CORRECT REASONING PATHS VISIT SHARED DECISION PIVOTS

Dongkyu Cho[1,*], **Amy B. Z. Zhang**[2], **Bilel Fehri**[2], **Sheng Wang**[2,3]
Rumi Chunara[1], **Hengrui Cai**[2,4], **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 paths 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, converges 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 path 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 human-annotated 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 (Wei et al., 2022). 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) (Ouyang et al., 2022) 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, semi-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 (Zelikman et al., 2022). 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 (Rafailov et al., 2023), treating SPRs as preferred positives and individual reasoning paths as negatives. The entire loop is model-driven and repeatable at scale (Shafayat et al., 2025). Experiments on LOGIQA(Liu et al., 2020), MEDQA(Jin et al., 2021), and MATH500(Hendrycks et al., 2021) 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.

Contributions.

- We formalize *decision pivots* and show that correct reasoning paths intersect on a set of *pivots*.

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

- We propose ROMA, a bootstrap→rewrite framework that aggregates K paths and, via a fine-tuned verifier, distills a compact, pivot-focused short-path reasoning (SPR).
- We build a self-training pipeline that leverages SPR to improve both task accuracy and reasoning 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.

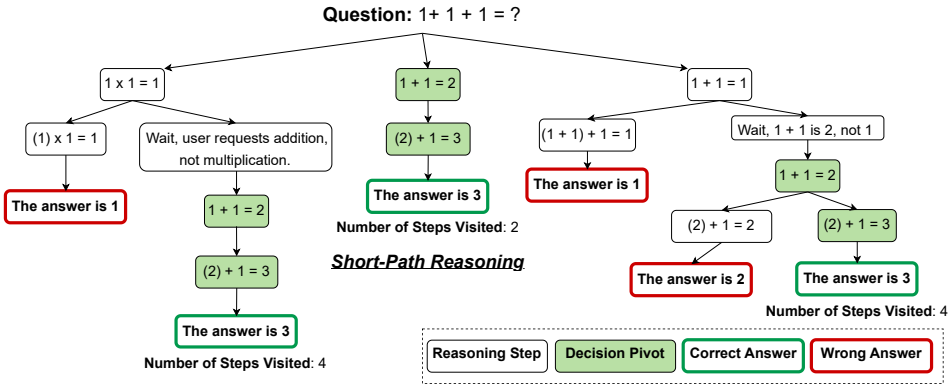


Figure 1: A CoT reasoning may take various paths to reach a final decision, naturally visiting redundant or incorrect thought steps. However, such errors are not easily captured by prior methods. Instead, we introduce the concept of decision pivots, which are a set of key information that a reasoning 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.

2 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 $p_\theta(r, \hat{y} \mid x)$ (Anthropic, 2024; Yang et al., 2025) 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 path) 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 ideal but unobserved 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, 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, while human verification is costly and lacks scalability. Formally, the objective can be stated as finding a reasoning path r that suffices $\max_r \; p_\theta\big(y \mid x, r\big)$, subject to r being as concise as possible while preserving correctness. This requires identifying only the key information necessary to support the answer, despite the lack of supervision and step-level verifiability (Zhou et al., 2025).

Positioning: 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 (Zelikman et al., 2022) 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 (Wang et al., 2023) aggregates predictions from multiple sampled reasoning paths, improving answer accuracy but producing no refined intermediate reasoning for training. RLVR (DeepSeek-AI et al., 2026; Zhang et al., 2025; Zhao et al., 2025) 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 (Golovneva et al., 2023a; Liu et al., 2023; Bai et al., 2022)

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 4. A detailed introduction of preliminary works is provided in Section A.4. In the following section, we seek to address the general problem of producing reasoning in the absence of hypothetical ground-truth reasoning and automated verifiers.

3 Reasoning Optimization via Multi-path Aggregation

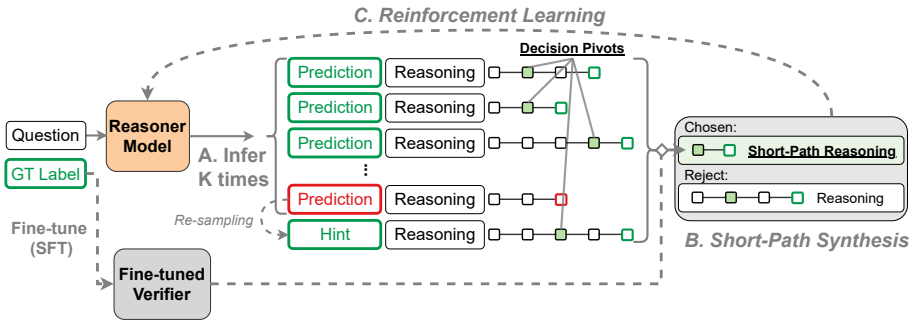


Figure 2: We present a novel self-training framework ROMA that leverages the concept of decision-pivots. Our self-training framework works in 3 stages. Given a question: (A) model produces multiple PREDICTION + REASONING pairs, ensuring we collect K reasoning paths using re-sampling. (B) Then, a fine-tuned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

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 – SPR), then feeds the SPR back into the model to self-improve its reasoning and generalization.

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 $\hat{y}$ (Yang et al., 2025). 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 (Hu et al., 2021) on p_θ) synthesizes a SPR $\hat{r}_i = v_\psi(x_i, \mathcal{R}_i^+)$.

3.1 Stage A: Multi-sample bootstrapping

For each (x_i, y_i), sample K pairs:

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

If $\mathcal{R}_i^+$ is empty, resample paths from $p_\theta(r \mid x_i, y_i)$ to ensure coverage (See Step A in Figure 2). This step explores diverse 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* paths (re-sampled after revealing the hint y_i on the initial incorrect prediction (Zelikman et al., 2022; Amani et al., 2025)) allow the model to route around its default heuristics and adopt novel paths that rationalize y_i.

3.2 STAGE B: VERIFIER-GUIDED SHORT-PATH SYNTHESIS

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 SPR that focuses on the decision pivots, and preserves the correct decision (See Step B in Figure 2):

$$\hat{r}_i \;=\; v_\psi(x_i, \mathcal{R}_i^+) \quad \text{subject to} \quad \arg\max_y p_\theta(y \mid x_i, \hat{r}_i) = y_i. \tag{2}$$

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 SPR, separate from the reasoner p_θ that produces a pool of K candidate paths. 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 SPR during training and is discarded at test time. For clarification, the verifier is trained only on the training split and never applied to test examples. The motivation is the specialization–generalization trade-off: fine-tuning on $\mathcal{D}$ can inject domain knowledge but tends to deteriorate general reasoning ability (Chu et al., 2025; Cho et al., 2025b; Wang et al., 2024; Ma et al., 2025). 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 reasoning paths to decision pivots, and consolidate them into faithful short paths. Here, the cost is minimal asv_ψ is obtained via small LoRA adapters on the same backbone. Empirically, verifier-guided rewriting provides stronger learning signals than self-rewriting with p_θ (see Fig. 4 and Tab. 1 in Section 4).

3.3 STAGE C: PAIRWISE PREFERENCE OPTIMIZATION

We align the generator p_θ to prefer the synthesized short path. For each (x_i, y_i), treat $\hat{r}_i$ as chosen and every other successful candidate $r \in \mathcal{R}_i^+ \setminus \{\hat{r}_i\}$ as rejected, forming

$$\mathcal{P}_i = \{(\hat{r}_i, r) : r \in \mathcal{R}_i^+ \setminus \{\hat{r}_i\}\}. \tag{3}$$

We then optimize a standard DPO objective (Rafailov et al., 2023) for $p_\theta(r \mid x, y)$ against a frozen reference p_{ref} (the exact loss is given in Appendix A.1). Intuitively, because $p_\theta(r \mid x, y) = \prod_t p_\theta(t_t \mid x, y, t_{<t})$, the preference update increases token probabilities precisely at the positions where $\hat{r}_i$ and alternatives differ—i.e., the pivot steps—while the implicit KL term (via p_{ref}) regularizes drift.

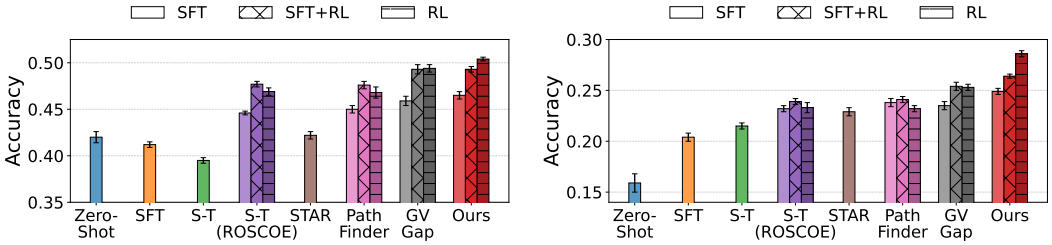
Discussion: Pivot-focused supervision. Assume each instance (x, y) admits a small pivot set $P(x)$ such that correctness is determined by visiting all pivots. Let $r \sim p_\theta(\cdot \mid x)$ and define $q_t := \Pr_\theta[\text{correct token at pivot step } t \in P(x) \mid x]$. Then the final-error probability satisfies $\Pr[\hat{y} \neq y] \;\leq\; \mathbb{E}_x\big[\sum_{t \in P(x)}(1-q_t)\big]$, so reducing the *pivot-miss risk* $\varepsilon_{\text{piv}} := \mathbb{E}_x \sum_{t \in P(x)}(1-q_t)$ reduces task error. Our synthesized SPR $\hat{r}_i$ differs from other successful paths primarily at pivot-bearing tokens. That is, by the token factorization of $\log p_\theta(r \mid x, y)$, the preference update increases q_t exactly at those disagreement positions, concentrating probability mass on pivot-preserving continuations. In addition, replacing long traces by $\hat{r}_i$ removes non-pivot tokens and reduces gradient noise away from pivots, further sharpening updates on the few steps that control correctness. Finally, because preference pairs are built from p_θ's own rollouts, updates act on the prefixes the model actually visits, mitigating distributional shift between training and generation. For a formal sketch of our analysis, please refer to Appendix A.2.

4 EXPERIMENTS

Benchmarks. We validate our approach across three public benchmarks to demonstrate cross-domain effectiveness. LOGIQA (Liu et al., 2020) is a general question-answer dataset that assesses logical reasoning with multiple choice questions (MCQs) where the models are evaluated mainly by accuracy. MEDQA (Jin et al., 2021) 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 (Hendrycks et al., 2021) is a mathematical benchmark commonly used to assess a model's reasoning capabilities. On these benchmarks, we compare different types of self-training baselines that utilize their own generated outputs to improve their reasoning capability, which in turn improves a model's downstream task performance (Zelikman et al., 2022; Golovneva et al., 2023b; Song et al., 2025). In all experiments, we used a held-out set of test data for evaluation.

Architectures and protocol. We evaluate our method on variants of Qwen-3 (Yang et al., 2025), 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 (Anthropic, 2024), LLama-3 (Grattafiori et al., 2024). 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 (Table 2). Further detail regarding the experimental setting is included in Section A.5

4.1 Experimental Results



(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.

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 LogiQA, MedQA, and MATH500, a consistent pattern emerges (Table 1): performance improves as supervision becomes more focused on decision pivots rather than full reasoning paths. Training without reasoning supervision or reusing zero-shot reasoning, fails to yield reliable gains and can even degrade performance, indicating that unguided reasoning does not provide a stable learning signal. Simply exposing the model to more reasoning text is therefore insufficient; without guidance, the model cannot distinguish decision-critical steps from redundant or spurious ones. Filtering self-generated reasoning using general-purpose quality metrics (e.g., ROSCOE) partially alleviates this issue by removing low-quality paths, but its effect plateaus. Such metrics capture surface-level coherence or fluency, yet remain agnostic to task-specific decision pivots, limiting their ability to drive further improvements.

In contrast, compressing multiple successful reasoning paths into a pivot-focused short-path reasoning (SPR) concentrates supervision on the minimal information shared across correct solutions. When paired with preference learning, this pivot-centric supervision consistently outperforms metric-filtered and self-training baselines across all benchmarks. The strongest gains arise when SPR synthesis is guided by a domain-tuned verifier (Section 3.2) , rather than the reasoner itself. The verifier injects task-specific priors, prunes spurious steps, and sharpens the preference signal during consolidation. This effect scales with domain knowledge requirements: it is most pronounced on MedQA, clear on LogiQA, and smaller yet consistent on MATH500, where intermediate correctness is already externally verifiable. These empirical trends directly support our theoretical analysis in Appendix A.2, which formalizes decision pivots as sparse, verifiable determinants of correctness. By concentrating supervision on pivot-satisfying paths and amplifying their influence

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

Method	Techniques	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 [11])	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 [46]	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 [12]	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 [39]	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	$\mathbf{0.504}_{\pm 0.002}$	$\mathbf{0.286}_{\pm 0.003}$	$\mathbf{0.692}_{\pm 0.003}$

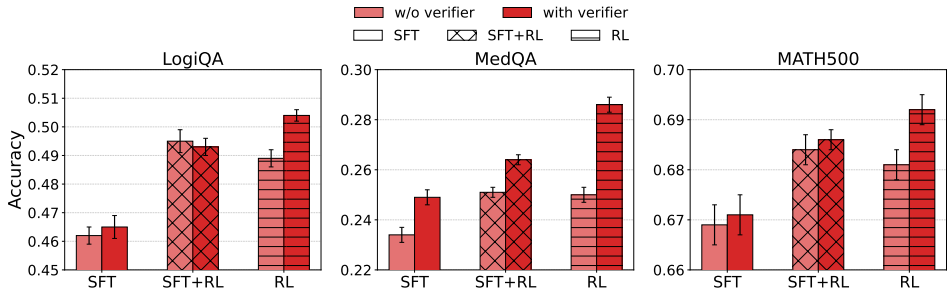


Figure 4: Effect of the fine-tuned verifier on downstream accuracy across post-training regimes. We compare our method with and without a verifier on LogiQA, MedQA, and MATH500. The verifier yields consistent gains and is most impactful in the expert MedQA domain, supporting our claim that a domain-tuned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

through verifier-guided consolidation, our method increases the alignment between training signals and task utility, explaining both the robustness and the scaling of the observed gains.

Effect of Fine-tuned Verifiers. Incorporating a fine-tuned verifier consistently improves downstream accuracy by synthesizing higher-quality, pivot-focused SPR (Table 1; Figure 4). Across SFT, SFT+RL, and RL regimes, the verifier refines supervision by filtering spurious steps, aligning paths to shared decision pivots, and injecting domain-specific priors during consolidation, resulting in a sharper and more informative learning signal. The magnitude of this effect follows task characteristics. Gains are largest on the knowledge-intensive MEDQA benchmark, where domain priors are crucial for identifying valid pivots. Improvements are smaller but consistent on MATH500, where intermediate correctness is already externally verifiable, and more modest on LOGIQA, which involves weaker domain constraints and higher reasoning ambiguity (Wu et al., 2025). In practice, we implement the verifier with LoRA adapters (Hu et al., 2021) for a favorable accuracy–compute trade-off; when domains shift, we retune the verifier to refresh its priors (Table 1). Ultimately, these results align with our analysis in Appendix A.2, which predicts larger gains when correctness depends on a sparse, domain-sensitive set of decision pivots that are easier to verify than to generate.

Are General Reasoning Metrics sufficient for preference data selection? Table 2 and Figure 5 show that general-purpose reasoning metrics such as ROSCOE are poorly aligned with downstream

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

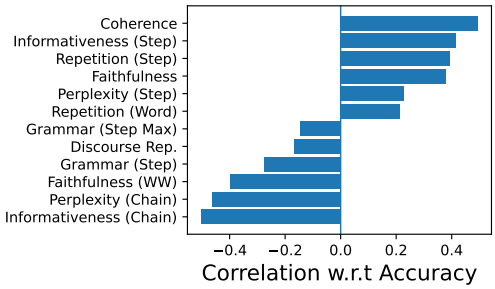


Figure 5: Pearson Correlation between ROSCOE scores and downstream LogiQA accuracy.

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

task accuracy, making them unreliable signals for preference data selection. We observe three consistent patterns. **(1)** As model size increases, most ROSCOE facets remain nearly flat or exhibit idiosyncratic behavior, while downstream accuracy improves substantially (Table 2). **(2)** Cross-scale correlations make this misalignment explicit: several ROSCOE scores are weakly correlated or even negatively correlated with accuracy (e.g., INFORMATIVENESS (CHAIN), FAITHFULNESS (WW)), and the strongest positive correlations are only modest (Figure 5). **(3)** This "flat-high" profile indicates ceiling effects that obscure the variations most predictive of correctness.

This behavior is expected by design. ROSCOE emphasizes fluency and consistency of reasoning chains, not whether they are sufficiently logical or factual to reach the correct answer. As a result, selecting preference data using ROSCOE removes clearly flawed paths at a general level but fails to prioritize the steps that truly rationalize the decision. In contrast, aggregating multiple correct trajectories into SPR explicitly targets shared decision pivots and when combined with a lightweight verifier, it yields larger and more reliable gains than metric-based selection.

Ablation Study. We analyze two key aspects: (1) the number of candidate reasoning paths K used to synthesize SPR, and (2) the number of self-training loops. **Candidate pool size (K).** 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. This reflects improved pivot coverage and reduced sensitivity to peculiar single-paths. Performance continues to improve with larger K but exhibits diminishing returns beyond K=5, indicating that the verifier can effectively consolidate a small, diverse set of paths into an equally informative short path. In practice, K=7 recovers approximately 95–100% of the performance achieved at K=10 while substantially reducing sampling and verification cost; we therefore adopt K=7 as the default. **Self-training loops.** Table 3b shows steady gains from one to two self-training loops as the model internalizes the synthesized short-path signals. A third loop yields

Table 4: Pivot retrieval rate (%) as a function of model size and number of zero-shot candidate reasoning samples K per prompt. A retrieval is counted when the generated candidate reasoning visits the human-annotated decision pivot set.

Model	K=10	K=100	K=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%

smaller but generally positive improvements, while additional loops lead to saturation and occasional regression on math-heavy tasks. We attribute this behavior to pivot-set saturation and mild distributional drift toward synthesized paths, suggesting that a small number of loops is sufficient to capture most of the benefit. 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 paths 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.

Qualitative Analysis on Decision Pivots. Our method assumes that correct solutions repeatedly traverse a small set of decision pivots and that a reasoning path must visit these pivots to reach the correct decision. To probe this assumption, we construct a human-labeled set of 500 Q&A items with pivot annotations (keywords or key logical steps justifying the answer). For each item, we draw K zero-shot samples and compute a pivot-retrieval rate: the fraction of items for which at least one of the K sampled paths explicitly *visits* the human-labeled pivot.[1] As shown in Table 4, retrieval generally *increases and then saturates* with K, with the largest gains from $K{=}10$ to $K{=}100$ and smaller increments beyond. Larger models tend to start closer to saturation even at $K{=}10$ (e.g., 8–14B $\sim$94–96%), while smaller models benefit more from additional resampling. Notably, even a 0.6B model recovers human pivots with high probability under modest resampling, supporting the view that correct traces repeatedly visit a sparse, shared set of pivots. Appendix A.2 connects these observations to our pivot-risk analysis and includes qualitative examples.

Decision Pivots for Reasoning Evaluation and Selection. We introduce a pivot-based verifiability score for reasoning assessment. Given a pool of K candidates $\{r_k\}$ and the 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 $P^\star = P(\hat{r})$. Importantly, $\hat{r}$ serves only as a reference consolidation of shared decision pivots, not as a correctness oracle. 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 (Bai et al., 2022)) for the scoring. To avoid circularity and judge bias, we use a frozen, cross-family verifier for pivot extraction and require structured outputs with evidence spans. Hence, extractions with low confidence or missing evidence are discarded. For preference-data selection (Table 5), 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 5: 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 [11]	$0.469_{\pm 0.004}$	$0.233_{\pm 0.005}$	$0.677_{\pm 0.005}$
PAS (Ours)	$0.466_{\pm 0.003}$	$0.236_{\pm 0.004}$	$0.680_{\pm 0.003}$
PAS + ROSCOE	$\mathbf{0.470}_{\pm 0.004}$	$\mathbf{0.238}_{\pm 0.003}$	$\mathbf{0.682}_{\pm 0.003}$

5 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

[1] A reasoning path is said to visit a pivot if it explicitly mentions a lemmatized text span corresponding to that pivot in the annotated lexicon. We apply normalization and disambiguation rules (e.g., synonym merging and context checks) as detailed in Appendix A.2.

their domain; using the reasoner for coverage and the verifier for consolidation makes the resulting paths 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.

ETHICS STATEMENT

The authors acknowledge and concur with the ICLR Code of Ethics, namely in its pursuit of (1) human well-being, (2) high standards of scientific excellence, (3) consideration for the societal impacts (i.e., harms) of AI, (4) honesty & trustworthiness, (5) fairness, (6) mutual respect for other researchers' works, (7) privacy, and (8) confidentiality.

REPRODUCIBILITY STATEMENT

For reproducibility, we provide the source code, experimental guidelines, and the scripts used in our experiments. Please refer to the README.md file in the supplementary materials on how to reproduce our experiments. We also used a fixed seed setting, which is implemented in the source code. We also include notebook (.ipynb) files to reproduce the figures appearing in our paper.

REFERENCES

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.

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.

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`.

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

Dong Kyu Cho, Inwoo Hwang, and Sanghack Lee. Peer pressure: Model-to-model regularization for single source domain generalization, 2025a. URL `https://arxiv.org/abs/2505.12745`.

Dongkyu Cho, Taesup Moon, Rumi Chunara, Kyunghyun Cho, and Sungmin Cha. Forget forgetting: Continual learning in a world of abundant memory, 2025b. URL `https://arxiv.org/abs/2502.07274`.

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`.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang,

Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2026. URL https://arxiv.org/abs/2501.12948.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. Active prompting with chain-of-thought for large language models, 2024. URL https://arxiv.org/abs/2302.12246.

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.

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, 2023a. URL https://arxiv.org/abs/2212.07919.

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*, 2023b.

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.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. URL https://arxiv.org/abs/2006.07733.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models, 2025. URL https://arxiv.org/abs/2306.08543.

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.

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.

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.

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.

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.

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.

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.

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.

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*, pp. 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-demo.21.

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.

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*, pp. 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/.

Jack Lu, Ryan Teehan, Jinran Jin, and Mengye Ren. When does verification pay off? a closer look at llms as solution verifiers, 2025. URL https://arxiv.org/abs/2512.02304.

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.

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.

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.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2021. URL https://arxiv.org/abs/2112.00114.

Simon Ott, Konstantin Hebenstreit, Valentin Liévin, Christoffer Egeberg Hother, Milad Moradi, Maximilian Mayrhauser, Robert Praas, Ole Winther, and Matthias Samwald. Thoughtsource: A central hub for large language model reasoning data. *Scientific Data*, 10(1), August 2023. ISSN 2052-4463. doi: 10.1038/s41597-023-02433-3. URL http://dx.doi.org/10.1038/s41597-023-02433-3.

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.

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.

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.

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.

Jon Saad-Falcon, E. Kelly Buchanan, Mayee F. Chen, Tzu-Heng Huang, Brendan McLaughlin, Tanvir Bhathal, Shang Zhu, Ben Athiwaratkun, Frederic Sala, Scott Linderman, Azalia Mirhoseini, and Christopher Ré. Shrinking the generation-verification gap with weak verifiers, 2025. URL https://arxiv.org/abs/2506.18203.

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.

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.

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.

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.

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.

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, pp. 24824–24837. Curran Associates, Inc., 2022.

Juncheng Wu, Sheng Liu, Haoqin Tu, Hang Yu, Xiaoke Huang, James Zou, Cihang Xie, and Yuyin Zhou. Knowledge or reasoning? a close look at how llms think across domains, 2025. URL https://arxiv.org/abs/2506.02126.

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.

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

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.

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.

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

A.1 DISCUSSION ON THE PROPOSED 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 paths 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 prompting. Unless noted, the generator p_θ is a [backbone name; e.g., Qwen-3 (Yang et al., 2025)] with a context window of $[X]$ tokens. We use a two-channel prompting scheme (*thinking* for internal draft, *answer* for external output) and strip any internal paths at training time except when explicitly collected 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 path 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 paths 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).

$$\begin{aligned}\mathcal{L}_{\mathrm{DPO}}(\theta) = -\sum_{i=1}^{N} \sum_{(r^+,r^-)\in\mathcal{P}_i} \log\sigma\Big(\beta\Big[& \\ & \underbrace{\log p_\theta(r^+ \mid x_i, y_i) - \log p_\theta(r^- \mid x_i, y_i)}_{\text{policy preference}} \\ & - \underbrace{\log p_{\mathrm{ref}}(r^+ \mid x_i, y_i) - \log p_{\mathrm{ref}}(r^- \mid x_i, y_i)}_{\text{implicit KL correction}}\Big]\Big)\end{aligned} \tag{4}$$

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) (see Equation (4)) 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 (Yang et al., 2025) 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 Theoretical Analysis on Decision Pivots

In this section, we provide a theoretical analysis of why pivot-focused short-path reasoning (SPR) improves self-training. The analysis combines two established perspectives: (i) the generation–verification gap (GV-gap) (Song et al., 2025) framework, which characterizes the effect of reweighting self-generated samples using a verifier, and (ii) a distillation viewpoint, where supervision concentrated on decision-critical tokens yields lower-variance and more sample-efficient updates (Gu et al., 2025).

Setup and notation. For each input x, a reasoner $p_\theta(r, \hat{y} \mid x)$ generates a reasoning path $r = (t_1, \ldots, t_\ell)$ and an answer $\hat{y}$. Let $s(x, r) \in [0, 1]$ denote task success or utility (Song et al., 2025)

(e.g., correctness), and define the expected success

$$J(\theta) := \mathbb{E}_x \, \mathbb{E}_{r \sim p_\theta(\cdot|x)}\big[s(x,r)\big]. \tag{5}$$

A verifier $v_\psi(x,r) \in [0,1]$ induces a weighting function $w(v_\psi(x,r)) \geq 0$ (i.e., nonnegative), which defines the reweighted distribution

$$p_\theta^{(w)}(r \mid x) \;=\; \frac{p_\theta(r \mid x)\, w(v_\psi(x,r))}{\mathbb{E}_{\tilde{r} \sim p_\theta(\cdot|x)}[w(v_\psi(x,\tilde{r}))]}. \tag{6}$$

Here, $v_\psi(x,r)$ denotes a scalar verifier score measuring the quality or correctness of a reasoning path, and $w(\cdot)$ maps this score to a nonnegative training weight. While $p_\theta^{(w)}(\cdot \mid x)$ is introduced for analysis, in our method the verifier uses its scores to select and consolidate multiple candidate paths into a single pivot-preserving short-path reasoning (SPR), which then serves as the distillation target. For clarification, however, while pivots are conceptual decision-critical units, we operationalize their detection via normalized surface forms for evaluation.

Verifier reweighting via the GV-gap. Under this setting, the effect of verifier-based reweighting (i.e., assigning more probability mass to certain paths and less to others; preference weighting) on expected task performance (see Figure 4 and Table 1) can be expressed by the following identity:

$$J^{(w)}(\theta) - J(\theta) = \mathbb{E}_x \left[\frac{\mathrm{Cov}_{r \sim p_\theta(\cdot|x)} \left(s(x,r),\, w(v_\psi(x,r)) \right)}{\mathbb{E}_{r \sim p_\theta(\cdot|x)}[w(v_\psi(x,r))]} \right]. \tag{7}$$

Thus, whenever the verifier score $w \circ v_\psi$ is positively correlated with task success under the current model's generated reasoning outputs, reweighting improves expected performance. This captures the standard GV-gap intuition that verification is beneficial when it correlates with correctness (Song et al., 2025; Saad-Falcon et al., 2025).

Pivot bottleneck assumption. We assume each instance x admits a small set of decision pivots $P(x)$ (not necessarily unique) such that missing one or more pivots constitutes a dominant failure mode. Define the indicator

$$H_x(r) := \mathbf{1}\{r \text{ visits all required pivots in } P(x)\}. \tag{8}$$

We decompose task success as

$$s(x,r) = H_x(r) \cdot \tilde{s}(x,r), \qquad \tilde{s}(x,r) \in [0,1]. \tag{9}$$

This modeling assumption captures tasks where correctness is bottlenecked by a small number of necessary reasoning events. Our empirical pivot-retrieval results (Table 4) support that such pivots are repeatedly expressed across successful reasoning paths.

A sufficient condition for positive GV-gap (sketch). Fix x and let $q(x) := \Pr_{r \sim p_\theta(\cdot|x)}[H_x(r) = 1]$. Suppose that

$$\Delta_s(x) := \mathbb{E}[\tilde{s} \mid H_x{=}1] - \mathbb{E}[\tilde{s} \mid H_x{=}0] \geq \delta > 0, \qquad \Delta_w(x) := \mathbb{E}[w(v_\psi) \mid H_x{=}1] - \mathbb{E}[w(v_\psi) \mid H_x{=}0] \geq \eta > 0. \tag{10}$$

Then, by a two-group covariance decomposition over $\{H_x{=}1, H_x{=}0\}$,

$$\mathrm{Cov}_r \left(s(x,r), w(v_\psi(x,r)) \right) \;\geq\; \delta\, \eta\, q(x)\big(1 - q(x)\big). \tag{11}$$

Combining this bound with equation 7 yields $J^{(w)}(\theta) - J(\theta) > 0$. We emphasize that this provides a sufficient condition explaining when verifier-guided reweighting is expected to help; it is not a universal guarantee.

Why pivot-focused short paths sharpen distillation. Our SPR teacher distribution $p_{\text{SPR}}(r \mid x)$ concentrates probability mass on short, pivot-preserving paths synthesized from correct candidates. Distillation by minimizing

$$\mathbb{E}_x \, \mathrm{KL}(p_{\text{SPR}}(\cdot \mid x) \,\|\, p_\theta(\cdot \mid x)) \tag{12}$$

reduces to cross-entropy under p_{SPR}. Ultimatery, by pruning non-pivot detours, SPR increases the density of pivot-bearing tokens in the supervision signal, concentrating gradient updates on decision-critical positions where candidate paths disagree. In contrast, distilling from longer, candidate

Table 6: 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
STAR [46]	1060.95
PATHFINDER [12]	761.53
Short-Path Reasoning (Ours)	**750.40**

reasoning chains allocates substantial gradient mass to stylistic or redundant tokens that are weakly coupled to correctness. This provides an optimization-level explanation for the empirical stability and efficiency gains of SPR shown in Section 4. Yet, we do not claim all reasoning tasks admit a strict pivot bottleneck. Rather, our above analysis explains why the approach is effective when such structure exists, as empirically shown in Section 4.

A.3 Experimental Results and Analysis

Effect on reasoning length without length incentives. We study how training data type affects the model's reasoning length under identical decoding, with no explicit incentives to be brief. Table 6 shows that pivot-focused supervision yields markedly more compact generations. Relative to zero-shot (1184.46 tokens on average), Max-ROSCOE reduces length to 1038.82 (-145.64, -12.3%), while our short-path reasoning (SPR) compresses further to 750.40 (-434.06, -36.7%). SPR is also -288.42 tokens (-27.8%) shorter than Max-ROSCOE without any length filtering or penalties, supporting the claim that focusing on decision pivots naturally shortens the path. These results align with recent observations that self-training elicits shorter reasoning outputs (Munkhbat et al., 2025). In our case, the pivot-focused SPR teacher offers a simple mechanism for this effect by pruning non-pivot tokens and lowering the teacher's entropy, so the model learns to prefer shorter paths even without any length incentives.

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 SPR 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.

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→1.7B (+0.223), with diminishing returns thereafter (+0.083 from 1.7B→4B; +0.021 from 4B→8B). Consistent with our experimental setup, we use the

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

0.6B backbone as the default for controlled ablations and self-training loops, and treat larger models as scale references to contextualize improvements.

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

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 paths 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 paths) helps amortize corpus-generation costs at scale.

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* paths (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.4 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) (Ouyang et al., 2022) on preference-labeled examples. A key challenge is constructing reliable preference data. One line of work derives labels from explicit metrics (Golovneva et al., 2023a; Prasad et al., 2023) or from model-based feedback (Liu et al., 2023; Kim et al., 2023; Bai et al., 2022; Lee et al., 2024) (LLM-as-judge). RLVR (reinforcement learning with verifiable rewards) removes human preference signals entirely by using programmatic verifiers to assign binary rewards, which are effective in domains like math or coding, where correctness is objectively checkable (DeepSeek-AI et al., 2026). Extensions seek to broaden this to more open ended reasoning (Ma et al., 2025; Huan et al., 2025), but still require task-specific verifiers. **Self-Training.** In parallel, self-training approaches leverage the model's own outputs as training data. The use of self-evolving supervisors was introduced earlier (Cho et al., 2025a; Grill et al., 2020), but the shift of focus onto large reasoning models is more contemporary. STaR (Zelikman et al., 2022) bootstraps from reasoning paths that yield correct answers; Self-Refine (Madaan et al., 2023) uses iterative self-critique; and other works exploit self-consistency as a proxy reward for RL (Shafayat et al., 2025), build self-editing/verifying systems (Zhang et al., 2025), or study the generation–verification gap

theoretically (Song et al., 2025; Lu et al., 2025). 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 (Golovneva et al., 2023a).

Reasoning Models. Chain-of-thought (CoT) prompting (Wei et al., 2022) 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 (Kojima et al., 2022). Research since has explored richer reasoning supervision: scratchpad-style intermediates (Nye et al., 2021), self-consistency via majority voting over multiple reasoning paths (Wang et al., 2023), and targeted data curation by aggregating reasoning corpora (Ott et al., 2023), actively selecting informative queries (Diao et al., 2024), or varying difficulty adaptively (Fu et al., 2022). Recent trends also include compressing CoTs for efficiency or readability, but these typically focus on token length reduction (Munkhbat et al., 2025; Hassid et al., 2025; Cheng & Durme, 2024) 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 supervision signal for self-training.

A.5 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 (von Werra et al., 2020). Regarding the datasets mentioned throughout our work, we use three public benchmarks, LOGIQA (Liu et al., 2020), MEDQA (Jin et al., 2021), and MATH500 (Hendrycks et al., 2021), all available through Huggingface's DATASETS library (Lhoest et al., 2021). 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 7 unless otherwise noted and evaluated on a single self-training loop. Please see Table 3 in Section 4 for a detailed explanation of the hyperparameter-tuning results. For all experiments, we used the Qwen-3 0.6B model (Yang et al., 2025). Regarding the human-annoated set of 500 Q&A samples with pivot annotations in Table 4, we must underline that these annotations are used solely for diagnostic analysis and do not affect training or model selection. 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 (Paszke et al., 2019).

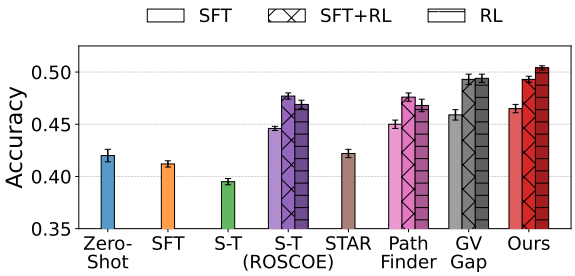


Figure 6: Comparison of self-training results on LogiQA.

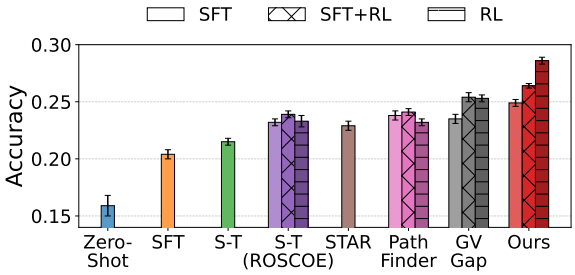


Figure 7: Comparison of self-training results on MedQA.

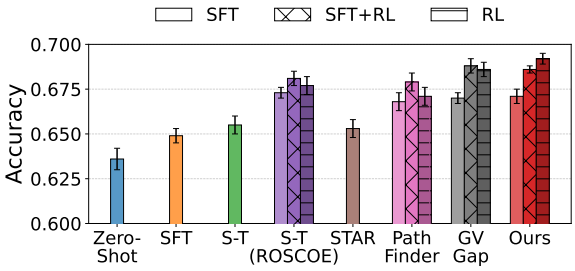


Figure 8: Comparison of self-training results on MATH500.

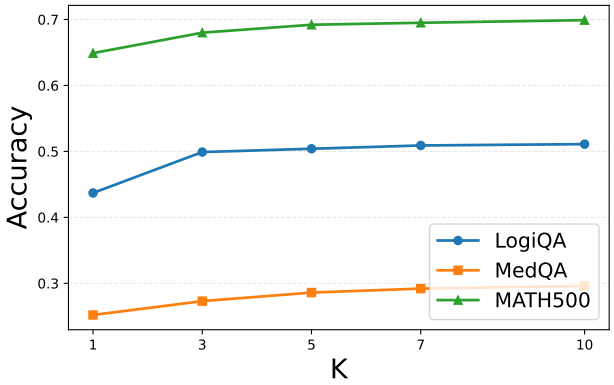


Figure 9: Effect of the candidate pool size K on downstream accuracy for LogiQA, MedQA, and MATH500. Larger K steadily improves performance, with diminishing returns near $K{=}10$.

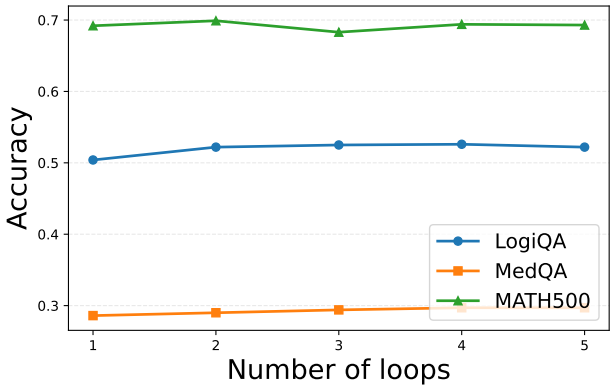


Figure 10: Effect of the number of self-training loops on downstream accuracy. Moderate looping (e.g., 2–4) helps, while excessive looping can overfit or oscillate.