasoning [15, 24], but still require task-specific verifiers. **Self-Training.** In parallel, self-training approaches leverage the model's own outputs as training data. STaR [41] bootstraps from reasoning paths that yield correct answers; Self-Refine [25] uses iterative self-critique; and other works exploit self-consistency as a proxy reward for RL [32], build self-editing/verifying systems [42], or study the generation–verification gap theoretically [34]. These methods improve reasoning quality by filtering or refining full reasoning paths, but generally treat each correct reasoning in isolation. In contrast, our approach intersects multiple successful reasoning paths to identify *decision pivots*—minimal, verifiable checkpoints that all correct solutions share—and rewrites them into concise short-path reasoning before preference optimization. This distillation step targets reasoning faithfulness and verifiability without relying on handcrafted verifiers or general reasoning metrics [9].

Reasoning Models. Chain-of-thought (CoT) prompting [39] established that models reason more effectively when they articulate intermediate steps, with later work showing that even a simple prompt (*"Let's think step by step"*) can elicit this behavior [18]. Research since has explored richer reasoning supervision: scratchpad-style intermediates [27], self-consistency via majority voting over multiple reasoning paths [36], and targeted data curation—aggregating reasoning corpora [8], actively selecting informative queries [21], or varying difficulty adaptively [7]. Recent trends also include compressing CoTs for efficiency or readability, but these typically focus on token length reduction [4, 12, 26] rather than identifying core, verifiable reasoning elements. Our method differs in that it uses multi-path intersection to recover a shared, verifiable reasoning core, rewrites it into a short path form, and uses this distilled form as a high-quality supervision signal—bridging the gap between raw reasoning data generation and verifiable, minimal-step reasoning refinement.

3 Problem Formulation: In search of Short-Path Reasoning

In this section, we formally introduce the problem setting. Specifically, we consider the general setting where, for each input–output pair (x, y), $x \in \mathcal{X}$ is a question and $y \in \mathcal{Y}$ is its ground-truth answer (e.g., a class label). A pre-trained reasoning model $p_\theta(r, \hat{y} \mid x)$ [2, 40] jointly generates a chain-of-thought reasoning path $r = (t_1, t_2, \ldots, t_l)$ of length l and a prediction $\hat{y}$ given x. When convenient we use the induced conditionals $p_\theta(y \mid x, r)$ (label probability given a trace) and $p_\theta(r \mid x, y) \propto p_\theta(r, y \mid x)$ (reasoning path distribution given the label). We denote the unobserved ground truth reasoning as $r^* = (t_1^*, t_2^*, \ldots, t_d^*)$ with $d \ll l$. The dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ contains N supervised examples.

Here, the main goal is to produce high-quality reasoning that closely resembles the ground-truth reasoning r^*. More specifically, we aim to obtain reasoning that is (1) correct with respect to the final answer, (2) concise in containing only the information necessary to support that answer, and (3) faithful in the sense of being causally responsible and correct for the model's prediction. However,

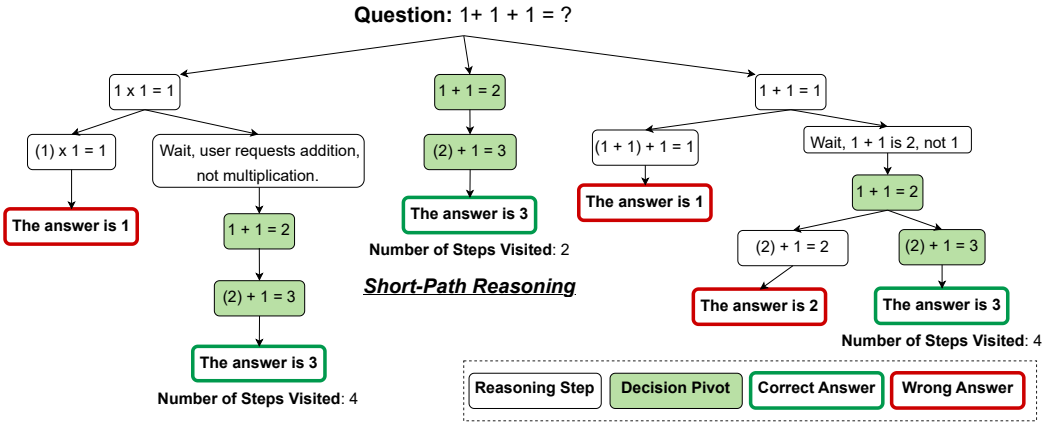


Figure 1: A model's chain-of-thought reasoning may take various paths to reach a final decision, naturally visiting redundant or incorrect thought steps. However, such errors are not easily captured by existing methods. Instead, we introduce the concept of decision pivots, which are a set of key information that a model's reasoning path must visit to reach a certain decision. In this sense, we aim to generate a concise, factual reasoning path that focuses on the decision-pivots, which we denote as Short-Path Reasoning (SPR).

several factors make this problem challenging. First, the absence of ground-truth reasoning. While the correct label y is observed, the true reasoning process leading to the label is not. Hence, there is no direct supervision for which intermediate reasoning steps are valid. Second, verifying intermediate reasoning steps is difficult, especially in open-ended reasoning tasks (e.g., general language tasks). Hence, individual reasoning steps are hard to verify automatically, while human verification is costly and lacks scalability. Formally, the objective can be stated as finding a reasoning trace r that

$$\max_{r} \; p_\theta(y \mid x, r) \tag{1}$$

subject to r being as concise as possible while preserving correctness. This requires identifying and retaining only the key information necessary to support the answer, despite the lack of explicit supervision and step-level verifiability [44].

Limitations of Prior Approaches. Several existing paradigms attempt to address this challenge but face inherent limitations in this general setting. Self-training methods such as STaR [41] bootstrap from model-generated reasoning that yield correct answers, but treat each reasoning path independently and do not explicitly seek minimality or commonality across multiple successful paths. Self-consistency [36] aggregates predictions from multiple sampled reasoning paths, improving answer accuracy but producing no refined intermediate reasoning for training. RLVR [6, 42, 43] uses verifiable rewards to supervise reasoning, but is restricted to domains where intermediate verification is feasible, such as math and code, and does not generalize to open-ended reasoning. Other metric-based or LLM-as-judge feedback methods [3, 9, 23] depend on either handcrafted metrics or opaque model judgments, which do not reflect the task. In other words, existing metrics do not reliably capture whether the reasoning truly contains only the information necessary for the decision, which we empirically show in Section 5. In the following section, we seek to address the general problem of producing reasoning in the absence of ground-truth reasoning and automated verifiers.

4 Reasoning Optimization via Multi-path Aggregation

In this section, we present a self-training framework that bootstraps a concise, human-aligned reasoning path for self-improvement of the model's reasoning capability. The key idea of our method lies in the concept of *decision pivots*, which are key logics / key words that are shared across reasoning paths that reach the same decision (See Figure 1). Under this assumption, our method bootstraps a pivot-focusing CoT reasoning by distilling multiple reasoning paths into a concise reasoning path that yields the correct prediction (i.e., short-path reasoning), then feeds the *short-path* reasoning back into the model to self-improve its reasoning capability, boosting its generalization.

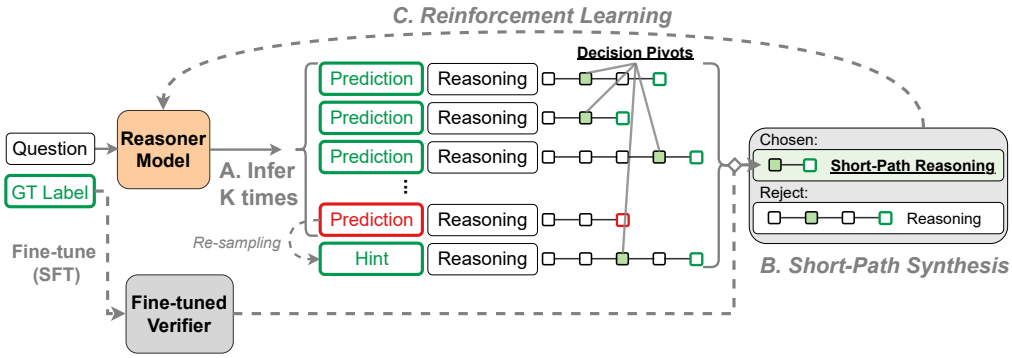


Figure 2: We present a novel self-training framework ROMA that leverages the concept of decision-pivots. Our self-training framework works in 3 stages. Given a question: (A) model produces multiple PREDICTION + REASONING pairs, ensuring we collect K reasoning paths using re-sampling. (B) Then, a fine-tuned verifier synthesizes a short-path reasoning that focuses on shared decision-pivots, generating a preference data pair. (C) Use the generated data for Reinforcement Learning (i.e., preference learning with DPO). This process can be repeated as a self-training loop that improves the model's reasoning capabilities.

Notation. Let $x \in \mathcal{X}$ be an input (e.g., a question) and $y \in \mathcal{Y}$ its ground-truth label. We use a pre-trained reasoning model $p_\theta(r, \hat{y} \mid x)$ that, given x, jointly produces a reasoning path $r = (t_1, \ldots, t_l)$ and a prediction $\hat{y}$ [2, 40]. We write $p_\theta(y \mid x, r)$ for the model's conditional over labels given a reasoning, and $p_\theta(r \mid x, y) \propto p_\theta(r, y \mid x)$ for the conditional over the reasoning given the label. The training corpus is $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ of size N. For each input-label pair (x_i, y_i), define a candidate set of joint samples $\mathcal{S}_i = \{(r_{ik}, \hat{y}_{ik})\}_{k=1}^K$ drawn from $p_\theta(\cdot \mid x_i)$, and the successful subset $\mathcal{R}_i^+ = \{\, r_{ik} : \hat{y}_{ik} = y_i \,\}$. Separate from the reasoner p_θ, we adopt a fine-tuned verifier $v_\psi : (x, \mathcal{R}) \mapsto \hat{r}$ trained on $\mathcal{D}$ (namely using LoRA adapters [14] on p_θ) synthesizes a short path reasoning $\hat{r}_i = v_\psi(x_i, \mathcal{R}_i^+)$.

4.1 Stage A: Multi-sample bootstrapping

For each (x_i, y_i), sample K pairs

$$\mathcal{S}_i = \big\{(r_{ik}, \hat{y}_{ik}) \sim p_\theta(\cdot \mid x_i)\big\}_{k=1}^K, \qquad \mathcal{R}_i^+ = \{\, r_{ik} : \hat{y}_{ik} = y_i \,\} \tag{2}$$

If $\mathcal{R}_i^+$ is empty, resample traces from $p_\theta(r \mid x_i, y_i)$ to ensure coverage (See Step A in Figure 2). This step explores diverse reasoning paths while keeping supervision verifiable via the paired prediction; conditioning on y_i functions as a fallback to avoid empty pools. Intuitively, the two sources of reasoning widen the coverage of the model's reasoning. Zero-shot correct reasoning paths reveal the model's canonical, high-probability pivot sets (i.e., what the model already trusts), while *guided* traces (re-sampled after revealing the hint y_i on the initial incorrect prediction [1, 41]) allow the model to route around its default heuristics and adopt novel paths that still rationalize y_i.

4.2 Stage B: Short-path synthesis with Fine-tuned Verifier

Given a candidate pool of multiple reasoning paths that reach the correct decision $\mathcal{R}_i^+$, the verifier v_ψ aggregates overlapping content across candidates and produces a refined short-path reasoning that focuses on the decision pivots, and preserves the correct decision (See Step B in Figure 2):

$$\hat{r}_i \;=\; v_\psi(x_i, \mathcal{R}_i^+) \quad \text{subject to} \quad \arg\max_y p_\theta(y \mid x_i, \hat{r}_i) = y_i. \tag{3}$$

The rewriting aggregates shared decision-pivots while discarding redundant reasoning steps; the correctness check guarantees that the resulting explanation leads to the correct decision. Please note that the verifier v_ψ is used only to synthesize $\hat{r}_i$ and is not our final model.

Discussion: Fine-tuned Verifiers. An important aspect of our method is adopting a fine-tuned verifier v_ψ to rewrite short-path reasoning, separate from the reasoner p_θ that produces a pool of K

candidate traces. This way, the reasoner p_θ remains domain-agnostic, while a lightweight verifier v_ψ (p_θ added with LoRA trained on $\mathcal{D}$) acquires domain priors to synthesize short-path traces during training and is discarded at test time. The motivation is the specialization–generalization trade-off: fine-tuning on $\mathcal{D}$ can inject domain knowledge but tends to deteriorate general reasoning ability [5, 24, 37]. By isolating this specialization in v_ψ, we preserve the general capability of p_θ while leveraging the verifier's domain priors to filter spurious steps, align traces to decision pivots, and consolidate them into faithful short paths. The cost is minimal—v_ψ is obtained via small LoRA adapters on the same backbone—and, empirically, verifier-guided rewriting provides stronger learning signals than self-rewriting with p_θ (see Figure 4 and Table 1).

4.3 Stage C: Pairwise preference optimization (DPO)

Lastly, we train the model p_θ with the short-path reasoning we have obtained. Specifically, we treat $\hat{r}_i$ as chosen and each $r \in \mathcal{R}_i^+ \setminus \{\hat{r}_i\}$ as a rejected sample for preference learning, forming

$$\mathcal{P}_i = \big\{(\hat{r}_i, r) : r \in \mathcal{R}_i^+ \setminus \{\hat{r}_i\}\big\}. \tag{4}$$

Using this, we update the main model's conditional $p_\theta(r \mid x, y)$ with DPO [31] against a frozen reference $p_{\text{ref}}(r \mid x, y)$:

$$\begin{aligned}\mathcal{L}_{\text{DPO}}(\theta) = -\sum_{i=1}^{N} \sum_{(r^+, r^-) \in \mathcal{P}_i} \log \sigma\Bigg(\beta \Big[&\underbrace{\log p_\theta(r^+ \mid x_i, y_i) - \log p_\theta(r^- \mid x_i, y_i)}_{\text{policy preference}} \\ &- \underbrace{\log p_{\text{ref}}(r^+ \mid x_i, y_i) + \log p_{\text{ref}}(r^- \mid x_i, y_i)}_{\text{implicit KL correction}} \Big]\Bigg),\end{aligned} \tag{5}$$

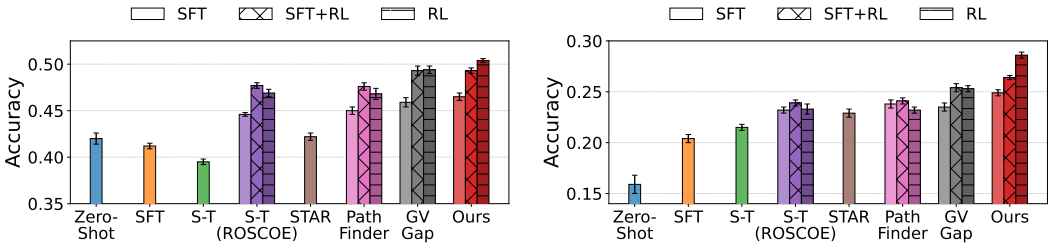
with temperature $\beta > 0$ and logistic σ. (See Step C in Figure 2) This preference learning increases the relative likelihood of the synthesized short-paths while regularizing toward the reference, aligning the model to favor concise, pivot-focused reasoning.

5 Experiments

5.1 Experimental Setting

Benchmarks. We validate our approach across three public benchmarks to demonstrate cross-domain effectiveness. LOGIQA [22] is a general question-answer dataset that assesses logical reasoning with expert-written multiple choice questions (MCQs) where the models are evaluated mainly by accuracy. MEDQA [16] is a medical MCQ benchmark built from professional licensing exams (e.g., USMLE), designed to evaluate clinical knowledge and reasoning based on accuracy. Lastly, MATH500 [13] is a mathematical reasoning benchmark commonly used to assess a model's reasoning capabilities. On these benchmarks, we compare different types of self-training methods that utilize their own generated outputs to improve their reasoning capability, which in turn improves a model's downstream task performance [10, 34, 41]. In all experiments, we used a held-out set of test data for evaluation.

Architectures and protocol. We evaluate our method on variants from the Qwen-3 family (0.6/1.7/4/8B/14B) [40], each of which already demonstrates strong zero-shot reasoning ability. Unless mentioned, the default model is set as the Qwen-3-0.6B model, considering the generation cost/latency of self-training (see Table 9). We further use several larger models (e.g., Claude 3 [2], LLama-3 [11]. Every experiment is repeated five times with independent random seeds, and we report the mean performance together with the corresponding standard errors. Details regarding the hyperparameters are presented in the sequel. We assess the effectiveness of our method in two aspects. (1) Downstream Task: Using the generated reasoning, we post-train the model and then assess the trained model on the downstream task (Table 1). (2) Qualitative analysis: Assess the output reasoning using general metrics e.g., ROSCOE[9] (Table 2). Further detail regarding the experimental setting is included in Section A.3



(a) Results in LogiQA

(b) Results in MedQA

Figure 3: Comparison of self-training results on LogiQA and MedQA. Our proposed method provides large self-improvement gains in domains (e.g., MedQA – healthcare, LogiQA – general language) that are generally difficult to verify the generated reasoning, compared to math/coding domains (e.g., MATH500 – math). S-T (ROSCOE) refers to self-training with ROSCOE-filtered reasoning. The error bars indicate the standard error over 5 runs.

Table 1: Accuracy on LogiQA, MedQA, and MATH500 using various types of generated reasoning, across different post-training methods (e.g., SFT, RL). We report the standard error across 5 runs.

Method	**Post-Training**	**LogiQA**	**MedQA**	**MATH500**
Zero-Shot	–	$0.420_{\pm 0.006}$	$0.159_{\pm 0.009}$	$0.636_{\pm 0.006}$
Trained without Reasoning	SFT	$0.412_{\pm 0.003}$	$0.204_{\pm 0.004}$	$0.649_{\pm 0.004}$
Self-Trained (Zero-Shot)	SFT	$0.395_{\pm 0.003}$	$0.215_{\pm 0.003}$	$0.655_{\pm 0.005}$
	SFT	$0.446_{\pm 0.002}$	$0.232_{\pm 0.003}$	$0.673_{\pm 0.003}$
Self-Trained (ROSCOE [9])	SFT+RL	$0.477_{\pm 0.003}$	$0.239_{\pm 0.003}$	$0.681_{\pm 0.004}$
	RL	$0.469_{\pm 0.004}$	$0.233_{\pm 0.005}$	$0.677_{\pm 0.005}$
STAR [41]	SFT	$0.422_{\pm 0.004}$	$0.229_{\pm 0.004}$	$0.653_{\pm 0.005}$
	SFT	$0.450_{\pm 0.004}$	$0.238_{\pm 0.004}$	$0.668_{\pm 0.005}$
PATHFINDER [10]	SFT+RL	$0.476_{\pm 0.004}$	$0.241_{\pm 0.003}$	$0.679_{\pm 0.005}$
	RL	$0.468_{\pm 0.006}$	$0.232_{\pm 0.003}$	$0.671_{\pm 0.005}$
	SFT	$0.459_{\pm 0.005}$	$0.235_{\pm 0.004}$	$0.670_{\pm 0.003}$
GV-GAP [33]	SFT+RL	$0.493_{\pm 0.005}$	$0.254_{\pm 0.004}$	$0.688_{\pm 0.004}$
	RL	$0.494_{\pm 0.004}$	$0.253_{\pm 0.003}$	$0.686_{\pm 0.004}$
	SFT	$0.462_{\pm 0.003}$	$0.234_{\pm 0.003}$	$0.669_{\pm 0.004}$
Ours (w/o verifier)	SFT+RL	$0.495_{\pm 0.004}$	$0.251_{\pm 0.002}$	$0.684_{\pm 0.003}$
	RL	$0.489_{\pm 0.003}$	$0.250_{\pm 0.003}$	$0.681_{\pm 0.003}$
	SFT	$0.465_{\pm 0.004}$	$0.249_{\pm 0.003}$	$0.671_{\pm 0.004}$
Ours	SFT+RL	$0.493_{\pm 0.003}$	$0.264_{\pm 0.002}$	$0.686_{\pm 0.002}$
	RL	$\mathbf{0.504}_{\pm 0.002}$	$\mathbf{0.286}_{\pm 0.003}$	$\mathbf{0.692}_{\pm 0.003}$

5.2 Experimental Results

We aim to answer the following questions to assess the effectiveness, efficiency, and generality of our proposed method: **(1)** Can it generate high-quality reasoning that improves (e.g., after training or zero-shot) generalization? (Table 1) **(2)** How do key design components—bootstrapping, rewriting—contribute to its effectiveness? Can the proposed self-training method provide good learning signals continually, improving the model on each loop? (Table 3) **(3)** Do decision pivots truly exist, and are they retrievable? (Table 4)

Main Results Across general logic (LogiQA), clinical (MedQA), and math (MATH500) benchmarks, we see a repeated pattern (see Table 1): the more training directs supervision onto the *decision pivots*, the better the model performs. When we fine-tune only on labels or reuse zero-shot reasoning data, accuracy stalls or even drops, because the model is not guided to distinguish which reasoning steps to keep and which to omit (for LogiQA, $0.420 \rightarrow 0.412$ and 0.395). On the other hand, filtering the self-generated reasoning data with a general metric like ROSCOE [9] helps, but its effect plateaus because "general quality" misses task-specific pivots (e.g., LogiQA SFT+DPO ≈ 0.477). By

compressing several successful traces into one *short-path reasoning* (SPR), supervision concentrates on those pivots paired with preference optimization (DPO); this reliably surpasses metric-filtered baselines across tasks (LOGIQA SPR+DPO 0.495; MEDQA ≈ 0.25; MATH500 0.684). The largest gains appear when a domain fine-tuned verifier v_ψ helps synthesize the short path (vs. using the reasoner p_θ). Intuitively, the verifier brings in the right priors, prunes spurious steps, and sharpens the learning signal through verification. The effect scales with domain knowledge needs—largest on MEDQA (DPO 0.316, an improvement of $+0.066$ over SPR without a verifier), clear on LOGIQA (0.518, $+0.029$), and smaller yet consistent on already-verifiable math (MATH500 0.702, $+0.021$). In short, our pivot-centric rewriting method (ROMA), especially with a verifier, yields both better accuracy and more faithful, compact explanations than label-only training, naive self-training, or generic-metric selection.

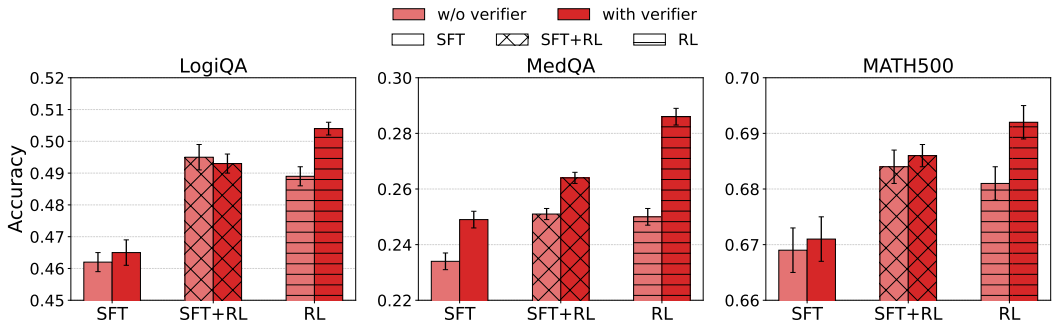


Figure 4: Effect of the fine-tuned verifier on downstream accuracy across post-training regimes. We compare our method with vs without a verifier on LogiQA, MedQA, and MATH500. The verifier yields consistent gains and is most impactful in the expert MedQA domain, supporting our claim that a domain-tuned verifier synthesizes higher-quality, pivot-focused short-path reasoning. Improvements are present but smaller on MATH500 (where an external verification signal already exists) and modest on LogiQA, aligning with our hypothesis and analysis.

Effect of Fine-tuned Verifiers. Here, we show that using a fine-tuned verifier can synthesize higher-quality reasoning by incorporating the domain knowledge it has obtained through fine-tuning. In Table 1, we find that adding a fine-tuned verifier synthesizes higher-quality reasoning than omitting it (Also see Figure 4). Beyond the aggregate improvements across SFT, SFT+RL, and RL, the verifier works by filtering spurious steps, aligning traces to decision pivots, and injecting domain priors during consolidation—raising pivot density and widening decision margins. The effect tracks task characteristics: it is most pronounced on knowledge-intensive MEDQA (e.g., RL: $+0.036$, SFT: $+0.015$), smaller but consistent on MATH500 where an external correctness signal already exists (RL: $+0.011$), and modest on LOGIQA (SFT: $+0.003$; SFT+RL: -0.002; RL: $+0.015$) (Figure 4). In practice, we implement the verifier with LoRA adapters [14] for a favorable accuracy–compute trade-off; when domains shift, we retune the verifier to refresh its priors (Table 1).

Are General Reasoning Metrics sufficient for preference data selection? In this section, we investigate the effectiveness of existing reasoning metrics. We report three observations. **(1)** As model size scales, most ROSCOE facets are nearly flat or idiosyncratic—e.g., FAITHFULNESS stays in 0.872–0.902, INFORMATIVENESS (STEP) in 0.875–0.893, and FAITHFULNESS (WW) in 0.939–0.948—while downstream accuracy rises substantially (from 0.486 at 1.7B to 0.672 at 14B; cf. 0.552 at 0.6B and 0.664 at 8B; Table 2). **(2)** Cross-size correlations make the misalignment explicit (Figure 5): several scores are near-zero or negative with respect to accuracy (e.g., INFORMATIVENESS (CHAIN) $r \approx -0.50$, FAITHFULNESS (WW) $r \approx -0.40$, PERPLEXITY (CHAIN) $r \approx -0.46$), and the best positives are only modest (e.g., COHERENCE $r \approx +0.49$). **(3)** The resulting "flat-high" profile suggests ceiling effects that obscure the changes most predictive of correctness. However, this is an expected behavior. By design, ROSCOE emphasizes internal consistency and fluency of a chain, not whether its pivots land on the correct answer. Hence, selecting by a general metric like ROSCOE reliably removes obvious errors but does not prioritize decision-critical steps. As a result, selecting the preference data pair using ROSCOE scores yields modest, but inconsistent gains: it filters flawed or redundant traces yet fails to target what drives task competence. On the other hand,

Table 2: ROSCOE scores and the downstream LogiQA accuracy across model sizes.

Metric	0.6B	1.7B	4B	8B	14B
Faithfulness	0.902	0.872	0.885	0.892	0.888
Informativeness (Step)	0.893	0.875	0.886	0.889	0.883
Informativeness (Chain)	0.934	0.938	0.944	0.928	0.924
Faithfulness (WW)	0.948	0.941	0.944	0.940	0.939
Repetition (Word)	0.023	0.012	0.009	0.018	0.021
Repetition (Step)	0.020	0.012	0.014	0.016	0.019
Discourse Rep.	0.007	0.002	0.001	0.002	0.004
Coherence	0.004	0.001	0.002	0.001	0.052
Perplexity (Step)	0.002	0.005	0.006	0.004	0.005
Perplexity (Chain)	0.068	0.128	0.116	0.076	0.084
Grammar (Step)	0.901	0.891	0.889	0.894	0.890
Grammar (Step Max)	0.557	0.355	0.371	0.339	0.442
Accuracy	0.552	0.486	0.629	0.664	0.672

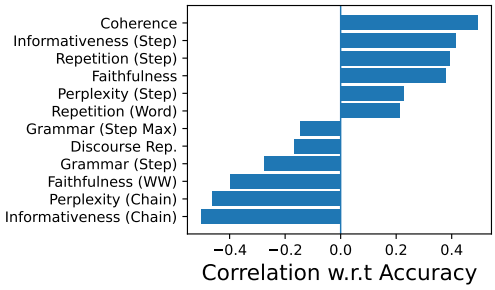


Figure 5: Pearson Correlation between ROSCOE scores and downstream LogiQA accuracy.

Table 3: Ablation study of our method. (Left) Varying the number of candidate reasoning paths K; (Right) varying the number of self-training loops.

(a) Effect of K (candidate pool size).

Method	K	LogiQA	MedQA	MATH500
Ours	1	0.437	0.252	0.649
Ours	3	0.499	0.273	0.680
Ours	5	0.504	0.286	0.692
Ours	7	0.509	0.292	0.695
Ours	10	**0.511**	**0.296**	**0.699**

(b) Effect of the number of self-training loops.

Method	Loops	LogiQA	MedQA	MATH500
Ours	1	0.504	0.286	0.692
Ours	2	0.522	0.290	**0.699**
Ours	3	0.525	0.294	0.683
Ours	4	**0.526**	0.297	0.694
Ours	5	0.522	**0.297**	0.693

aggregating multiple correct trajectories into *short-path reasoning* focuses supervision on decision pivots and—when paired with a lightweight verifier—delivers larger, more reliable improvements than metric-based selection.

Ablation Study. In this section, we provide the results of the ablation study on our method. Specifically, we study (1) the effect of K (the number of reasoning paths collected in the candidate pool), and (2) the number of self-training loops. **K (candidate pool size).** As shown in Table 3a, increasing K consistently improves performance across all benchmarks, with the largest gains occurring when moving from K=1 to K=3 (better pivot coverage and fewer single-path idiosyncrasies). Returns diminish beyond K=5, indicating that the verifier can consolidate a relatively small, diverse set of traces into an equally effective short path. In practice, K=7 recovers ≈95–100% of the K=10 peak while reducing sampling and verification cost, so we adopt K=7 as our default for the accuracy–compute trade-off. **Self-training loops.** Table 3b shows steady improvements from one to two loops as the model internalizes the synthesized short-path signals. A third loop often yields smaller but still positive gains; beyond that, improvements saturate and can slightly regress on math-heavy tasks (likely pivot-set saturation and mild distributional drift toward synthesized traces). We therefore use three loops in the main results, with early stopping on validation accuracy to avoid overfitting to the generated distribution. To sum up, the takeaways are as follows: (i) Larger candidate pools primarily help by increasing pivot diversity rather than sheer quantity; once the verifier sees enough distinct routes, additional traces add little value. (ii) Iterative self-training is beneficial but exhibits diminishing returns; modest loop counts (2–3) strike a good balance between quality and compute.

Table 4: Pivot retrieval rate (%) as a function of model size and number of zero-shot samples Q per prompt. A retrieval is counted when the generated reasoning visits (i.e., mentions) the ground truth decision pivot.

Model	Q=10	Q=100	Q=500
Qwen-3 0.6B	87%	89%	93%
Qwen-3 1.7B	85%	92%	95%
Qwen-3 4B	90%	95%	97%
Qwen-3 8B	94%	95%	97%
Qwen-3 14B	96%	96%	97%
Llama-3-70B	94%	95%	96%
Claude-3.7-Sonnet	95%	96%	95%

Qualitative Analysis on Decision Pivots. The core of our method lies in the concept of decision pivots. More specifically, our method assumes that a decision pivot must be visited in order to reach a correct decision. In order to validate this claim, we perform a qualitative analysis using a human-labeled set of *decision pivots*, where human

annotators highlight the keywords/key logic that justify the label (i.e., decision pivots). The dataset consists of 500 Q&A samples with questions, answers, and human-annotated decision pivots. Then, for each sample, we run multiple (Q) zero-shot inferences and see how often the model can retrieve the ground truth decision pivots. In Table 4, we report the *pivot-retrieval rate* as the fraction of the zero-shot reasoning among the Q candidates that explicitly visits the human-labeled pivot (Table 4). Across all backbones, retrieval increases monotonically with K, with the largest gains observed when moving from Q=10 to Q=100 and markedly smaller gains beyond Q=100 (diminishing returns). Larger models begin closer to the ceiling even with small q (e.g., 8–14B already exceed ∼94–96% at Q=10), while smaller models benefit more from additional resampling. In other words, even the smallest 0.6B model was able to visit the decision pivot in its reasoning path, suggesting that correct solutions repeatedly traverse a sparse, shared set of decision pivots, and that modest resampling suffices to surface those pivots with high probability.

Effect of reasoning data on the reasoning output length of the trained model. We also study the effect of the generated reasoning data on the output reasoning length in Table 5. Specifically, we compare the length of the model output trained with various types of reasoning. We find that training on shorter, curated reasoning path data yields markedly more compact generations. This aligns with recent observations in Munkhbat et al. [26], that self-training elicits shorter reasoning outputs. Relative to zero-shot reasoning (1184.46 tokens on average), Max-ROSCOE reduces length to $1,038.82$ (-145.64 tokens), while our Short-Path Reasoning (SPR) compresses further to 750.40 (-434.06 tokens). Model trained with SPR is also roughly a quarter (27.8%) shorter than Max-ROSCOE (-288.42 tokens), without any explicit filtering over the reasoning length.

Table 5: Effect of generated reasoning on the trained model's average output reasoning length across 1.5k samples.

Trained with.	Average #. of Tokens
Zero-Shot Reasoning	1184.46
Metric-filtered Reasoning	1038.82
STAR41	1060.95
PATHFINDER10	761.53
Short-Path Reasoning (Ours)	**750.40**

Decision Pivots as Metrics for Reasoning. We introduce a pivot-based verifiability score for reasoning assessment. Given a pool of K candidates $\{r_k\}$ and the ROMA short-path $\hat{r}$ (our best available reasoning), let $P(r)$ denote pivots extracted and validated by the verifier v_ψ from path r, and let $P^\star = P(\hat{r})$. We score each candidate by the F_1 overlap with the short-path pivots, $\frac{2\,|P(r)\cap P^\star|}{|P(r)|+|P^\star|+\varepsilon}$ (with small $\varepsilon > 0$ for stability). The intuition is that $\hat{r}$ consolidates the necessary decision pivots, so a verifiable candidate should cover most of $P^\star$ while avoiding spurious pivots. In practice, we use v_ψ (LLM-as-judge [3]) for the scoring. For preference-data selection (Table 6), the pivot score alone matches ROSCOE on LogiQA (0.466 vs. 0.469) and yields modest gains on MedQA (0.236 vs. 0.233) and MATH500 (0.680 vs. 0.677). Crucially, combining the pivot score with ROSCOE delivers the best results across all three benchmarks (0.470 / 0.238 / 0.682), indicating that verifiability via decision pivots is complementary to general CoT quality metrics and leads to more effective data curation. While our experiments employ DPO, the signal is algorithm-agnostic and can extend to other RLHF variants (e.g., GRPO, PPO) for both offline selection and on-policy filtering.

Table 6: Comparison of different preference data selection methods on LOGIQA, MEDQA, and MATH500. We report the downstream task accuracy after RL (DPO) training.

Preference Data	LogiQA	MedQA	MATH500
N/A (Zero-Shot)	$0.420_{\pm 0.006}$	$0.159_{\pm 0.009}$	$0.636_{\pm 0.006}$
ROSCOE [9]	$0.469_{\pm 0.004}$	$0.233_{\pm 0.005}$	$0.677_{\pm 0.005}$
PAS (Ours)	$0.466_{\pm 0.003}$	$0.236_{\pm 0.004}$	$0.680_{\pm 0.003}$
PAS + ROSCOE	$\mathbf{0.470}_{\pm 0.004}$	$\mathbf{0.238}_{\pm 0.003}$	$\mathbf{0.682}_{\pm 0.003}$

6 Concluding Remarks

Decision pivots provide a principled lens for improving and refining reasoning models at scale. By mining these pivots and compressing explanations into short-path reasoning, our self-training framework delivers concurrent gains in task accuracy and reasoning quality, all without human-annotated reasoning. Crucially, we decouple *reasoning* and *rewriting*: the reasoner supplies diverse reasoning paths while a lightweight, domain-tuned verifier rewrites them. Although such fine-tuned verifiers tend to sacrifice broad, general-purpose reasoning ability, they rewrite more precisely within their domain; using the reasoner for coverage and the verifier for consolidation makes the resulting traces more verifiable, pivot-focused, and ultimately more effective supervision. The improvements

on multiple benchmarks (e.g., LogiQA, MedQA, MATH500) suggest that the concept decision pivots are broadly applicable and open a path toward human-level reasoning.

References

[1] Mohammad Hossein Amani, Aryo Lotfi, Nicolas Mario Baldwin, Samy Bengio, Mehrdad Farajtabar, Emmanuel Abbe, and Robert West. Rl for reasoning by adaptively revealing rationales. *arXiv preprint arXiv:2506.18110*, 2025.

[2] Anthropic. The claude 3 model family: Opus, sonnet, haiku — model card. `https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf`, 2024. Model card.

[3] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022. doi: 10.48550/arXiv.2212.08073. URL `https://arxiv.org/abs/2212.08073`.

[4] Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations, 2024. URL `https://arxiv.org/abs/2412.13171`.

[5] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training, 2025. URL `https://arxiv.org/abs/2501.17161`.

[6] DeepSeek-AI et al. R1-zero: Scaling verifiable rewards for reasoning. *arXiv preprint arXiv:2501.12948*, 2024.

[7] Yao Fu, Hao Zhang, Rui Zhang, Hong Liu, Adam Trischler, and Bingsheng Yao. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2212.04437*, 2022.

[8] Max Glockner, Johannes Betz, and Hinrich Schütze. Thoughtsource: A central hub for cross-task chain-of-thought datasets. *arXiv preprint arXiv:2308.14767*, 2023.

[9] Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Roscoe: A suite of metrics for scoring step-by-step reasoning, 2023. URL `https://arxiv.org/abs/2212.07919`.

[10] Olga Golovneva, Sean O'Brien, Ramakanth Pasunuru, Tianlu Wang, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. Pathfinder: Guided search over multi-step reasoning paths. *arXiv preprint arXiv:2312.05180*, 2023.

[11] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian et al. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

[12] Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint arXiv:2505.17813*, 2025.

[13] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[14] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL `https://arxiv.org/abs/2106.09685`.

[15] Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning, 2025. URL `https://arxiv.org/abs/2507.00432`.

[16] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[17] Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. *arXiv preprint arXiv:2310.08491*, 2023. doi: 10.48550/arXiv.2310.08491. URL https://arxiv.org/abs/2310.08491. ICLR 2024.

[18] Takeshi Kojima, Shixiang Shawn Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022.

[19] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024. URL https://arxiv.org/abs/2309.00267. PMLR 235:26874–26901.

[20] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-demo.21.

[21] Bingjie Li, Zhanghao Liu, Ming Wang, and Minlie Huang. Active chain-of-thought prompting for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[22] Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020.

[23] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.153. URL https://aclanthology.org/2023.emnlp-main.153/.

[24] Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-reasoner: Advancing llm reasoning across all domains, 2025. URL https://arxiv.org/abs/2505.14652.

[25] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023. URL https://arxiv.org/abs/2303.17651.

[26] Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. Self-training elicits concise reasoning in large language models, 2025. URL https://arxiv.org/abs/2502.20122.

[27] Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Micah Goldblum, et al. Show your work: Scratchpads for intermediate computation with language models. In *International Conference on Learning Representations (ICLR)*, 2022.

[28] 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. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

[30] Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. Receval: Evaluating reasoning chains via correctness and informativeness, 2023. URL https://arxiv.org/abs/2304.10703.

[31] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2023. URL https://arxiv.org/abs/2305.18290.

[32] Sheikh Shafayat, Fahim Tajwar, Ruslan Salakhutdinov, Jeff Schneider, and Andrea Zanette. Can large reasoning models self-train?, 2025. URL https://arxiv.org/abs/2505.21444.

[33] Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. Mind the gap: Examining the self-improvement capabilities of large language models. *arXiv preprint arXiv:2412.02674*, 2024.

[34] Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. Mind the gap: Examining the self-improvement capabilities of large language models, 2025. URL https://arxiv.org/abs/2412.02674.

[35] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

[36] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2023.

[37] Yihan Wang, Si Si, Daliang Li, Michal Lukasik, Felix Yu, Cho-Jui Hsieh, Inderjit S Dhillon, and Sanjiv Kumar. Two-stage llm fine-tuning with less specialization and more generalization, 2024. URL https://arxiv.org/abs/2211.00635.

[38] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022.

[39] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, et al. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2023.

[40] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[41] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

[42] Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin godel machine: Open-ended evolution of self-improving agents, 2025. URL https://arxiv.org/abs/2505.22954.

[43] Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data, 2025. URL https://arxiv.org/abs/2505.03335.

[44] Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing general reasoning without verifiers, 2025. URL https://arxiv.org/abs/2505.21493.

A Appendix

Algorithm 1: Reasoning Optimization via Multi-path Aggregation

Inputs: p_θ (reasoner), v_ψ (verifier; LoRA on p_θ), dataset $\mathcal{D}$, samples K, rounds L, DPO temp. β, batch size B
Output: Updated parameters θ
$p_{\text{ref}} \leftarrow$ `StopGradient`(p_θ) ; `// Frozen DPO reference`
for $\ell \leftarrow 1$ **to** L **do**
  $\mathcal{P} \leftarrow \varnothing$;
  foreach $(x_i, y_i) \in \mathcal{D}$ **do**
    `// A: sample and filter candidates using eq. (2); stop when` K `paths collected`
    $\mathcal{S}_i \leftarrow \varnothing,\ R_i^+ \leftarrow \varnothing$;
    while $|\mathcal{S}_i| < K$ **do**
      draw $r \sim p_\theta(r \mid x_i)$; `// per eq. (2)`
      $\mathcal{S}_i \leftarrow \mathcal{S}_i \cup \{r\}$;
      if $\arg\max_y p_\theta(y \mid x_i, r) = y_i$ **then**
        $R_i^+ \leftarrow R_i^+ \cup \{r\}$
    if $|R_i^+| = 0$ **then**
      continue
    `// B: synthesize short path (eq. (3))`
    $\hat{r}_i \leftarrow v_\psi(x_i, R_i^+)$;;
    if $\arg\max_y p_\theta(y \mid x_i, \hat{r}_i) \neq y_i$ **then**
      $\hat{r}_i \leftarrow \arg\max_{r \in R_i^+} p_\theta(y_i \mid x_i, r)$
    `// C: build preferences (eq. (4))`
    $\mathcal{P} \leftarrow \mathcal{P} \cup \{(\hat{r}_i, r) : r \in R_i^+ \setminus \{\hat{r}_i\}\}$;
  foreach minibatch $\mathcal{B} \subset \mathcal{P}$ of size B **do**
    Update θ with $\mathcal{L}_{\text{DPO}}(\theta; \mathcal{B}, p_{\text{ref}}, \beta)$ using eq. (5);
return θ

A.1 Discussion on Method

Listing 1: Prompt for consolidating K reasoning paths into a single refined Short-Path Reasoning (SPR).

```
You are an expert analyst. You will be given several correct reasoning paths.
↪  Rewrite a refined reasoning that focuses on shared key
↪  information/keywords.

I have the following reasoning paths:
{chr(10).join([f"Reasoning {i+1}: {r}" for i, r in enumerate(k_reasonings)])}

Begin by providing a list of shared decision pivots. Only include the decision
↪  pivots when they are visited by the majority of the provided reasoning
↪  paths in the candidate pool.

Then, aggregate the multiple reasoning paths and provide a single refined
↪  reasoning that:

1) focuses on the decision pivots (keywords/key information) that are shared
↪  across the candidate paths
2) Avoids repetition
```

Intuition. Reasoning traces that reach the same correct decision tend to converge on a small set of decision pivots—minimal, verifiable facts or logical checkpoints that are causally responsible for the answer—while differing in style and superfluous steps. Our method exploits this structure: (i)

surface diverse but successful paths, (ii) intersect them to recover the shared pivots, and (iii) rewrite a compact Short-Path Reasoning (SPR) that keeps only pivot-bearing content. Preference learning then aligns the model toward these concise, faithful explanations.

Backbone and thinking mode. We use the Qwen-3 family, which supports two output channels: thinking content (intermediate chain-of-thought) and output content (the final, externally visible answer or explanation). Unless stated otherwise, we enable thinking mode by default during data generation so that the model emits both channels; we selectively log these channels, as detailed below in Stage A.

Stage A: Multi-sample bootstrapping (diversity with guardrails). For each (x_i, y_i), we obtain a candidate pool of size K by interleaving zero-shot sampling and a guided fallback:

1. Zero-shot pass. Prompt the model to predict the label. It returns thinking (reasoning for its prediction) plus result (predicted label). If the predicted label matches y_i, we save the thinking trace into the pool R_i^+; these paths reflect the model's canonical high-probability pivots.
2. Guided fallback on errors. If the prediction is incorrect, we provide the ground-truth label y_i and ask the model to justify it step-by-step (without revealing that y_i was given as a hint). The model again emits thinking plus result (an external, stepwise explanation for y_i). To avoid hint-contamination of internals, we discard the thinking and save only the external explanation (result) into R_i^+. These guided traces widen pivot coverage by routing around the model's default heuristics.

We repeat until $|R_i^+| = K$ (resampling as needed). This produces a pool of successful paths that agree on the label while differing in which pivots they articulate and how they organize them.

Stage B: Short-path synthesis with a lightweight verifier (pivot recovery). Given R_i^+, a small verifier v_ψ (a LoRA-tuned adapter on the same backbone) aggregates overlapping content and rewrites a single concise reasoning $\hat{r}_i$ that (a) enumerates the shared decision pivots visited by the majority of candidates and (b) produces a compact chain that foregrounds only those pivots. We use the prompt in Listing 1, summarized as:

- Step 1 (pivot mining). List the shared decision pivots and include only those visited by the majority of paths. This majority rule suppresses idiosyncratic steps and raises the signal-to-noise ratio of the pivot set.
- Step 2 (pivot-centric rewrite). Aggregate the paths and provide a single refined reasoning that focuses on shared pivots and avoids repetition. This enforces minimality and fluency while preserving causal sufficiency.

Finally, we verify that $\arg\max_y p_\theta(y \mid x_i, \hat{r}_i) = y_i$ before accepting $\hat{r}_i$ (a simple correctness check). The verifier is used only at training time for synthesis; it is discarded at test time to keep inference cost unchanged. See also Listing 1 for the exact template.

Stage C: Preference optimization (DPO over pairs). We treat the synthesized $\hat{r}_i$ as chosen and each remaining $r \in R_i^+ \setminus \{\hat{r}_i\}$ as rejected, forming pairs $(\hat{r}_i, r)$. Direct Preference Optimization (DPO) then increases the relative likelihood of $\hat{r}_i$ under the policy while regularizing toward a frozen reference. Intuitively, DPO turns the pivot-dense rewrite into a stronger learning signal than any single raw path.

How each component contributes. (i) Bootstrapping supplies diverse, label-consistent evidence, mixing the model's canonical routes (zero-shot) with counter-heuristic routes (guided) to stabilize pivot extraction. (ii) Verifier-guided synthesis converts many paths into one faithful, minimal SPR by explicitly mining majority pivots and pruning spurious steps; this raises pivot density and tightens decision margins. (iii) Preference learning aligns the generator toward SPR—rewarding causal sufficiency and compactness rather than length or stylistic goodness—and does so without hand-crafted external metrics or human rationales.

Prompt for Step B (exact). See Listing 1 for the full template used to (1) list shared pivots and (2) produce the refined SPR. This two-stage instruction was crucial in practice: eliciting pivots before rewriting makes the subsequent consolidation measurably more stable than asking for a single rewrite in one shot.

Remark on modes. Thinking mode is leveraged only to collect richer raw material in Stage A (and we carefully choose which channel to retain); the final trained model can be deployed with or without thinking mode at inference, since the verifier is not required at test time.

A.2 Experimental Results and Analysis

Table 7: Zero-Shot Classification accuracy on LogiQA Classification with various types of generated reasoning.

Additional Reasoning	Acc.
None (Zero-Shot)	0.420
Random Reasoning	0.405
Max ROSCOE reasoning	0.438
Short-path reasoning (Ours)	**0.459**

Does reasoning affect downstream task performance? Alternatively, we can assess the quality of generated reasoning by adding the generated reasoning during zero-shot prediction. Specifically, we compare three different types of generated reasoning (1) Zero-Shot Reasoning, (2) Max ROSCOE Reasoning, and (3) Short-Path Reasoning (Ours) by appending the chain-of-thought reasoning in-context. We report the results in Table 7. Here, we observe three takeaways. First, merely appending random zero-shot reasoning provides a small benefit over using no reasoning at all ($\uparrow$ 0.012), suggesting a generic "reasoning-priming" effect. Second, higher-quality reasoning paths help more: Max-ROSCOE improves to 0.571 accuracy ($\uparrow$ 0.025). Third, our pivot-focused Short-Path Reasoning yields the largest gains ($\uparrow$ 0.053), outperforming Max-ROSCOE by $\uparrow$ 0.028 accuracy. Notably, the F1 improvements consistently exceed accuracy gains, indicating that concise, pivot-dense paths improve positive/negative discrimination rather than only shifting the decision threshold. These results support our hypothesis that the transfer signal comes from shared decision pivots rather than from longer or stylistically rich explanations.

Table 8: Zero-Shot accuracy on LogiQA with different backbones.

Method	**Overall Acc.**
Zero-Shot (Qwen-3 0.6B)	0.420
Zero-Shot (Qwen-3 1.7B)	0.643
Zero-Shot (Qwen-3 4B)	0.726
Zero-Shot (Qwen-3 8B)	0.747

Effect of backbone on downstream task. In Table 8, we analyze the effect of the model on the zero-shot downstream task accuracy. In zero-shot classification, accuracy improves monotonically with model scale from 0.420 (Qwen-3 0.6B) to 0.643 (1.7B), 0.726 (4B), and 0.747 (8B), a net gain of +0.327. The largest jump occurs between 0.6B$\rightarrow$1.7B (+0.223), with diminishing returns thereafter (+0.083 from 1.7B$\rightarrow$4B; +0.021 from 4B$\rightarrow$8B). Consistent with our experimental setup, we use the 0.6B backbone as the default for controlled ablations and self-training loops, and treat larger models as scale references to contextualize improvements.

Comparison of Inference Latency As shown in Table 9, end-to-end latency with chain-of-thought increases with model size—8–12 s (Qwen-3 0.6B), 14–20 s (1.7B), 25–27 s (4B), 42–48 s (8B), and 55–60 s (14B) per sample. Because our data-generation pipeline emits K reasoning traces per example (with possible re-sampling), the wall-clock cost grows approximately linearly with K. For instance, at $K{=}10$ this corresponds to $\sim$1.3–2.0 min (0.6B), $\sim$4.2–4.5 min (4B), $\sim$7–8 min (8B), and $\sim$9–10 min (14B) per example. Thus, both model size and K are key efficiency knobs; our short-path synthesis (which yields shorter traces) helps amortize corpus-generation costs at scale.

Table 9: Average Inference Time of varying models.

Method (with Thinking)	Latency per Sample
Qwen-3-0.6B	8–12 s
Qwen-3-1.7B	14–20 s
Qwen-3-4B	25–27 s
Qwen-3-8B	42–48 s
Qwen-3-14B	55–60 s

Zero-shot vs. Guided Reasoning for Pivot Diversity At a high level, we exploit two sources of generated reasoning to widen coverage of the task's decision-pivot space. *Zero-shot correct* reasoning paths reveal the model's canonical, high-probability pivot sets (i.e., what the model already trusts), while *guided* traces (generated after revealing y on initial mispredictions) allow the model to route around its default heuristics and surface counter-heuristic or novel pivots that still justify y. Aggregating both sources before short-path synthesis yields a pool of successful paths that agree on the label yet differ in which pivots they invoke and how they organize them. This, in turn, improves the stability of pivot extraction and increases the chance that the final short-path reasoning captures minimal but diverse justifications that generalize.

A.3 Experimental Details

In this section, we report the experimental details of our paper. In all of our experiments, we have used the HuggingFace's TRL (Transformer Reinforcement Learning) library [35]. Regarding the datasets mentioned throughout our work, we use three public benchmarks, LogiQA [22], MedQA [16], and MATH500 [13], all available through Huggingface's DATASETS library [20]. In all datasets, we used the default train-test split provided in the original papers. Regarding the hyperparameters (e.g., SFT learning rate and RL learning rate, number of training epochs), we used the default setting provided in the TRL library. For instance, the DPO learning rate was set as $1e-06$, the epochs set as 3, using LoRA attention dimension of 16 with alpha of 32. When performing SFT, we used 2 epochs with a learning rate of $4e-05$. Regarding unique hyperparameters, we set K (the number of reasoning candidate pool) as 5 and evaluated on a single self-training loop. Please see Table 3 in Section 5 for a detailed explanation of the hyperparameter-tuning results. For all experiments, we used the Qwen-3 0.6B model [40]. Lastly, regarding the computing resources, we used five NVIDIA A100 GPUs for all experiments. For our experiments, we used the 2.2.1 version of Pytorch [29].

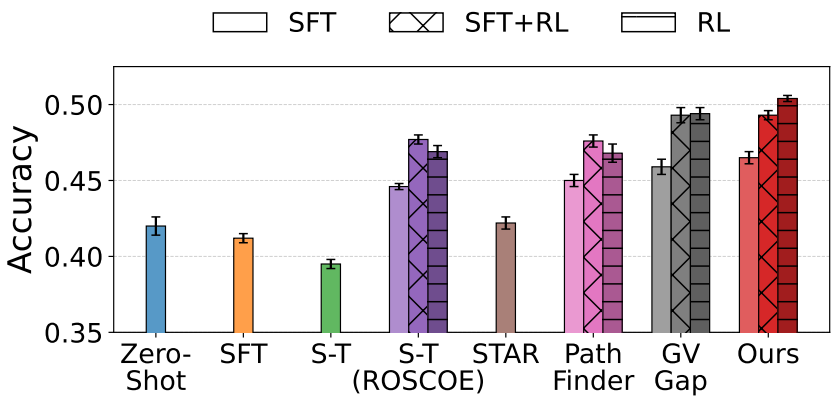


Figure 6: Comparison of self-training results on LogiQA.

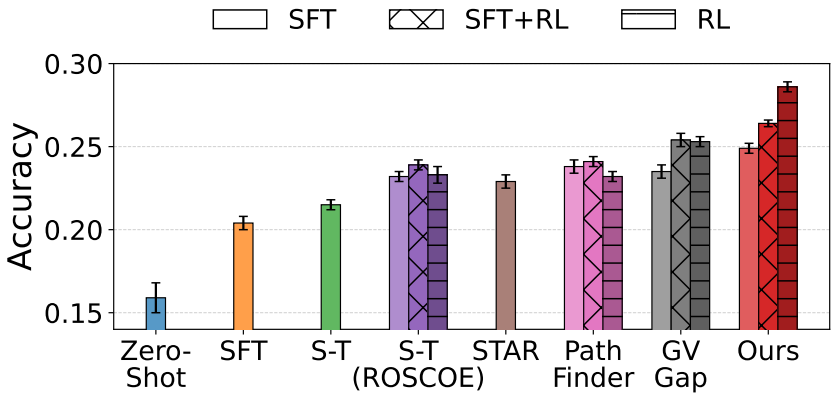


Figure 7: Comparison of self-training results on MedQA.

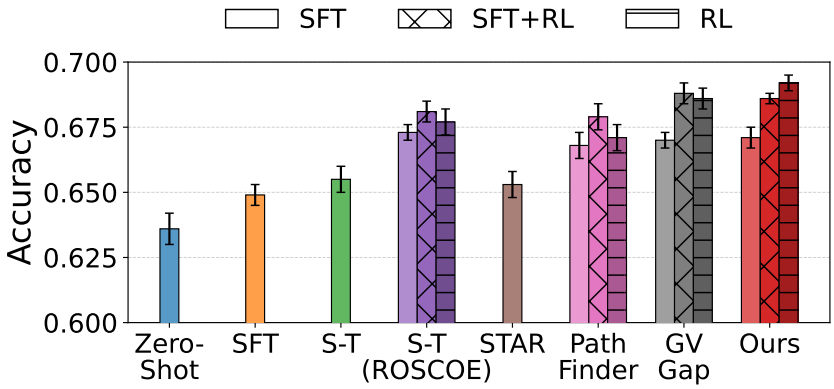


Figure 8: Comparison of self-training results on MATH500.

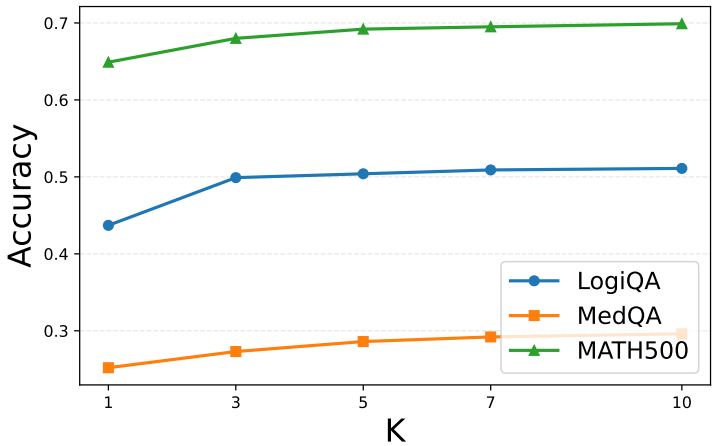


Figure 9: Effect of the candidate pool size K on downstream accuracy for LogiQA, MedQA, and MATH500. Larger K steadily improves performance, with diminishing returns near $K{=}10$.

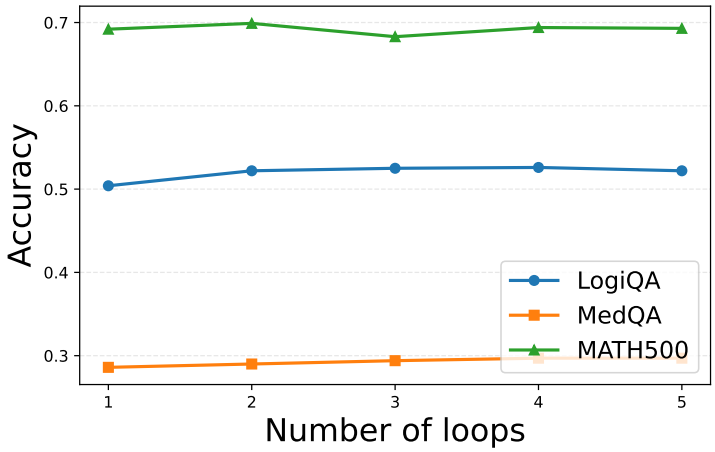


Figure 10: Effect of the number of self-training loops on downstream accuracy. Moderate looping (e.g., 2–4) helps, while excessive looping can overfit or oscillate.