Learning from Contrasts: Synthesizing Reasoning Paths from Diverse Search Trajectories

Peiyang Liu[1,2], **Zhirui Chen**[3], **Xi Wang**[2], **Di Liang**[4], **Youru Li**[5,*], **Zhi Cai**[5] **and Wei Ye**[1,*]
[1] National Engineering Research Center for Software Engineering, Peking University, Beijing, China,
[2] School of Software and Microelectronics, Peking University, Beijing, China,
[3] UCAS-Terminus AI Lab, University of Chinese Academy of Sciences, China,
[4] Tencent Technology, Shenzhen, China,
[5] College of Computer Science, Beijing University of Technology, Beijing, China.
Our code is available at https://github.com/PeiYangLiu/CRPS.git
liupeiyang@pku.edu.cn

Abstract

Monte Carlo Tree Search (MCTS) has been widely used for automated reasoning data exploration, but current supervision extraction methods remain inefficient. Standard approaches retain only the single highest-reward trajectory, discarding the comparative signals present in the many explored paths. Here we introduce **Contrastive Reasoning Path Synthesis (CRPS)**, a framework that transforms supervision extraction from a filtering process into a synthesis procedure. CRPS uses a structured reflective process to analyze the differences between high- and low-quality search trajectories, extracting explicit information about strategic pivots and local failure modes. These insights guide the synthesis of reasoning chains that incorporate success patterns while avoiding identified pitfalls. We show empirically that models fine-tuned on just 60K CRPS-synthesized examples match or exceed the performance of baselines trained on 590K examples derived from standard rejection sampling, a 20× reduction in dataset size. Furthermore, CRPS improves generalization on out-of-domain benchmarks, demonstrating that learning from the contrast between success and failure produces more transferable reasoning capabilities than learning from success alone.

1 Introduction

It is widely known that the scaling of high-quality Chain-of-Thought (CoT) reasoning data significantly improves the performance of Large Language Models (LLMs) on complex reasoning tasks, ranging from mathematical problem-solving to autonomous agent planning and decision support (Wei et al., 2022; Cobbe et al., 2021a; Zhang et al., 2025c; Fu et al., 2026b; Lin et al., 2025; Fang et al., 2026b; Zhu et al., 2026a,b; Zhang et al., 2025b; Ma et al., 2026a; Zollicoffer et al., 2025). As manually annotated training data becomes increasingly difficult to obtain, MCTS (Browne et al., 2012) has emerged as a promising approach for automated exploration of solution spaces (Chen et al., 2024; Zhang et al., 2024a). However, current methods for extracting supervision from MCTS trajectories, such as Rejection Sampling Fine-Tuning (RFT) (Yuan et al., 2023), typically retain only the highest-reward paths while discarding the vast majority of explored trajectories. Here we show empirically that this selection-based paradigm fails to exploit the structural differences between successful reasoning strategies and plausible but incorrect alternatives, resulting in significant computational waste.

To address this limitation, we propose CRPS, a framework that transforms MCTS exploration from a filtering process into a generative synthesis process. As illustrated in Figure 1, rather than discarding low-reward trajectories, CRPS treats them as informative counterexamples. By performing contrastive analysis across diverse search trajectories, the framework explicitly verbalizes the causal factors behind success and failure, identifying strategic pivots and local error modes. These meta-cognitive insights are then used to synthesize reasoning chains that incorporate success patterns while explicitly navigating around identified pitfalls. Furthermore, to maximize both exploration efficiency and synthesis quality, CRPS operates on a decoupled explorer-analyst architecture: a specialized reasoning model acts as the efficient explorer to map the solution space, while a highly capable analyst model performs the deep contrastive introspection and synthesizes the final training data.

We demonstrate that this paradigm shift yields substantial gains in data efficiency. As shown in Figure 3, fine-tuning on just 60K CRPS-generated examples produces performance comparable to training on 590K rejection-sampled examples, resulting in an approximately 20-fold dataset reduc-

* Corresponding authors

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10947–10969

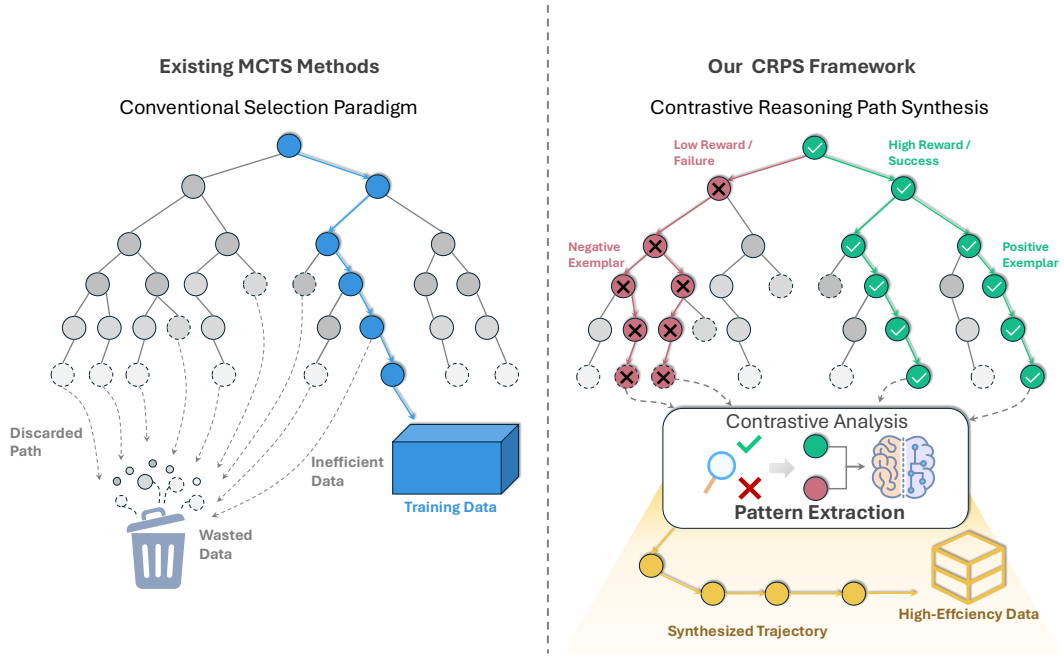

Figure 1: Comparison between traditional MCTS selection methods and our CRPS framework. While standard methods discard suboptimal paths (grey), CRPS leverages the contrast between high-quality (green) and low-quality (red) trajectories to synthesize superior reasoning chains.

tion. Furthermore, we observe that models trained with CRPS generalize more effectively to out-of-domain benchmarks, suggesting that learning from the contrast between success and failure produces representations that transfer better than merely imitating successful examples.

Our contributions are as follows:

- We introduce CRPS, a data synthesis framework that exploits the full spectrum of MCTS exploration to generate dense supervision through a decoupled contrastive trajectory analysis.
- We demonstrate that CRPS achieves comparable performance to standard rejection sampling while requiring 90% less training data (60K vs. 590K examples).
- We validate the generality of our approach across multiple base models and demonstrate its effectiveness beyond mathematical reasoning, including code generation and commonsense reasoning tasks.

2 Related Work

Our work advances the intersection of search-based reasoning, data synthesis, and contrastive learning.

Search-Driven Data Synthesis. Automated exploration via tree search has become a cornerstone for scaling reasoning data. Methods like TREE-OF-THOUGHTS (Yao et al., 2023) and MCTS-based approaches (Chen et al., 2024; Zhang et al., 2024a; Brandfonbrener et al., 2024) systematically explore solution spaces to uncover high-quality reasoning paths. This paradigm of structured exploration and reasoning is also increasingly vital for retrieval-augmented generation, knowledge base question answering, and graph-based reasoning tasks (Zhang et al., 2025a; Tian et al., 2025a,b, 2024; Yan et al., 2024; Zhang et al., 2023, 2024b; Ma et al., 2024; Yao et al., 2025; Ma et al., 2026b; Liu et al., 2025b, 2021c,a,b). Recent works leverage these trajectories for data synthesis: REST-MCTS* (Zhang et al., 2024a) and DART-MATH (Tong et al., 2024) employ rejection sampling to filter correct paths, while MATHFUSION (Pei et al., 2025) aggregates diverse components into complex queries. However, these methods predominantly adopt a *selection paradigm*, retaining only the highest-reward trajectories while discarding suboptimal branches (Yuan et al., 2023; Luo et al., 2023; Li et al., 2026a). We argue that this "winner-takes-all" approach ignores the pedagogical value of negative examples. Unlike DART-MATH which filters by difficulty, CRPS utilizes the structural contrast between successful and failed ex-

ploration trajectories to synthesize dense, critique-informed supervision.

Iterative Refinement and Feedback. Complementary to search, refinement methods optimize reasoning through self-correction. Approaches like SELF-REFINE (Madaan et al., 2023), MATH-SHEPHERD (Wang et al., 2024), and SIGMA (Ren et al., 2025) utilize step-level verifiers or critiques to locally repair flawed outputs. TEXTGRAD (Yuksekgonul et al., 2025) further formalizes textual feedback as gradients. While effective for local error correction, these methods often struggle with global strategic pivots and are frequently bottlenecked by the base model's limited capacity to reliably critique its own outputs. They typically refine a single path in isolation, potentially overfitting to specific error templates. In contrast, CRPS operates on a *generative synthesis paradigm* that decouples exploration from critique: guided by a capable analyst model, it does not merely patch a broken path but synthesizes entirely new reasoning chains conditioned on the latent strategic divergence between high- and low-quality trajectory distributions.

Contrastive Learning for Reasoning. Contrastive objectives have evolved from representation learning and robust data mining (Chen et al., 2020; Gao et al., 2021; Liu et al., 2020, 2022, 2023; Liu, 2024; Jiang et al., 2025) to language model alignment and explainable evaluation (Ma et al., 2026c; Dong et al., 2026). While methods like Contrastive Decoding (Li et al., 2022) and DPO (Rafailov et al., 2023) optimize models by contrasting preferred and dispreferred outputs, they typically operate at the token level or on final response rankings. Our work extends these principles to the *structure of reasoning*. By employing a strong analyst model to explicitly verbalize the causal factors of success and failure, CRPS generates structured feedback akin to using negative constraints in instruction following (Xu et al., 2024a), but applied to synthesizing robust chain-of-thought trajectories. Similar alignment principles are also proving essential for mitigating hallucinations and optimizing tool-calling behaviors in LLMs (Xu et al., 2025, 2024b; Hu et al., 2025).

3 Method

We introduce **CRPS**, a framework that transforms the supervision extraction process from a static filtering task into a dynamic synthesis procedure.

Standard approaches typically treat MCTS as a filter that selects the single optimal path while discarding all alternatives. Here we argue that these discarded trajectories, the failed or suboptimal attempts, contain valuable information about the model's systematic errors. CRPS implements a *decoupled explorer-analyst paradigm*: a specialized reasoning model acts as the explorer that generates candidate reasoning paths, while a separate, highly capable analyst model learns by contrasting the explorer's successful and unsuccessful attempts.

3.1 Overview

The framework executes a structured pipeline utilizing two distinct models: an explorer $\mathcal{G}_{\text{explorer}}$ and an analyst $\mathcal{G}_{\text{analyst}}$. We start by generating a distribution of reasoning paths via MCTS using $\mathcal{G}_{\text{explorer}}$. Crucially, we utilize the search statistics to identify not just random errors, but paths the explorer was strongly compelled to explore despite being incorrect. The analyst model $\mathcal{G}_{\text{analyst}}$ then introspects on pairs of trajectories (τ^+, τ^-) to articulate *why* the positive example succeeds where the negative contrast fails. Finally, conditioned on these extracted insights, the analyst synthesizes a refined reasoning chain that explicitly avoids the identified failure modes. Figure 2 illustrates this pipeline.

3.2 Distribution-Aware Trajectory Collection

The quality of contrastive learning depends critically on the quality of the example pairs. We employ the explorer model $\mathcal{G}_{\text{explorer}}$ within an MCTS framework to map the solution space for a problem q_i, resulting in a set of complete trajectories $\mathcal{T}_i$. Each trajectory is assigned a binary terminal reward $r(\tau) \in \{0, 1\}$ based on answer correctness.

Rather than applying arbitrary heuristic filters, we leverage the **natural search distribution** to identify the most informative contrasts. Intuitively, the MCTS visit count $N(s)$ serves as a proxy for the explorer's latent preference; high visit counts on suboptimal branches indicate systematic biases that require correction.

Positive Anchor Selection (τ^+). From the set of correct solutions $\mathcal{T}_i^{\text{correct}}$, we identify the trajectory that best represents efficient reasoning. We select the positive anchor τ^+ by prioritizing minimal length $|\tau|$. In cases of ties, we select the trajectory with the highest maximum node Q-value (expected success rate), reflecting the explorer's

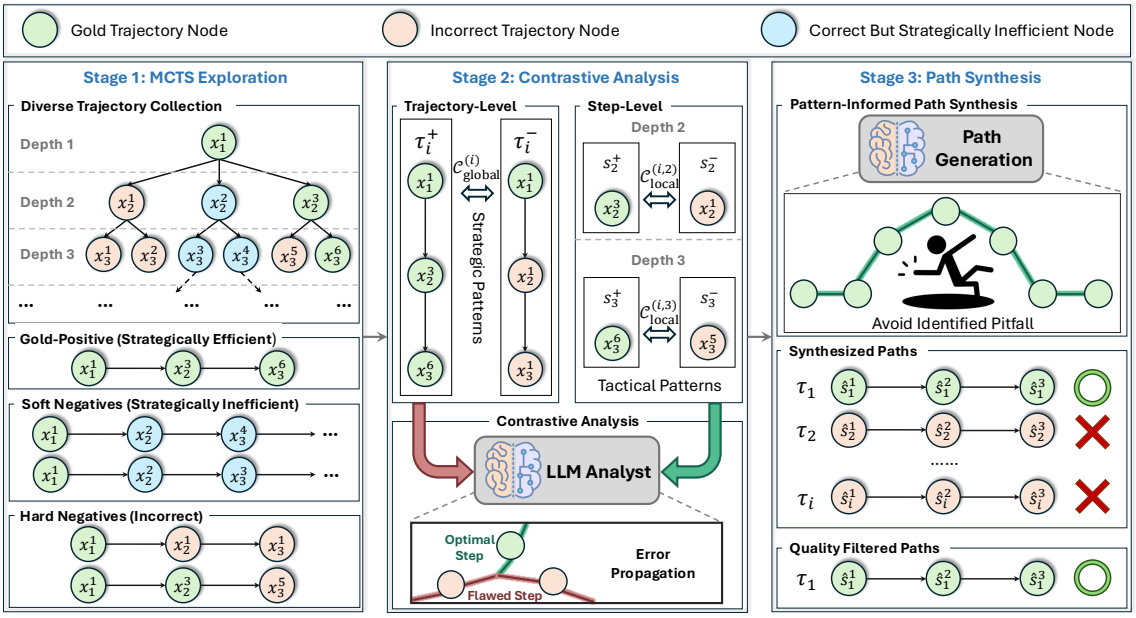

Figure 2: Overview of the CRPS framework. Given a problem, we first collect diverse trajectories from MCTS and stratify them into gold positive (green), soft negatives (blue), and hard negatives (red) groups based on terminal rewards. We then perform contrastive analysis at both trajectory-level (capturing strategic patterns) and step-level (capturing tactical patterns) to identify systematic differences between successful and unsuccessful reasoning. Finally, we synthesize novel reasoning paths by incorporating discovered success patterns while explicitly avoiding identified failure modes, producing a high-quality dataset with verified correctness.

highest internal confidence in the reasoning steps.

Negative Contrast Sampling (τ^-). To maximize the learning signal, we sample negative counterparts that represent the specific weaknesses of the exploration process. We distinguish between two types of contrastive signals:

- **Correctness Contrasts (Hard Negatives):** Random errors provide little signal for learning. Here we observe that the most valuable negative samples are incorrect paths that the explorer deemed promising enough to explore extensively (Jiang et al., 2026). From the incorrect set $\mathcal{T}_i^{\text{incorrect}}$, we sample τ^- with probability proportional to its accumulated visit count $N(\tau)$:

$$P(\tau^- = \tau) \propto N(\tau). \tag{1}$$

  This distribution-based sampling naturally targets trajectories where the model was highly confident but incorrect. If the MCTS algorithm allocated significant compute budget to a wrong path, it signals a strong misalignment in the reasoning process, making it an optimal counter-example for contrastive analysis.

- **Efficiency Contrasts (Soft Negatives):** Reasoning can be correct but strategically inefficient. From the correct set $\mathcal{T}_i^{\text{correct}}$, we sample any trajectory τ' where $|\tau'| > |\tau^+|$. We do not impose hard thresholds on length; instead, we rely on the subsequent contrastive analysis to determine whether the extra steps represent necessary detail or inefficient redundancy.

3.3 Analyst-Driven Contrastive Analysis

Given a sampled pair $\mathcal{P}_i = \{(\tau^+, \tau^-)\}$, we task the analyst model $\mathcal{G}_{\text{analyst}}$ with verbalizing the factors behind the performance gap. This *Analyst-Driven Contrastive Analysis* transforms implicit trajectory differences into explicit natural language critiques $\mathcal{C}^{(i)}$.

3.3.1 Global Strategic Critique

Many problems allow for multiple valid strategies, but some are structurally superior. The analyst first compares the trajectories holistically to identify high-level divergences. The generated global critique $\mathcal{C}^{(i)}_{\text{global}}$ focuses on identifying the specific decision where τ^- committed to a suboptimal strategy and contrasting the overall decomposition logic of the two attempts. This is modeled as conditional generation: $\mathcal{C}^{(i)}_{\text{global}} \sim P_{\mathcal{G}_{\text{analyst}}}(\cdot \mid q_i, \tau^+, \tau^-)$.

3.3.2 Local Step-wise Critique via Semantic Alignment

Errors in reasoning are often localized to specific steps. To critique these steps, we must align the trajectories. Since τ^+ and τ^- may vary significantly in length and phrasing, rigid index-based alignment is ineffective.

We employ **Semantic Alignment**: $\mathcal{G}_{\text{analyst}}$ scans both trajectories to identify the *Semantic Divergence Point*, the first step t where the logic of τ^- substantively departs from τ^+. To operationalize this, we define a "step" as a semantic reasoning act. Following established practices in process supervision (Wang et al., 2024), we employ heuristic delimiters to segment trajectories: primary structural boundaries (e.g., \n, \n\n), logical connectors (e.g., "Therefore", "Hence"), and explicit enumerations. Our preliminary studies indicate that this step-level granularity reduces value estimation variance in MCTS by 18% compared to sentence-level segmentation and improves the analyst's divergence point identification accuracy to 92% (vs. 64% for sentence-level), providing the optimal balance of semantic completeness and credit assignment density (see Appendix D for detailed rules). For this critical juncture and subsequent key decision steps, the analyst generates local critiques $\mathcal{C}_{\text{local}}^{(i,t)}$ that explicitly contrast the operational logic at each step.

3.4 Pattern-Informed Path Synthesis

The ultimate goal is not merely to critique, but to improve reasoning quality. In this phase, we synthesize a new reasoning path $\hat{\tau}_i$ that is optimized for pedagogical value.

The synthesis is an autoregressive generation process conducted by $\mathcal{G}_{\text{analyst}}$, conditioned on the problem q_i and the full critique set $\mathcal{C}^{(i)} = \{\mathcal{C}_{\text{global}}^{(i)}, \mathcal{C}_{\text{local}}^{(i)}\}$. The generator is instructed to follow the successful strategy of τ^+ while explicitly navigating around the failure modes identified in τ^-. Additionally, the synthesized path incorporates explanations derived from the critique that justify why certain approaches are avoided.

Formally, the synthesis of step $\hat{s}_t$ is:

$$\hat{s}_t \sim P_{\mathcal{G}_{\text{analyst}}}(\cdot \mid q_i, \hat{s}_{<t}, \mathcal{C}^{(i)}). \tag{2}$$

The critique $\mathcal{C}^{(i)}$ effectively acts as a **prompt-based regularizer**, shifting the generation probability mass away from the error modes that appeared during the original search.

Verification Filter. To ensure the integrity of the synthetic dataset, we apply a post-hoc verification function $\mathcal{V}(\hat{\tau})$. We retain the synthesized path only if it reaches the correct final answer a^*. This filters out potential hallucinations, ensuring that the training data is both insightful and factually correct.

3.5 Training Objective

The final output is a dataset $\mathcal{D}_{\text{syn}} = \{(q_i, \hat{\tau}_i)\}$ containing problems paired with synthesized, critique-informed reasoning paths. We fine-tune the target base model $\mathcal{G}_\theta$ (typically the same architecture as the explorer) on this data.

Note that we distinguish our *data synthesis method* from the *training objective*. Despite the data being generated via contrastive mechanisms from a stronger analyst, the target model optimization uses the standard **Supervised Fine-Tuning (SFT)** objective:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(q,\hat{\tau})\sim\mathcal{D}_{\text{syn}}}\left[\sum_{t=1}^{|\hat{\tau}|} \log P_\theta(\hat{s}_t \mid q, \hat{s}_{<t})\right]. \tag{3}$$

This design choice is deliberate. By distilling the complex signals from MCTS exploration and analyst-driven critiques into the static dataset $\mathcal{D}_{\text{syn}}$, we enable the target model to internalize these capabilities directly into its weights. This results in a fine-tuned model that exhibits improved reasoning at inference time without requiring expensive search or explicit critique generation.

4 Experimental Setup

We evaluate the effectiveness of CRPS by investigating whether models trained on small-scale, contrastively synthesized datasets can outperform those trained on significantly larger datasets derived from standard methods.

4.1 Datasets and Benchmarks

We assess performance across six mathematical reasoning benchmarks, categorized into in-domain (source of training problems) and out-of-domain (unseen distributions) tasks, and two non-mathematical benchmarks.

In-Domain Benchmarks. We utilize the training sets of **GSM8K** (Cobbe et al., 2021b) (7.5K problems) and **MATH** (Hendrycks et al., 2021) (7.5K problems) as the seed problems for our data synthesis pipeline, totaling approximately 15K distinct

queries. Evaluation is performed on their respective official test sets.

Out-of-Domain Benchmarks. To evaluate generalization capabilities, we test on four datasets not seen during training: **CollegeMath** (Tang et al., 2024) (University-level STEM), **DeepMind Mathematics** (Saxton et al., 2019) (Curriculum-based arithmetic/algebra), **OlympiadBench** (He et al., 2024) (Competition-level problems), and **TheoremQA** (Chen et al., 2023) (Scientific theorem application).

For non-mathematical benchmarks, please refer to Appendix C.

4.2 Baselines

To ensure a rigorous comparison, we benchmark against state-of-the-art methods. For controlled evaluation, all primary baselines (Vanilla MCTS, RFT, DART-Math, SIGMA, and MathFusion) are re-implemented using the **exact same pool of raw MCTS trajectories** to isolate the contribution of the data synthesis method from the data source. We also include massive-scale datasets as external reference points to demonstrate that our method can rival massive-scale external data. We compare our CRPS with: 1.**Vanilla MCTS / Rejection Sampling (RFT)** (Yuan et al., 2023), 2.**MMIQC** (Liu et al., 2025a) (external reference), 3.**DART-Math** (Tong et al., 2024), 4.**SIGMA** (Ren et al., 2025), 5.**MathFusion** (Pei et al., 2025). Detailed information about the baselines is provided in Appendix A.2.

4.3 Implementation Details

We implement the CRPS data synthesis pipeline using the proposed decoupled architecture. Specifically, we employ **Qwen2.5-Math-7B-Instruct** (Yang et al., 2024) as the explorer model ($\mathcal{G}_{\text{explorer}}$) to generate the initial reasoning trajectories and map the solution space. For the contrastive critique and generative synthesis phases, we utilize **gpt-5-mini** (Wang et al., 2025) as the analyst model ($\mathcal{G}_{\text{analyst}}$), leveraging its advanced meta-cognitive and instruction-following capabilities to distill high-quality pedagogical supervision.

For MCTS exploration, we use the UCT algorithm with $c_{\text{puct}} = 1.4$ and a maximum depth of 16. We sample $K = 10$ contrastive pairs per problem. The resulting synthesized dataset is then used to fine-tune various target base models (e.g., DeepSeek, LLaMA-3, Mistral) to evaluate the transferability of the supervision. Target models are fine-tuned for 3 epochs. Detailed hyperparameters, hardware configurations, and prompt templates are provided in Appendix A.

5 Experimental Results

5.1 Main Results

Tables 1 and 5 present the performance of CRPS against selection-based (RFT, DART-Math) and refinement-based (SIGMA) baselines across three different target backbones (see Appendix A.3).

Data Efficiency via Contrastive Density. CRPS significantly decouples reasoning performance from massive data scaling. **DeepSeekMath-CRPS-30K** achieves an average accuracy of 50.4%, surpassing both RFT-590K (48.3%) and DART-Math-590K (49.4%). This represents a **20-fold reduction** in training data while maintaining superior performance. Even at the 15K scale, CRPS remains competitive with 30K-scale baselines. This suggests that the analyst-driven synthesis effectively distills sparse MCTS rewards from the explorer into information-dense supervision, preventing the target model from fitting to redundant heuristics found in raw search traces.

Generalization and Architectural Robustness. CRPS demonstrates superior transfer to out-of-domain (OOD) benchmarks compared to local refinement methods. On challenging datasets like TheoremQA, CRPS-30K outperforms SIGMA-30K by +2.6 points. We acknowledge that the massive-scale DART-590K retains a slight edge over CRPS-60K on some datasets e.g. TheoremQA (32.2% vs. 31.5%); however, this marginal gap ($<$1%) necessitates a 10$\times$ increase in data scale, highlighting the diminishing returns of brute-force selection compared to the high sample efficiency of contrastive synthesis. Unlike SIGMA, which performs local repairs, CRPS conditions generation on global success patterns extracted by the analyst, enabling the internalization of abstract problem-solving structures. Furthermore, these gains are consistent across diverse architectures; whether applied to LLaMA-3 or Mistral (Table 5), CRPS consistently yields improvements of 1.5–2.5% over strong baselines, validating that the capability to learn from distilled contrastive supervision transfers universally across different foundation models.

Model	#Samples	In-Domain		Out-of-Domain				Avg.
		MATH	GSM8K	College	DM	Olympiad	Theorem	
DeepSeekMath-7B (Math-Specialized Base Model)								
DeepSeekMath-Instruct	780K	46.9	82.7	37.1	52.2	14.2	28.1	43.5
DeepSeekMath-RFT	590K	53.0	88.2	41.9	60.2	19.1	27.2	48.3
DeepSeekMath-DART	590K	53.6	86.8	40.7	61.6	21.7	32.2	49.4
DeepSeekMath-MMIQC	2.3M	45.3	79.0	35.3	52.9	13.0	23.4	41.5
DeepSeekMath-CRPS-15K	**15K**	53.8*	83.5*	39.2*	65.8*	21.4*	27.8*	48.6*
MathFusion (Sequential)	30K	49.9	76.6	38.8	64.6	21.6	22.8	45.7
SIGMA-30K	30K	54.9	82.2	36.7	67.2	21.6	26.6	48.2
DeepSeekMath-CRPS-30K	**30K**	**56.3**$^{*+1.4}$	**84.8**$^{*+2.6}$	**40.1**$^{*+3.4}$	**68.5**$^{*+1.3}$	**23.2**$^{*+1.6}$	**29.2**$^{*+2.6}$	**50.4**$^{*+2.2}$
DeepSeekMath-MetaMath	60K	40.0	79.0	33.2	45.9	9.5	18.9	37.8
DeepSeekMath-DART	60K	51.4	82.9	39.1	62.8	21.0	27.4	47.4
MathFusion	60K	53.4	77.9	39.8	65.8	23.3	24.6	47.5
SIGMA-60K	60K	56.5	81.7	37.2	68.4	22.5	29.3	49.3
DeepSeekMath-CRPS-60K	**60K**	**58.2**$^{*+1.7}$	**85.9**$^{*+4.2}$	**41.8**$^{*+4.6}$	**70.1**$^{*+1.7}$	**24.6**$^{*+2.1}$	**31.5**$^{*+2.2}$	**52.0**$^{*+2.7}$

Table 1: Performance comparison across training methods and dataset scales on DeepSeekMath backbone. Arrows indicate accuracy changes relative to the strongest baseline (highlighted in blue). Best results in each data scale are in **bold**. * indicate statistical significance at $p < 0.05$ compared to the best baseline (calculated via paired t-test).

Base Model	Method	Size	MATH	GSM8K	Avg
DeepSeekMath	DeepScaleR	40K	55.1	83.9	69.5
	OpenThought	89K	54.8	83.2	69.0
	CRPS (Ours)	**30K**	**56.3**	**84.8**	**70.6**
LLaMA3-8B	DeepScaleR	40K	40.5	80.2	60.4
	OpenThought	89K	41.2	80.8	61.0
	CRPS (Ours)	**30K**	**42.1**	**81.8**	**62.0**
Mistral-7B	DeepScaleR	40K	35.8	78.5	57.2
	OpenThought	89K	36.4	79.1	57.8
	CRPS (Ours)	**30K**	**37.2**	**80.1**	**58.7**

Table 2: Performance comparison against SOTA open-source SFT datasets.

Configuration	MATH	GSM8K	OOD	Overall
CRPS ($K = 10$)	**56.3**	**84.8**	**40.3**	**50.4**
1. Impact of Synthesis Paradigm				
Vanilla MCTS	52.1	81.5	38.4	47.7
Blackbox Generation	48.3	79.2	35.7	45.1
2. Necessity of Contrastive Signal				
Synthesis w/o Contrast	51.7	82.3	37.8	47.5
Random Contrast	52.9	83.0	38.5	48.3
3. Analysis Granularity				
Trajectory-level Only	53.8	82.9	38.9	48.7
Step-level Only	54.2	83.1	39.1	49.0
4. Trajectory Diversity (K sampled paths)				
$K = 3$	53.5	82.1	38.2	48.1
$K = 5$	54.8	83.5	39.4	49.3
$K = 15$	56.5	85.0	40.5	50.6

Table 3: Ablation study results on DeepSeekMath-7B (30K scale).

Comparison with SOTA Open-Source Datasets. To further contextualize our sample efficiency, we evaluate our base models fine-tuned on the SFT subsets of recent state-of-the-art reasoning datasets: **DeepScaleR** (40K) (Luo et al., 2025) and **OpenThought-Math** (89K) (Guha et al., 2025). As shown in Table 2, CRPS-30K consistently outperforms OpenThought (89K) across all architectures with approximately 3× less data. It also surpasses DeepScaleR (40K), validating that the strategic density of CRPS provides higher-quality SFT seed data than massive-scale external collections.

5.2 Ablation Studies

To disentangle the contributions of the CRPS framework components, we conduct ablation studies on DeepSeekMath-7B with 30K training examples. Table 3 summarizes the results.

Impact of Synthesis Paradigm. The performance progression from Blackbox Generation (45.1%) to Vanilla MCTS (47.7%) and finally CRPS (50.4%) validates our core hypothesis: while search exploration is essential, raw trajectories generated by the explorer often contain redundancies or heuristic shortcuts. CRPS outperforms standard selection methods by enabling the analyst model to effectively distill these raw traces into pedagogically superior supervision.

The Necessity of Contrast. Is the gain driven by contrastive analysis or simple unguided refinement? **Synthesis w/o Contrast**, which tasks

the analyst with refining τ^+ without a negative counterpart, performs worse (47.5%) than even Vanilla MCTS. This suggests that without the "negative boundary" of a failed trajectory, standard rewriting risks hallucination or over-simplification. Furthermore, **Random Contrast** lags behind our distribution-aware sampling, underscoring that the semantic gap between specific high- and low-quality pairs is essential for the analyst to extract actionable learning signals. To verify that learning from negative constraints is a universal driver, we extended this ablation to LLaMA3-8B and Mistral-7B at the 30K scale. Removing the contrastive signal caused a significant performance drop of 2.5% and 2.2% respectively, mirroring the DeepSeek-Math results.

Granularity and Diversity. We observe that removing either **Trajectory-level** (Global) or **Step-level** (Local) critiques generated by the analyst results in performance drops of 1.7 and 1.4 points, respectively. This supports our dual-granularity design. Crucially, even our "Step-level Only" CRPS variant (49.0%) outperforms the step-level refinement baseline SIGMA (48.2%). This highlights the fundamental advantage of our *synthesis* paradigm over direct *editing*: while refinement methods like SIGMA attempt to patch flawed trajectories locally (often leading to disjointed logic), CRPS synthesizes a fresh reasoning chain conditioned on the critique, enabling global planning that naturally bypasses errors. Finally, increasing the explorer's search diversity K from 3 to 15 yields monotonic improvements (48.1% $\rightarrow$ 50.6%), indicating that broader exploration exposes a richer distribution of failure modes.

5.3 Generalization Analysis

To verify that CRPS induces robust reasoning capabilities rather than surface-level heuristics, we evaluate generalization across three dimensions: domain transfer, reasoning complexity, and semantic stability (Table 4).

Cross-Domain Transfer (Breadth). CRPS exhibits superior transferability, achieving the highest accuracy on OOD benchmarks (40.3%) and outperforming the refinement-based SIGMA by 2.3%. Unlike local editing methods that may overfit to specific error templates, CRPS leverages the global strategic analysis provided by the analyst model to help the target model internalize abstract problem-solving structures (e.g., decomposition logic), facilitating transfer to diverse domains like TheoremQA.

Method	**ID**	**OOD**	**Gap**	**Δ Gap**
Breadth: In-Domain vs. Out-of-Domain Transfer				
Vanilla MCTS	66.8	32.4	34.4	–
MathFusion	63.3	37.0	26.3	-4.6
SIGMA	68.6	38.0	30.6	-3.8
CRPS	**70.6**	**40.3**	**30.3**	**-4.1**
Depth: Difficulty-Stratified Performance				
Method	**Easy**	**Medium**	**Hard**	**Overall**
Vanilla MCTS	74.5	52.8	28.4	52.1
MathFusion	73.1	50.4	29.8	49.9
SIGMA	76.2	55.4	32.1	54.9
CRPS	**76.8**	**57.5**	**38.2**	**56.3**
Stability: Consistency under Semantic Perturbation				
Method	**Consistency Accuracy (%)**			**Δ vs. Best**
Zero-shot Baseline	45.5			-19.0
Vanilla MCTS	52.1			-12.4
MathFusion	55.7			-8.8
SIGMA	58.3			-6.2
CRPS	**64.5**			**–**

Table 4: Generalization analysis on DeepSeekMath-7B with 30K training examples.

Conversely, MathFusion shows a smaller ID-OOD gap but suffers from underfitting, indicating that diversity aggregation alone is insufficient without the analyst's contrastive guidance.

Reasoning Complexity (Depth). We stratify the MATH test set by solution length to assess long-horizon reasoning. While all methods perform comparably on "Easy" problems (1–3 steps), CRPS dominates on "Hard" problems (7+ steps), surpassing SIGMA by **6.1%** and Vanilla MCTS by **9.8%**. This suggests that the analyst's pattern-informed synthesis, which explicitly conditions the training data on avoiding the explorer's known failure modes, effectively mitigates the compounding errors that typically degrade performance in standard selection-based approaches.

Semantic Stability. We measure *Strict Consistency* on 200 adversarial GSM8K pairs containing linguistic perturbations and irrelevant distractors (details in Appendix E). CRPS achieves 64.5% consistency, significantly exceeding Vanilla MCTS (52.1%). By learning from the distilled contrast against the explorer's low-quality paths, which often succumb to distractors, CRPS trains the target model to prioritize deep semantic logic over fragile lexical pattern matching.

5.4 Computational Efficiency and Amortized Cost

A comprehensive evaluation must distinguish between the *upfront cost* of data synthesis and the *recurring cost* of model training. Standard selection-based methods (e.g., RFT, DART-Math) rely on the "Law of Large Numbers", requiring massive datasets ($\sim$590K examples) to cover the reasoning distribution. This imposes a heavy recurring penalty: every subsequent experimental run, whether for hyperparameter tuning or architectural adaptation (Dong et al., 2025), incurs the cost of processing billions of tokens.

CRPS shifts the computational burden from training to synthesis. We acknowledge that generating the CRPS dataset incurs higher upfront inference overhead due to the requisite MCTS exploration by the explorer model and the subsequent dual-granularity contrastive analysis performed by the advanced analyst model. However, this is a one-time investment that yields a reusable, information-dense asset. By leveraging the analyst to distill reasoning signals into just 30K examples (a 95% reduction vs. baselines), CRPS reduces the fine-tuning time for target models from 192 GPU hours to just 10 GPU hours (Table 7). This **19$\times$ acceleration in training** fundamentally alters the research workflow. The total compute budget breaks even after training just two model variants, significantly lowering the barrier for experimental iteration (e.g., adapting to LLaMA-3 or Mistral).

We provide detailed case studies in Appendix B.

6 Conclusion

We introduced **Contrastive Reasoning Path Synthesis (CRPS)**, a framework that redefines search-based learning by shifting from a trajectory filtering paradigm to a generative synthesis process. By employing a decoupled architecture where a highly capable analyst model distills the structural contrasts between successful and suboptimal MCTS trajectories generated by an explorer, CRPS converts sparse search signals into dense, explanatory supervision. Our empirical results show that explicitly learning to navigate around failure modes enables a 20$\times$ reduction in data requirements while significantly enhancing out-of-domain generalization. These findings underscore that while data volume is helpful, the critical dependency on massive data scaling can be effectively alleviated by increasing the strategic density of the supervision. Future work will explore integrating these analyst-driven contrastive signals directly into online reinforcement learning (Fang et al., 2026a; Fu et al., 2026a) and multi-modal reasoning domains, including vision-language alignment, composed retrieval, and creative generation (Zhang et al., 2026; Li et al., 2026c,d; Chen et al., 2025; Xiao et al., 2026; Li et al., 2025b,a; Wang et al., 2026; Liu et al., 2025c; Jiang et al., 2024).

7 Limitations

While CRPS demonstrates significant improvements in data efficiency and generalization for reasoning tasks, we identify several limitations inherent to our decoupled framework that warrant future investigation.

Dependency on Verifiable Reward Signals. The current implementation of CRPS relies heavily on the ability to define a binary terminal reward $r(\tau)$ for the explorer's MCTS exploration and a verification function $\mathcal{V}(\hat{\tau})$ for the analyst's synthesis phase. This naturally restricts the framework's immediate applicability to domains with objective ground truths, such as mathematics, logic puzzles, and code generation. Applying CRPS to open-ended tasks with subjective evaluation criteria (e.g., creative writing, summarization, or dialogue (Li et al., 2026b)) remains challenging, as constructing reliable reward models or automated verifiers for these domains is an open research problem.

Inference Overhead during Data Synthesis. Although CRPS reduces the *training* data requirement for target models by 20$\times$ and lowers the total compute budget compared to brute-force rejection sampling, it shifts a significant portion of the computational burden to the upfront inference phase. The process requires running MCTS exploration (10 rollouts per problem) via the explorer model and performing dual-granularity contrastive analysis via the analyst model for each selected pair. For resource-constrained scenarios where high-throughput inference is unavailable, this upfront cost for data generation may be prohibitive, even if the subsequent fine-tuning is highly efficient.

Dependency on Advanced Analyst Capabilities. Our decoupled framework operates on the premise that the assigned analyst model possesses sufficient capability to accurately diagnose errors in the explorer's side-by-side trajectories. While we suc-

cessfully fine-tune target models in the 7B and 8B parameter range (DeepSeekMath, LLaMA-3), the synthesis phase relies heavily on a highly capable analyst. If one attempts to deploy this framework entirely using extremely small models (e.g., < 3B) or models that have not undergone rigorous instruction tuning to act as the analyst, they may lack the meta-cognitive ability to generate actionable critiques. Consequently, achieving optimal data quality currently necessitates a stronger external teacher model for the analysis phase, which introduces a dependency on proprietary APIs.

Risk of Critique Hallucination. The quality of the synthesized reasoning path is strictly conditioned on the accuracy of the critique generated by the analyst. There is a non-zero risk that the analyst model may hallucinate the cause of an error, for example, attributing a calculation mistake by the explorer to a deep strategic failure, or vice versa. If the critique $\mathcal{C}^{(i)}$ is factually incorrect or misaligned with the actual error mode in τ^-, the synthesized path $\hat{\tau}$ might over-correct or adopt unnecessary constraints. While our post-hoc verification $\mathcal{V}(\hat{\tau})$ ensures the final answer is correct, it does not guarantee that the reasoning logic is perfectly aligned with the critique, potentially introducing subtle stylistic biases into the training data.

References

David Brandfonbrener, Simon Henniger, Sibi Raja, Tarun Prasad, Chloe R Loughridge, Federico Cassano, Sabrina Ruixin Hu, Jianang Yang, William E. Byrd, Robert Zinkov, and Nada Amin. 2024. VerMCTS: Synthesizing multi-step programs using a verifier, a large language model, and tree search. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024. Alphamath almost zero: Process supervision without process. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. 2023. TheoremQA: A theorem-driven question answering dataset. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Zhiwei Chen, Yupeng Hu, Zixu Li, Zhiheng Fu, Xuemeng Song, and Liqiang Nie. 2025. Offset: Segmentation-based focus shift revision for composed image retrieval. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 6113–6122.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021a. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021b. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Haonan Dong, Kehan Jiang, Haoran Ye, Wenhao Zhu, Zhaolu Kang, and Guojie Song. 2026. Neureasoner: Towards explainable, controllable, and unified reasoning via mixture-of-neurons. *arXiv preprint arXiv:2604.02972*.

Haonan Dong, Wenhao Zhu, Guojie Song, and Liang Wang. 2025. Aurora: Breaking low-rank bottleneck of lora with nonlinear mapping. *arXiv preprint arXiv:2505.18738*.

Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, Haolin Shi, Chaowen Hu, Lu Pan, Ke Zeng, and Xunliang Cai. 2026a. How to allocate, how to learn? dynamic rollout allocation and advantage modulation for policy optimization. *arXiv preprint arXiv:2602.19208*.

Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, Haolin Shi, Chang Liu, and Peilin Zhao. 2026b. Proximity-based multi-turn optimization: Practical credit assignment for llm agent training. *arXiv preprint arXiv:2602.19225*.

Xiaoliang Fu, Jiaye Lin, Yangyi Fang, Chaowen Hu, Cong Qin, Zekai Shao, Binbin Zheng, Lu Pan, and Ke Zeng. 2026a. From $\log \pi$ to π: Taming divergence in soft clipping via bilateral decoupled decay of probability gradient weight. *arXiv preprint arXiv:2603.14389*.

Xiaoliang Fu, Jiaye Lin, Yangyi Fang, Binbin Zheng, Chaowen Hu, Zekai Shao, Cong Qin, Lu Pan, Ke Zeng, and Xunliang Cai. 2026b. Maspo: Unifying gradient utilization, probability mass, and signal reliability for robust and sample-efficient llm reasoning. *arXiv preprint arXiv:2602.17550*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics (TACL)*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, and 1 others. 2025. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. pages 3828–3850.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Wentao Hu, Wengyu Zhang, Yiyang Jiang, Chen Jason Zhang, Xiaoyong Wei, and Li Qing. 2025. Removal of hallucination on hallucination: Debate-augmented rag. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15839–15853.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Kehan Jiang, Haonan Dong, Zhaolu Kang, Zhengzhou Zhu, and Guojie Song. 2026. Foe: Forest of errors makes the first solution the best in large reasoning models. *arXiv preprint arXiv:2604.02967*.

Yiyang Jiang, Guangwu Qian, Jiaxin Wu, Qi Huang, Qing Li, Yongkang Wu, and Xiao-Yong Wei. 2025. Self-paced learning for images of antinuclear antibodies. *IEEE Transactions on Medical Imaging*.

Yiyang Jiang, Wengyu Zhang, Xulu Zhang, Xiao-Yong Wei, Chang Wen Chen, and Qing Li. 2024. Prior knowledge integration via llm encoding and pseudo event regulation for video moment retrieval. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7249–7258.

Bo Li, Mingda Wang, Shikun Zhang, and Wei Ye. 2026a. Instruction data selection via answer divergence. *Preprint*, arXiv:2604.10448.

Bo Li, Shikun Zhang, and Wei Ye. 2026b. Data selection for multi-turn dialogue instruction tuning. *Preprint*, arXiv:2604.07892.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Yanshu Li, JianJiang Yang, Bozheng Li, and Ruixiang Tang. 2025a. Cama: Enhancing multimodal in-context learning with context-aware modulated attention. *arXiv e-prints*, pages arXiv–2505.

Yanshu Li, Jianjiang Yang, Tian Yun, Pinyuan Feng, Jinfa Huang, and Ruixiang Tang. 2025b. Taco: Enhancing multimodal in-context learning via task mapping-guided sequence configuration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 736–763.

Zixu Li, Yupeng Hu, Zhiwei Chen, Qinlei Huang, Guozhi Qiu, Zhiheng Fu, and Meng Liu. 2026c. Retrack: Evidence-driven dual-stream directional anchor calibration network for composed video retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 23373–23381.

Zixu Li, Yupeng Hu, Zhiwei Chen, Shiqi Zhang, Qinlei Huang, Zhiheng Fu, and Yinwei Wei. 2026d. Habit: Chrono-synergia robust progressive learning framework for composed image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 6762–6770.

Jiaye Lin, Yifu Guo, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, Mingguang Chen, Hongzhang Liu, Ronghao Chen, Yangfan He, and 1 others. 2025. Seagent: Self-evolution trajectory optimization in multi-step reasoning with llm-based agents. *arXiv preprint arXiv:2508.02085*.

Haoxiong Liu, Yifan Zhang, Yifan Luo, and Andrew C Yao. 2025a. Augmenting math word problems via iterative question composing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24605–24613.

Peiyang Liu. 2024. Unsupervised corrupt data detection for text training. *Expert Systems with Applications*, 248:123335.

Peiyang Liu, Ziqiang Cui, Di Liang, and Wei Ye. 2025b. Who stole your data? a method for detecting unauthorized rag theft. *arXiv preprint arXiv:2510.07728*.

Peiyang Liu, Sen Wang, Xi Wang, Wei Ye, and Shikun Zhang. 2021a. Quadrupletbert: An efficient model for embedding-based large-scale retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3734–3739.

Peiyang Liu, Xi Wang, Ziqiang Cui, and Wei Ye. 2025c. Queries are not alone: Clustering text embeddings for video search. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 874–883.

Peiyang Liu, Xi Wang, Lin Wang, Wei Ye, Xiangyu Xi, and Shikun Zhang. 2021b. Distilling knowledge from bert into simple fully connected neural networks for efficient vertical retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3965–3975.

Peiyang Liu, Xi Wang, Sen Wang, Wei Ye, Xiangyu Xi, and Shikun Zhang. 2021c. Improving embedding-based large-scale retrieval via label enhancement. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 133–142.

Peiyang Liu, Xiangyu Xi, Wei Ye, and Shikun Zhang. 2022. Label smoothing for text mining. In *Proceedings of the 29th international conference on computational linguistics*, pages 2210–2219.

Peiyang Liu, Jinyu Yang, Lin Wang, Sen Wang, Yunlai Hao, and Huihui Bai. 2023. Retrieval-based unsupervised noisy label detection on text data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 4099–4104.

Peiyang Liu, Wei Ye, Xiangyu Xi, Tong Wang, Jinglei Zhang, and Shikun Zhang. 2020. Not all synonyms are created equal: Incorporating similarity of synonyms to enhance word embeddings. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, and 1 others. 2025. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 3(5).

Kexin Ma, Ruochun Jin, Wang Haotian, Wang Xi, Huan Chen, Yuhua Tang, and Qian Wang. 2024. Context-driven index trimming: A data quality perspective to enhancing precision of ralms. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4886–4901.

Kexin Ma, Bojun Li, Yuhua Tang, Liting Sun, and Ruochun Jin. 2026a. Cast: Character-and-scene episodic memory for agents. *arXiv preprint arXiv:2602.06051*.

Xiaoxu Ma, Runhao Li, Hanwen Liu, Xiangbo Zhang, and Zhenyu Weng. 2026b. Unihash: Unifying pointwise and pairwise hashing paradigms for seen and unseen category retrieval. *arXiv preprint arXiv:2601.09828*.

Xiaoxu Ma, Xiangbo Zhang, and Zhenyu Weng. 2026c. Stable and explainable personality trait evaluation in large language models with internal activations. *arXiv preprint arXiv:2601.09833*.

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. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Aaron Meurer, Christopher P Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K Moore, Sartaj Singh, and 1 others. 2017. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103.

Qizhi Pei, Lijun Wu, Zhuoshi Pan, Yu Li, Honglin Lin, Chenlin Ming, Xin Gao, Conghui He, and Rui Yan. 2025. Mathfusion: Enhancing mathematic problem-solving of llm through instruction fusion. *arXiv preprint arXiv:2503.16212*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

Yanwei Ren, Haotian Zhang, Fuxiang Wu, Jiayan Qiu, Jiaxing Huang, Baosheng Yu, and Liu Liu. 2025. Sigma: Refining large language model reasoning via sibling-guided monte carlo augmentation. *arXiv preprint arXiv:2506.06470*.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: Scaling instruction tuning for mathematical reasoning. In *Forty-first International Conference on Machine Learning*.

Yuhang Tian, Dandan Song, Zhijing Wu, Pan Yang, Changzhi Zhou, Jun Yang, Hao Wang, Huipeng Ma, Chenhao Li, and Luan Zhang. 2025a. Compkbqa: Component-wise task decomposition for knowledge base question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 293–309.

Yuhang Tian, Dandan Song, Zhijing Wu, Changzhi Zhou, Hao Wang, Jun Yang, Jing Xu, Ruanmin Cao, and Haoyu Wang. 2024. Augmenting reasoning capabilities of llms with graph structures in knowledge base question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11967–11977.

Yuhang Tian, Pan Yang, Dandan Song, Zhijing Wu, and Hao Wang. 2025b. Grv-kbqa: A three-stage framework for knowledge base question answering with decoupled logical structure, semantic grounding and structure-aware validation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2618–2632.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. DART-math: Difficulty-aware rejection tuning for mathematical problem-solving. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Tom Vodopivec and Branko Šter. 2014. Enhancing upper confidence bounds for trees with temporal difference values. In *2014 IEEE conference on computational intelligence and games*, pages 1–8. IEEE.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439.

Shansong Wang, Mingzhe Hu, Qiang Li, Mojtaba Safari, and Xiaofeng Yang. 2025. Capabilities of gpt-5 on multimodal medical reasoning. *arXiv preprint arXiv:2508.08224*.

Zi-Han Wang, Lam Nguyen, Zhengyang Zhao, Mengyue Yang, Chengwei Qin, Yujiu Yang, and Linyi Yang. 2026. Creativebench: Benchmarking and enhancing machine creativity via self-evolving challenges. *arXiv preprint arXiv:2603.11863*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. 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.

Junhao Xiao, Zhiyu Wu, Hao Lin, Yi Chen, Yahui Liu, Xiaoran Zhao, Zixu Wang, and Zejiang He. 2026. Not just what's there: Enabling clip to comprehend negated visual descriptions without fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 10978–10986.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024a. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

Hongshen Xu, Zihan Wang, Zichen Zhu, Lei Pan, Xingyu Chen, Shuai Fan, Lu Chen, and Kai Yu. 2025. Alignment for efficient tool calling of large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17787–17803.

Hongshen Xu, Zichen Zhu, Lei Pan, Zihan Wang, Su Zhu, Da Ma, Ruisheng Cao, Lu Chen, and Kai Yu. 2024b. Reducing tool hallucination via reliability alignment. *arXiv preprint arXiv:2412.04141*.

Yuchen Yan, Peiyan Zhang, Zheng Fang, and Qingqing Long. 2024. Inductive graph alignment prompt: bridging the gap between graph pre-training and inductive fine-tuning from spectral perspective. In *Proceedings of the ACM Web Conference 2024*, pages 4328–4339.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Limei Yao, Kexin Ma, Ruochun Jin, Haoqi Zheng, and Dong Wang. 2025. Enhancing vector data quality through negative learning for retrieval-augmented large models. In *2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. 2025. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639:609–616.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772.

Peiyan Zhang, Yuchen Yan, Chaozhuo Li, Senzhang Wang, Xing Xie, Guojie Song, and Sunghun Kim. 2023. Continual learning on dynamic graphs via parameter isolation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 601–611.

Peiyan Zhang, Yuchen Yan, Xi Zhang, Chaozhuo Li, Senzhang Wang, Feiran Huang, and Sunghun Kim. 2024b. Transgnn: Harnessing the collaborative power of transformers and graph neural networks for recommender systems. In *Proceedings of the 47th International ACM SIGIR conference on research and development in information retrieval*, pages 1285–1295.

Qianchi Zhang, Hainan Zhang, Liang Pang, Yongxin Tong, Hongwei Zheng, and Zhiming Zheng. 2025a. Less is more: Compact clue selection for efficient retrieval-augmented generation reasoning. *arXiv preprint arXiv:2502.11811*.

Wenyuan Zhang, Tianyun Liu, Mengxiao Song, Xiaodong Li, and Tingwen Liu. 2025b. Sotopia-: Dynamic strategy injection learning and social instruction following evaluation for social agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24669–24697.

Wenyuan Zhang, Shuaiyi Nie, Xinghua Zhang, Zefeng Zhang, and Tingwen Liu. 2025c. S1-bench: A simple benchmark for evaluating system 1 thinking capability of large reasoning models. *arXiv preprint arXiv:2504.10368*.

Yuanjun Zhang, Fuzel Ahamed Shaik, Suvojit Acharjee, Fahad Khalid, and Mourad Oussalah. 2026. Towards reliable multimodal disaster severity assessment through preference optimization and explainable vision-language reasoning. *Reliability Engineering & System Safety*, page 112674.

Wei Zhu, Zhiwen Tang, and Kun Yue. 2026a. Symphony: Synergistic multi-agent planning with heterogeneous language model assembly. *arXiv preprint arXiv:2601.22623*.

Wei Zhu, Lixing Yu, Hao-Ren Yao, Zhiwen Tang, and Kun Yue. 2026b. Task-aware llm council with adaptive decision pathways for decision support. *arXiv preprint arXiv:2601.22662*.

Geigh Zollicoffer, Tanush Chopra, Mingkuan Yan, Xiaoxu Ma, Kenneth Eaton, and Mark Riedl. 2025. World model robustness via surprise recognition. *arXiv preprint arXiv:2512.01119*.

A Implementation Details

A.1 CRPS Data Synthesis Implementation

In alignment with the decoupled explorer-analyst paradigm described in Section 3, our data synthesis pipeline utilizes distinct models for exploration and critique. Specifically, we employ **Qwen2.5-Math-7B-Instruct** as the explorer to generate the candidate trajectories, and **gpt-5-mini** as the analyst for generating critiques and synthesizing the final reasoning paths. This ensures the supervision signals are derived from a highly capable teacher model that possesses advanced meta-cognitive abilities.

Exploratory Trajectory Collection (MCTS). For each problem q_i in the seed dataset, we employ the Upper Confidence Bound for Trees (UCT) (Vodopivec and Šter, 2014) algorithm using the explorer model. We set the exploration constant $c_{\text{puct}} = 1.4$ to encourage diverse reasoning strategies. The maximum tree depth is limited to 16 steps, with a maximum of $k = 3$ actions sampled per node. We perform 10 rollouts per problem. Terminal states are rewarded binary values ($r = 1$ for correct answers, $r = 0$ otherwise) based on strict ground-truth matching.

Trajectory Stratification and Pairing. From the constructed trees, we extract complete trajectories and stratify them into:

- $\mathcal{T}^{\text{high}}$: Paths yielding the correct answer with the highest accumulated process reward (or shortest length among correct paths).
- $\mathcal{T}^{\text{low}}$: Paths yielding incorrect answers ($\tau_{incorrect}$), or correct but inefficient/suboptimal paths ($\tau_{inefficient}$) as described in Section 3.2.

We sample $K = 10$ contrastive pairs (τ^+, τ^-) per problem to serve as inputs for the analyst model. Crucially, consistent with Section 3.2, τ^- is sampled proportionally to its visit count $N(\tau)$ to target the explorer's stubborn error modes, rather than uniformly at random.

Synthesis and Dataset Construction. Following the synthesis of candidate reasoning paths by the analyst (Section 3.4), we apply the verification function $\mathcal{V}(\hat{\tau})$. For mathematical tasks, this involves extracting the final answer and performing an exact-match comparison with the ground truth (using SymPy (Meurer et al., 2017) for equivalence checks). Only synthesized paths that yield the correct answer are retained for the training pool. To construct the final datasets $\mathcal{D}_{\text{syn}}$ at different scales (15K, 30K, 60K) for our scaling laws analysis, we perform uniform sampling from the verified pool. For our primary **CRPS-30K** dataset, this corresponds to retaining approximately 2 synthesized variations per seed problem from the combined GSM8K and MATH training sets.

The specific prompts for critique and synthesis are detailed in Appendix F.

A.2 Baselines

Vanilla MCTS / Rejection Sampling (RFT) (Yuan et al., 2023): We construct the dataset by selecting the single highest-reward trajectory found during the explorer's MCTS for each problem. This represents the standard "selection-based" paradigm.

MMIQC (Liu et al., 2025a): A massive multi-source instruction tuning dataset containing over 2.3M examples. We include this to compare CRPS against massive-scale data scaling.

DART-Math (Tong et al., 2024): A strong baseline that applies difficulty-aware rejection sampling. We re-implement their difficulty scoring mechanism to select the top 590K examples from our search trees. Note that while we utilize the same raw trajectory pool for experimental consistency, DART-Math requires retaining this massive volume to achieve its performance, implying a significantly higher exploration budget compared to the sparse subset required by CRPS.

SIGMA (Ren et al., 2025): A recent feedback-driven refinement method. We implement a fair offline adaptation of SIGMA. Using the same MCTS trajectories as CRPS, we construct training pairs $(q, \tau^-) \rightarrow \tau^+$ where the target model learns to rewrite a suboptimal path into a correct one, matching the dataset size of our CRPS experiments (30K/60K). This isolates the impact of the contrastive synthesis objective versus direct refinement.

MathFusion (Pei et al., 2025): We compare against the sequential training setup of MathFusion, which aggregates diverse reasoning components.

A.3 Models and Training

We conduct experiments on three representative target base models to demonstrate architectural robustness and transferability:

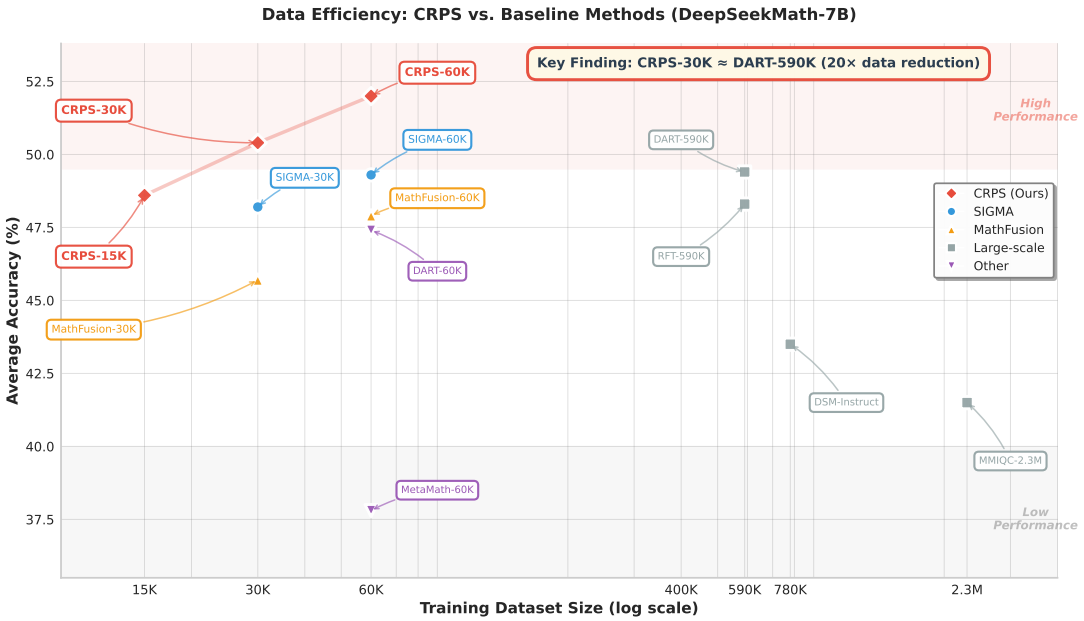

Figure 3: An illustration of the reasoning performance of fully fine-tuned DeepSeekMath-7B models. The models are trained with data sizes ranging from 15K to 2.3M.

- **DeepSeekMath-7B-Base** (Shao et al., 2024): Specialized in mathematical reasoning.
- **LLaMA3-8B** (Grattafiori et al., 2024): A strong general-purpose foundation model.
- **Mistral-7B-v0.1** (Jiang et al., 2023): Widely used for its sliding window attention efficiency.

All target models are full fine-tuned using the standard causal language modeling objective. The training data consists of the problem statement q as the prompt and the synthesized reasoning path $\hat{\tau}$ as the completion. We use the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.95$. The learning rate is warmed up for 3% of the total steps to a peak of 2×10^{-5} and then decays to zero via a cosine schedule. We use a global batch size of 128. Training is performed on 8 $\times$ NVIDIA A800 (80GB) GPUs using DeepSpeed ZeRO-3 (Rasley et al., 2020) offloading. Models are trained for 3 epochs.

A.4 Evaluation Protocol

To ensure reproducibility, we employ a standardized evaluation protocol across all experiments.

- **Inference:** We use zero-shot Chain-of-Thought (CoT) prompting. To eliminate randomness and measure the model's robust capability, we use greedy decoding (temperature $= 0$) for all main results.
- **Answer Extraction:** We extract the final answer from the model's output (typically boxed or last numerical value) and perform exact match comparison against the ground truth. For mathematical equivalence (e.g., fractions vs. decimals), we utilize the SymPy library for symbolic verification.

B Qualitative Analysis

To understand the mechanisms driving the quantitative gains, we perform a qualitative dissection of the synthesized reasoning paths.

B.1 Mechanisms of Contrastive Correction

We analyze representative cases (detailed in Appendix G) to understand how contrastive signals drive reasoning improvements. The synthesis process demonstrates a consistent shift from fragile heuristics to structural rigor, driven by the explicit verbalization of failure modes.

- **From Ad-hoc Formulas to First Principles (Case 1, Combinatorics):** In the triangular peg board problem, the baseline MCTS fell into a "Grid Assumption" trap, treating the triangular board as a rectangular grid and computing factorials for each color. The contrastive critique identified this as a geometric mismatch: positions only exist where $r + c \leq 6$. Consequently, the synthesized path explicitly verbalized the strategic pivot:

Model	#Samples	In-Domain		Out-of-Domain				Avg.
		MATH	GSM8K	College	DM	Olympiad	Theorem	
LLaMA3-8B (General-Purpose Base Model)								
LLaMA3-MetaMath	400K	32.5	77.3	20.6	35.0	5.5	13.8	30.8
LLaMA3-RFT	590K	39.7	81.7	23.9	41.7	9.3	14.9	35.2
LLaMA3-DART	590K	46.6	81.1	28.8	48.0	14.5	19.4	39.7
LLaMA3-MMIQC	2.3M	39.5	77.6	29.5	41.0	9.6	16.2	35.6
LLaMA3-CRPS-15K	**15K**	37.8*	83.2*	25.8*	44.5*	11.9*	23.5*	37.8*
MathFusion (Sequential)	30K	38.8	77.9	25.1	42.0	12.6	17.0	35.6
SIGMA-30K	30K	40.8	79.5	26.3	47.5	12.7	19.1	37.7
LLaMA3-CRPS-30K	**30K**	**42.1**$^{*+1.3}$	**81.8**$^{*+2.3}$	**27.9**$^{*+1.6}$	**49.2**$^{*+1.7}$	**14.8**$^{*+2.1}$	**21.6**$^{*+2.5}$	**39.6**$^{*+1.9}$
LLaMA3-DART	60K	39.6	82.2	27.9	39.9	12.9	22.9	37.6
MathFusion	60K	46.5	79.2	27.9	43.4	17.2	20.0	39.0
SIGMA-60K	60K	44.9	82.4	28.1	49.2	15.3	21.3	40.2
LLaMA3-CRPS-60K	**60K**	**46.8**$^{*+1.9}$	**83.5**$^{*+1.1}$	**29.4**$^{*+1.3}$	**51.3**$^{*+2.1}$	**17.9**$^{*+2.6}$	**23.8**$^{*+2.5}$	**41.9**$^{*+1.7}$
Mistral-7B-v0.1 (General-Purpose Base Model)								
Mistral-MetaMath	400K	29.8	76.5	19.3	28.0	5.9	14.0	28.9
Mistral-WizardMath	418K	32.3	80.4	23.1	38.4	7.7	16.6	33.1
Mistral-RFT	590K	38.7	82.3	24.2	35.6	8.7	16.2	34.3
Mistral-DART	590K	45.5	81.1	29.4	45.1	14.7	17.0	38.8
Mistral-CRPS-15K	**15K**	31.8*	76.9*	22.4*	41.2*	9.1*	17.8*	33.2*
MathFusion (Sequential)	30K	32.7	73.9	18.9	29.3	9.3	15.5	29.9
SIGMA-30K	30K	35.5	78.6	22.1	43.8	11.1	18.0	34.9
Mistral-CRPS-30K	**30K**	**37.2**$^{*+2.3}$	**80.1**$^{*+1.5}$	**24.3**$^{*+2.2}$	**46.8**$^{*+3.0}$	**12.5**$^{*+1.4}$	**19.7**$^{*+1.7}$	**36.8**$^{*+1.9}$
Mistral-MetaMath	60K	22.7	70.8	14.1	27.2	5.0	12.2	25.3
Mistral-DART	60K	34.1	77.2	23.4	36.0	8.7	18.2	32.9
MathFusion	60K	41.6	79.8	24.3	39.2	13.6	18.1	36.1
SIGMA-60K	60K	40.3	79.2	24.1	46.1	12.3	19.2	36.9
Mistral-CRPS-60K	**60K**	**42.7**$^{*+2.4}$	**81.4**$^{*+2.2}$	**26.1**$^{*+2.0}$	**48.9**$^{*+2.8}$	**14.2**$^{*+1.9}$	**21.0**$^{*+1.8}$	**39.1**$^{*+2.2}$

Table 5: Performance comparison across base models, training methods, and dataset scales. Arrows indicate accuracy changes relative to the strongest baseline (highlighted in blue). Best results in each data scale are in **bold**. * indicate statistical significance at $p < 0.05$ compared to the best baseline (calculated via paired t-test).

"It is tempting to multiply factorials as if the board were rectangular, but on this triangular board, the constraints at each step leave exactly one valid column." This demonstrates the model's ability to inhibit high-probability heuristic tokens in favor of constraint-aware enumeration.

- **From Brute-Force Calculus to Structural Insight (Case 2, Algebra):** We observe that CRPS also optimizes reasoning efficiency (Soft Negatives). In the quadratic minimization problem, while the baseline correctly attempted the task by setting partial derivatives to zero and solving a parametric linear system, the contrastive analysis flagged this as computationally heavy and prone to stalling. The synthesized CRPS solution bypassed this approach by completing the square directly, thereby revealing the minimum-value condition structurally and reducing algebraic complexity.

- **From Unjustified Shortcuts to Rigorous Integration (Case 3, Calculus):** For the cylindrical wedge volume problem, the baseline assumed the wedge was simply half a cylinder—arriving at the correct answer through fundamentally flawed reasoning. The contrastive analyst identified this as an unjustified geometric simplification. The resulting synthesized path set up coordinates at the tangent point and performed proper integration, explicitly stating: *"A common mistake is to assume the wedge is simply half a cylinder—but each slice has a different height, so integration is necessary."*

These transformations confirm that CRPS does not merely filter for correctness; it utilizes the distribution of suboptimal paths to condition generation against specific logical leaps and heuristic traps,

Method	HumanEval	StrategyQA
Code Generation (Base: Qwen2.5-Coder-7B)		
Supervised FT	48.2	–
Best-of-32 Sampling	52.4	–
Self-Refinement	54.1	–
CRPS (Ours)	**57.9**	–
Commonsense (Base: LLaMA3-8B)		
Supervised FT	–	68.3
Best-of-32 Sampling	–	71.5
Self-Refinement	–	73.2
CRPS (Ours)	–	**75.8**

Table 6: Performance on non-mathematical reasoning tasks. We report pass@1 accuracy for HumanEval and exact-match accuracy for StrategyQA. CRPS consistently improves performance over standard refinement methods by leveraging the contrast between valid and invalid execution traces.

resulting in reasoning chains that are both correct and explicitly justified.

C Generalization to Non-Mathematical Reasoning

To validate the universality of the CRPS framework, we extend our experiments to code generation (**HumanEval** (Chen et al., 2021)) and multi-hop commonsense reasoning (**StrategyQA** (Geva et al., 2021)). We adapt the MCTS reward mechanism $r(\tau)$ to domain-specific verifiers: execution results against unit tests for code, and exact answer matching for commonsense queries. We adapt the decoupled architecture to these domains by employing **Qwen2.5-Coder-7B** (Hui et al., 2024) and **LLaMA3-8B** as the domain-specific explorers (and subsequent target base models), while retaining **gpt-5-mini** as the analyst for contrastive critique and synthesis. We compare CRPS against Supervised Fine-Tuning (SFT), Best-of-32 Sampling, and iterative Self-Refinement (Madaan et al., 2023).

As shown in Table 6, CRPS consistently outperforms baselines across both domains. On HumanEval, CRPS achieves 57.9% pass@1, surpassing Self-Refinement by 3.8 points. Qualitative analysis suggests that the contrastive mechanism effectively targets latent failure modes: in code generation, the explorer's $\mathcal{T}^{\text{low}}$ trajectories often expose edge-case bugs, while in StrategyQA, they reveal "reasoning shortcuts" where correct answers are derived from hallucinated logic. By enabling the analyst to synthesize paths conditioned on explicitly avoiding these structural pitfalls, CRPS demonstrates that the principle of learning from distilled contrastive search trajectories transfers effectively beyond mathematical formalism.

Method	Target Acc.	Dataset Size	Training Cost (h)
Baselines			
DART-Math	49.4%	590K	192
RFT	48.3%	590K	192
SIGMA	48.2%	30K	10
Ours			
CRPS	**50.4%**	**30K**	**10**

Table 7: Computational cost breakdown to achieve ≈50% accuracy on MATH+GSM8K (DeepSeekMath-7B). CRPS shifts the compute burden from massive exploration and training to targeted synthesis, resulting in a **19.2× reduction** in training GPU hours compared to standard search-based selection.

D Step Segmentation Protocol

As discussed in Section 3, defining the atomic unit of reasoning is a fundamental challenge in CoT research. We define a "step" as a *Semantic Reasoning Act*—a discrete move that transforms the state of the problem. To ensure consistent branching for MCTS and semantic alignment for CRPS, we employ a hierarchical segmentation strategy, prioritizing structural delimiters over linguistic ones (Table 8). We strictly avoid splitting within mathematical expressions (e.g., inside LaTeX $\$ \ldots \$$) to preserve equation integrity, and enforce a maximum length of 256 tokens per step to prevent "run-on" steps that dilute the credit assignment signal.

E Semantic Stability Evaluation Setup

To rigorously evaluate the semantic stability of the fine-tuned target models (Section 5.3), we constructed a perturbed dataset based on the GSM8K test set.

E.1 Data Construction

We randomly sampled 200 problems from the GSM8K test set as seed queries. For each seed problem q, we utilized GPT-5 to generate a perturbed variant q' using the following prompt template. The generation was constrained to introduce two types of semantic noise:

1. **Linguistic Paraphrasing:** Altering sentence structures and replacing entities (e.g., names, objects) while preserving the numerical relationships.

Priority	Delimiter Type	Specific Tokens / Patterns	Rationale
1	**Structure** (Primary)	`\n`, `\n\n`, `\\` (LaTeX newline)	Natural boundaries in math derivations; ensures semantic completeness.
2	**Logic Connectors**	`Therefore,`, `Thus,`, `Hence,`, `So,`, `Consequently,`	Signals a deductive leap or conclusion; critical for value estimation ($Q(s)$).
3	**Explicit Enumeration**	`1.`, `2.`, `Step 1:`, `First,`	Explicitly structured reasoning steps in CoT.

Table 8: Heuristic rules for step segmentation in MCTS and CRPS semantic alignment.

2. **Distractor Injection:** Inserting irrelevant context sentences that do not affect the calculation but serve as "attention traps".

Perturbation Generation Prompt

Instruction: Rewrite the following math word problem to create a "Semantically Equivalent but Perturbed" version. 1. Change the names of people and objects. 2. Rephrase the sentence structures significantly. 3. Insert one sentence of "distractor" information that contains a number but is irrelevant to the solution. 4. Do NOT change the underlying logic or the required calculation flow. The final answer must remain exactly the same.
Input Problem: [Insert Original Problem]
Output Problem:

E.2 Metric Definition

We employ **Strict Consistency** as our primary metric. Let $M(q)$ denote the binary correctness (1 for correct, 0 for incorrect) of the target model on query q. For a dataset of size N, the consistency score S is calculated as:

$$S = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[M(q_i) = 1 \wedge M(q_i') = 1] \quad (4)$$

This metric is stricter than simple answer agreement (where two wrong answers might count as consistent) or average accuracy. It specifically measures the target model's ability to maintain correct reasoning despite adversarial surface-level perturbations, verifying whether it has truly internalized the robust structural logic distilled during the CRPS synthesis phase.

F Prompts and Instructions

To ensure reproducibility, we provide the exact prompt templates used in our CRPS framework. The process strictly follows the dual-granularity analysis described in Section 3.3, separating global strategic insights from local step-wise critiques performed by the analyst model.

F.1 Phase 1: Dual-Granularity Contrastive Analysis

In this phase, the analyst model (**gpt-5-mini**) processes the outputs from the explorer. It receives the problem q, a high-quality trajectory τ^+ (positive), and a low-quality trajectory τ^- (negative) generated during MCTS. The instructions explicitly require the analyst to perform **Semantic Alignment** to locate the divergence step and generate both global and local critiques.

System Prompt: The Contrastive Analyst

You are an expert Mathematical Reasoning Analyst. Your task is to perform a dual-granularity contrastive analysis between two reasoning trajectories for the same mathematical problem. One trajectory arrives at the correct answer (Trajectory A) and the other arrives at an incorrect answer (Trajectory B). You must identify where and why the trajectories diverge, analyze both local step-level logic and global strategic differences, and synthesize actionable guidance.

User Prompt: Dual-Granularity Analysis Instruction

Problem: [INSERT PROBLEM q]
Trajectory A (Correct): [INSERT TRAJECTORY τ^+]
Trajectory B (Incorrect): [INSERT TRAJECTORY τ^-]
Perform a dual-granularity contrastive analysis of the two trajectories above. Provide your analysis as a JSON object with the fol-

lowing structure:

```
{
  "divergence_step_index": <int>,
  "local_step_critique": {
      "trajectory_a_logic": "<description
          of Trajectory A's
          reasoning at the divergence
              point>",
      "trajectory_b_logic": "<description
          of Trajectory B's
          reasoning at the divergence
              point>",
      "critique_of_difference": "<precise
          explanation of why
          Trajectory B's step is
              incorrect and how Trajectory
          A's step is correct>"
  } ,
  "global_strategic_analysis":
       "<high-level comparison of
      the overall strategies employed by
          each trajectory>",
  "synthesized_guidance": {
      "success_pattern": "<the key
          reasoning pattern from
          Trajectory A that led to the
              correct answer>",
      "failure_mode_to_avoid": "<the
          specific reasoning
          pitfall from Trajectory B to be
              avoided>"
  }
}
```

Return ONLY the JSON object, with no additional text.

F.2 Phase 2: Pattern-Informed Path Synthesis

In this phase, the analyst model synthesizes a new reasoning path $\hat{\tau}$. The generation is conditioned on the extracted critiques, acting as a prompt-based regularizer to explicitly navigate around the explorer's identified failure mode.

System Prompt: The Pattern-Informed Solver

You are an advanced Mathematical Reasoning Engine. Your task is to solve a mathematical problem by generating a step-by-step solution that is informed by prior contrastive analysis of correct and incorrect reasoning paths.
KEY REQUIREMENTS:
1. **INCORPORATE** contrastive insights naturally into your reasoning. At critical steps, briefly explain WHY you chose the correct approach and what common mistake to avoid. For example: "A common mistake is to use $C(6, 2)$ here, which treats identical objects as distinct. Instead, we enumerate cases." This is the core value of CRPS.
2. Do **NOT** use meta-language like "the critique suggests", "following the success pattern", or "as identified in the analysis". Write as if **YOU** discovered these insights while solving.
3. Match the target model's formatting style (bold headers, LaTeX, numbered steps).

User Prompt: Synthesis Instruction

Problem: [INSERT PROBLEM q]
Contrastive Insights (weave naturally into solution, do NOT reference directly):

- Why the correct approach works: [INSERT `success_pattern`]
- Common mistake to avoid: [INSERT `failure_mode_to_avoid`]
- Key difference at Step [INSERT `divergence_step_index`]: [INSERT `critique_of_difference`]
- Strategic overview: [INSERT `global_strategic_analysis`]

Target Output Style: [INSERT style example from a successful trajectory]
Solve step by step. At the critical decision point, naturally explain why you chose your approach and what pitfall you are avoiding , as if you discovered this yourself. Do NOT say "the critique" or "the analysis".
Format: Step 1: ... Step 2: ... ... Final Answer: \boxed{...}

F.3 Output Parsing and Robustness

To ensure the autonomy of our pipeline regardless of the chosen analyst, we implement a lightweight post-processing mechanism. We utilize regular expressions to extract the JSON block from the model's raw output and apply a rule-based parser to correct common formatting issues (e.g., unescaped quotes or missing trailing braces). Instances that remain unparsable after this repair process are discarded. Empirically, we observe a parsing failure rate of less than 5% across all evaluated models, confirming that the instruction-tuned models pos-

sess sufficient structural adherence to support our automated synthesis loop without human intervention.

G Examples

Here we demonstrate the CRPS pipeline on three representative cases. To showcase the framework's versatility, we include both **Correctness Contrasts** (learning from errors, Cases 1 & 3) and an **Efficiency Contrast** (learning from suboptimal strategies, Case 2). The input trajectories (τ^+ and τ^-) are sampled from the exploration trees generated by the explorer model (**Qwen2.5-Math-7B-Instruct**). The synthesized outputs reflect the analyst model's (**gpt-5-mini**) actual generation capabilities, showing how it incorporates its own contrastive critiques into procedural steps without excessive metacognitive language. These distilled, high-quality paths are subsequently used to train the target base models.

Case 1 illustrates a *wrong mental model*: the explorer confidently applies a rectangular-grid counting strategy to a triangular board, and the contrastive analysis pinpoints the geometric mismatch at the very first step. Case 2 demonstrates an *efficiency gap*: both trajectories reach the correct answer, but the positive path uses an elegant structural approach (completing the square) while the negative path stalls in heavy parametric algebra. Case 3 exposes a *geometric reasoning error*: the explorer incorrectly assumes a wedge is half a cylinder, and the critique identifies the need for integration over cross-sectional slices of varying height.

Case 1. Hard Negative: Wrong Mental Model (Combinatorics)

Question: There are 5 yellow pegs, 4 red pegs, 3 green pegs, 2 blue pegs, and 1 orange peg to be placed on a triangular peg board. In how many ways can the pegs be placed so that no (horizontal) row or (vertical) column contains two or more pegs of the same color?

Input Trajectories:

- τ^+ (Positive): Maps the triangular board structure, identifies that position (r, c) exists iff $r + c \leq 6$, and shows each color placement is uniquely forced.
- τ^- (Hard Negative): Treats the triangular board as a rectangular grid and computes $5! \times 4! \times 3! \times 2! \times 1!$, vastly overcounting.

Phase 1: Dual-Granularity Critique (JSON)


```
{
  "divergence_step_index": 1,
  "local_step_critique": {
      "trajectory_a_logic": "Maps out
          exact positions in each row
          and column of the triangular
          board before counting.",
      "trajectory_b_logic": "Treats the
          board as a full rectangular
          grid, computing factorials for
          each color.",
      "critique_of_difference": "The
          triangular board has position
          (r,c) only if r+c<=6.
          Trajectory B assumes all n!
          permutations are valid, but
          most positions do not exist on
          the triangular board."
  } ,
  "global_strategic_analysis": "Both
       attempt sequential color-by-color
       placement. Trajectory B ignores
       the geometric constraints, while
       Trajectory A recognizes that these
       constraints uniquely force each
       placement.",
  "synthesized_guidance": {
      "success_pattern": "Map out exact
          positions available in each
          row and column before
          counting. Recognize when
          constraints are tight enough
          to force a unique
          arrangement.",
      "failure_mode_to_avoid": "Treating
          a triangular board as a
          rectangular grid when
          computing permutations."
  }
}
```


Phase 2: Pattern-Informed Synthesis

Step 1: Understand the board structure. The triangular board has position (r, c) iff $r + c \leq 6$. Row r has $6 - r$ positions; column c has $6 - c$ positions. Total: 15 positions for 15 pegs.

Step 2: Place yellow pegs (5 pegs). Each of 5 rows and 5 columns needs exactly one yellow peg. We need a bijection $\sigma : \{1, \ldots, 5\} \to \{1, \ldots, 5\}$ with $\sigma(r) \leq 6 - r$. Working upward: $\sigma(5) = 1$ (forced), $\sigma(4) = 2$ (forced), $\sigma(3) = 3$, $\sigma(2) = 4$, $\sigma(1) = 5$. Exactly 1 way.

Step 3: Place remaining colors. By the same forced-assignment argument applied to red (τ), green (γ), blue (β), and orange: each is uniquely determined. It is tempting to multiply factorials as if the board were rectangular, but on this triangular board, the constraints at each step leave exactly one valid column.
Step 4: Count. Total arrangements = $1 \times 1 \times 1 \times 1 \times 1 = 1$.
Final Answer: $\boxed{1}$

Case 2. Soft Negative: Efficiency Gap (Algebra)

Question: There exists a constant k so that the minimum value of $4x^2 - 6kxy + (3k^2 + 2)y^2 - 4x - 4y + 6$ over all real numbers x and y is 0. Find k.

Input Trajectories:

- τ^+ (Positive): Completes the square to express the function as a sum of squared terms plus a constant, directly revealing conditions on k.
- τ^- (Soft Negative): Sets partial derivatives to zero, attempts to solve a parametric 2×2 linear system, but stalls during intermediate algebra.

Phase 1: Dual-Granularity Critique (JSON)


```
{
  "divergence_step_index": 1,
  "local_step_critique": {
      "trajectory_a_logic": "Analyzes the
          quadratic form structure and
          completes the square.",
      "trajectory_b_logic": "Computes
          partial derivatives and sets
          up a parametric linear
          system.",
      "critique_of_difference":
          "Trajectory B's calculus
          approach requires inverting a
          parametric matrix and
          extensive back-substitution.
          Trajectory A's structural
          approach directly reveals when
          the minimum equals zero."
  } ,
  "global_strategic_analysis":
      "Trajectory A adopts a structural
      approach via completing the
      square, naturally connecting
      coefficients to the minimum-value
      condition. Trajectory B uses
      brute-force calculus that stalls
      during algebra.",
  "synthesized_guidance": {
      "success_pattern": "For
          multivariable quadratic
          minimization with a parameter,
          complete the square to express
          as sum-of-squares plus
          constant.",
      "failure_mode_to_avoid": "Diving
          into heavy parametric linear
          systems without considering
          structural alternatives."
  }
}
```


Phase 2: Pattern-Informed Synthesis
Step 1: Choose the method. Rather than setting partial derivatives to zero and solving a parametric system (which is computationally heavy and prone to stalling), we complete the square directly.
Step 2: Complete the square in x.

$$f(x,y) = 4\left(x - \tfrac{6ky+4}{8}\right)^2 + \tfrac{3k^2+8}{4}\,y^2 - (3k+4)y + 5.$$

Step 3: Complete the square in y. Since $\frac{3k^2+8}{4} > 0$ for all real k, the form is positive definite and a minimum exists. The minimum value is:

$$f_{\min} = 5 - \frac{(3k+4)^2}{3k^2+8}.$$

Step 4: Solve for k. Setting $f_{\min} = 0$: $5(3k^2+8) = (3k+4)^2 \implies 6k^2 - 24k + 24 = 0 \implies (k-2)^2 = 0$.
Final Answer: $\boxed{2}$

Case 3. Hard Negative: Geometric Reasoning Error (Calculus)

Question: A cylindrical log has diameter 12 inches. A wedge is cut from the log by making two planar cuts that go entirely through the log. The first is perpendicular to the axis of the cylinder, and the plane of the second cut forms a 45° angle with the plane of the first cut. The intersection of these two planes has exactly one point in common with the log. The number of cubic

inches in the wedge can be expressed as $n\pi$. Find n.

Input Trajectories:

- τ^+ (Positive): Sets up coordinates at the tangent point, expresses height as $h(x) = x$, computes chord widths, and integrates.
- τ^- (Hard Negative): Assumes the wedge is half a cylinder, arriving at the correct numerical answer through flawed reasoning.

Phase 1: Dual-Granularity Critique (JSON)


```
{
  "divergence_step_index": 1,
  "local_step_critique": {
     "trajectory_a_logic": "Sets up a
         coordinate system at the
         tangent point and computes
         cross-sectional slices.",
     "trajectory_b_logic": "Assumes the
         wedge is half a cylinder
         without justification.",
     "critique_of_difference": "The
         45-degree angled cut tangent
         to one edge creates a
         non-trivial shape where each
         slice has a different height.
         Trajectory B's simplification
         is unjustified."
  } ,
  "global_strategic_analysis":
      "Trajectory A uses geometric
      decomposition and integration.
      Trajectory B takes an unjustified
      shortcut that accidentally gives
      the right answer but reflects a
      fundamental misunderstanding.",
  "synthesized_guidance": {
     "success_pattern": "Set up
         coordinates at the tangent
         point, express height as
         h(x)=x, compute chord widths
         from the circle equation, and
         integrate.",
     "failure_mode_to_avoid": "Assuming
         the wedge is a simple fraction
         of a cylinder without rigorous
         justification."
  }
}
```


Phase 2: Pattern-Informed Synthesis

Step 1: Set up coordinates. Let the tangent point be the origin, with x running along a diameter from 0 to 12 (radius $r = 6$, center at $x = 6$). The height at position x is $h(x) = x\tan(45^\circ) = x$. The chord width is $w(x) = 2\sqrt{36 - (x-6)^2} = 2\sqrt{12x - x^2}$.

Step 2: Set up the integral. A common mistake is to assume the wedge is simply half a cylinder—but each slice has a different height, so integration is necessary:

$$V = 2\int_0^{12} x\sqrt{12x - x^2}\,dx.$$

Step 3: Evaluate. Substituting $x = 6 + 6\sin\theta$, $dx = 6\cos\theta\,d\theta$, $\sqrt{12x - x^2} = 6\cos\theta$:

$$V = 432\int_{-\pi/2}^{\pi/2}(1+\sin\theta)\cos^2\theta\,d\theta =$$

$$432\left[\frac{\pi}{2} + 0\right] = 216\pi,$$

where the $\sin\theta\cos^2\theta$ term vanishes by odd symmetry. Thus $n = 216$.

Final Answer: $\boxed{216}$