

NeuReasoner: Towards Explainable, Controllable, and Unified Reasoning via Mixture-of-Neurons

Haonan Dong^{1*} Kehan Jiang^{2*} Haoran Ye¹
Wenhao Zhu¹ Zhaolu Kang² Guojie Song^{1†}

¹State Key Laboratory of General Artificial Intelligence,
School of Intelligence Science and Technology, Peking University

²School of Software and Microelectronics, Peking University

*Equal contribution †Corresponding author

✉ hndong25@stu.pku.edu.cn, gjsong@pku.edu.cn

Abstract

Large Reasoning Models (LRMs) have recently achieved remarkable success in complex reasoning tasks. However, closer scrutiny reveals persistent failure modes compromising *performance* and *cost*: **I) Intra-step level**, marked by calculation or derivation errors; **II) Inter-step level**, involving oscillation and stagnation; and **III) Instance level**, causing maladaptive over-thinking. Existing endeavors target isolated levels without *unification*, while their black-box nature and reliance on RL hinder *explainability* and *controllability*. To bridge these gaps, we conduct an in-depth white-box analysis, identifying key neurons (*Mixture of Neurons*, MoN) and their fluctuation patterns associated with distinct failures. Building upon these insights, we propose *NeuReasoner*, an *explainable*, *controllable*, and *unified* reasoning framework driven by MoN. Technically, *NeuReasoner* integrates lightweight MLPs for failure detection with a special token-triggered self-correction mechanism learned via SFT. During inference, special tokens are inserted upon failure detection to actuate controllable remedial behaviors. Extensive evaluations across six benchmarks, six backbone models (8B~70B) against nine competitive baselines, demonstrate that *NeuReasoner* achieves performance gains of up to 27.0% while reducing token consumption by 19.6% ~ 63.3%.

1 Introduction

Reasoning capability stands as a fundamental cornerstone of human intelligence and serves as an essential pathway toward AGI (Johnson-Laird, 2010). Recently, facilitated by Chain-of-Thought (CoT) techniques (Wei et al., 2022), Large Language Models (LLMs) demonstrate remarkable potential in complex tasks, such as mathematics (Suzgun et al., 2023), coding (Chen et al., 2021), and science (Lewkowycz et al., 2022), by employing step-by-step reasoning processes. With the advent of Large Reasoning Models (LRMs) such

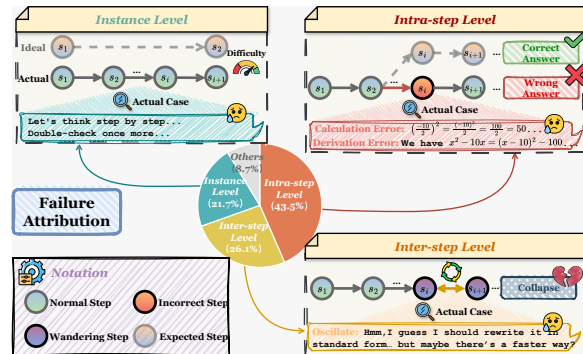


Figure 1: Distribution and illustration of failure modes across different levels using DeepSeek-R1-Distill-Qwen-7B on MATH (Hendrycks et al., 2021).

as OpenAI-o1 (OpenAI et al., 2024), DeepSeek-R1 (Guo et al., 2025), and Gemini-2.5 pro (Team, 2025), Reinforcement Learning (RL)-driven training paradigms further expand the boundaries of reasoning (Yue et al., 2025; Wang et al., 2025b; Yu et al., 2025). This paradigm shift elicit the spontaneous emergence of human-like cognitive patterns within models, including self-verification, reflection, and multi-path exploration (Guo et al., 2025).

Failure Attribution. Despite the remarkable success of LRMs, a close look from the dual perspective of *performance* and *cost* reveals distinct failure modes concealed within the reasoning process, which we systematically identify (Figure 1). Specifically, **I) Intra-step level**: flaws in causal deduction or deviation into incorrect branches during critical steps compromise the final answer (Yao et al., 2025); **II) Inter-step level**: models risk entrapment in reasoning stagnation, where they oscillate between similar trajectories without substantive progress, thereby consuming excessive tokens before eventual collapse (Shojaee et al., 2025); and **III) Instance level**: the failure to calibrate query difficulty induces over-thinking, resulting in a substantial waste of token costs (Chen et al., 2024a).

Research Dilemma. Prior studies attempt to address these failure modes *individually*, where **I)**

Process Reward Models (PRMs) such as Math-Shepherd (Wang et al., 2024), AutoPRM (Chen et al., 2024c), and OpenPRM (Zhang et al., 2025e) are employed for fine-grained step-level supervision; **II**) structured frameworks like ToT (Yao et al., 2023), GoT (Besta et al., 2024), and PGTS (Li, 2025) extend the breadth and depth of exploration; and **III**) approaches including O1-Pruner (Luo et al., 2025), AdaptThink (Zhang et al., 2025b), and ARM (Wu et al., 2025a) adaptively adjust reasoning length. However, significant research gaps persist across three critical axes: **❶ Explainability**, due to the black-box nature of models, in-depth analysis regarding the internal mechanisms underlying failure modes remains absent; **❷ Controllability**, RL-based paradigms encourage the spontaneous emergence of behaviors, rendering the reasoning process unpredictable and difficult to control; and **❸ Unification**, existing methods lack a unified solution targeting these failure modes collectively.

Present Framework. To bridge these gaps, inspired by cognitive science findings that distinct brain regions govern specific functions (Kanwisher, 2010), we hypothesize that three key neuron clusters within LRMs contribute significantly to the three levels of failure modes. Through in-depth analysis of neuron activation dynamics during reasoning, we identify these clusters, termed **Mixture of Neurons (MoN)**, where each cluster functions as an *expert*. Furthermore, we scrutinize the fluctuation patterns of each expert under its corresponding failure mode. Building upon this systematic white-box analysis paradigm, we propose **NeuReasoner**, an *explainable*, *controllable* and *unified* reasoning framework driven by MoN. Technically, **NeuReasoner** ♥ trains lightweight MLPs to monitor and predict the fluctuation patterns of each expert; ♥ through dataset reconstruction, leverages SFT to condition the model to utilize special tokens as triggers that elicit specific behavioral patterns, facilitating controllable self-correction; and ♥ leverages parallel MLPs for online monitoring during inference, inserting special tokens upon identifying failure modes to trigger corresponding behaviors, effectively mitigating failures. Our contributions are summarized as follows:

* **Insightful Analysis.** We systematically summarize the failure modes of LRMs, and inspired by cognitive science, conduct an in-depth white-box analysis at a fine-grained neuron level. This leads

to the identification of MoN that contribute to failure modes, alongside their fluctuation patterns.

- * **Practical Solution.** We propose **NeuReasoner**, an explainable and controllable unified reasoning framework driven by MoN. It integrates lightweight MLPs for failure detection with a special token-triggered self-correction mechanism learned via SFT. During inference, special tokens are dynamically inserted upon failure detection to actuate controllable remedial behaviors.
- * **Experimental Validation.** Extensive experiments on six complex benchmarks and six backbone models against nine baselines demonstrate that **NeuReasoner** achieves performance gains of up to 27.0% while simultaneously reducing token consumption by 19.6% ~ 63.3%. Furthermore, case studies vividly illustrate reasoning details, verifying the explainability and controllability.

2 Dive into Neurons

Inspired by insights from prior research indicating that intermediate layers within LRMs encode the richest representations (Galichin et al., 2025; Skean et al., 2025; Jin et al., 2025), we focus our analysis on the FFN and attention heads in the middle layer. Specifically, we perform **(i) MoN Identification**, leveraging the advanced attribution algorithm of LLMs to identify neuron clusters that contribute most significantly to failure modes of different levels (► Section 2.1); and **(ii) Fluctuation Analysis**, further uncovering the specific fluctuation patterns of MoN associated with the emergence of failure modes (► Section 2.2).

2.1 MoN Identification

To conduct a systematic white-box investigation identifying which specific neuron clusters within the middle layer contribute most significantly to the failure modes of different levels, we employ DePass (Hong et al., 2025), an advanced attribution algorithm of LLM. Given the target layer $l^* = L/2$, let \mathcal{C} denote the set of all neurons (FFN and attention heads) within layer l^* . For a given input sequence, DePass enables the calculation of the independent contribution of each neuron $c \in \mathcal{C}$ toward the token generation at time step t . Let $\phi(c, t) \in \mathbb{R}$ denote the attribution score of c regarding the output logit of the target token y_t :

$$\phi(c, t) = \mathbf{w}_{y_t}^\top \mathbf{h}_{dec}^{(l^*)}(t, c), \quad (1)$$

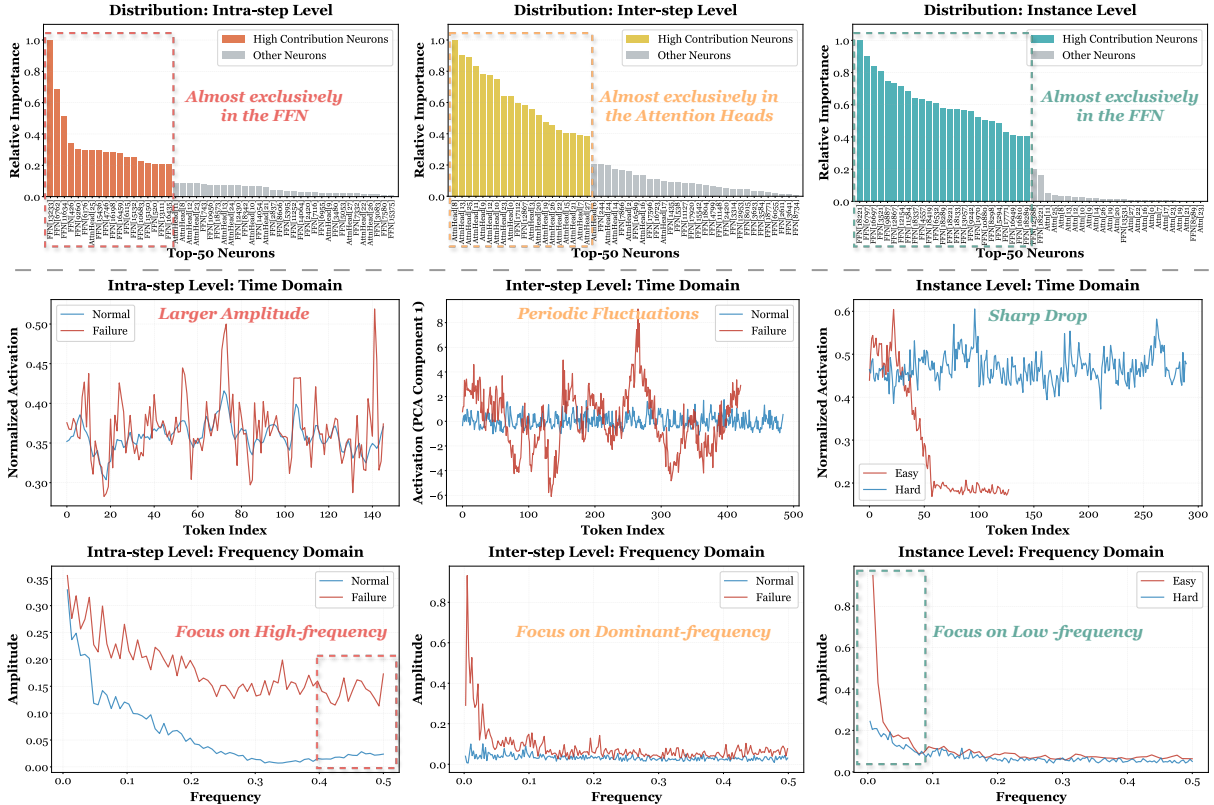


Figure 2: (*Upper*) Distribution of key neurons across distinct failure modes using DeepSeek-R1-Distill-Qwen-7B on MATH. (*Lower*) Time- and frequency-domain analysis of MoN for positive and negative sample pairs.

where \mathbf{w}_{y_t} represents the LM head vector for y_t , and $\mathbf{h}_{dec}^{(l^*)}(t, c)$ corresponds to the decomposed hidden state associated with c .

Mixture of Neurons. We define the set of time steps T as follows: **I) Intra-step level**, the sequence of tokens within the specific erroneous step; **II) Inter-step level**, the sequence of tokens corresponding to the initial steps of multiple attempts; **III) Instance level**, the first K steps at the beginning of the reasoning process. To isolate neurons exerting a persistent and dominant influence across all time steps, we compute the intersection of the most significant neurons $\mathcal{N}^* = \bigcap_{t \in T} \text{TopK}_{c \in \mathcal{C}}(\phi(c, t))$, where $\text{TopK}_{c \in \mathcal{C}}(\cdot)$ retrieves the set of the top- k neurons exhibiting the highest attribution scores. The \mathcal{N}^* across all levels collectively constitute what we term **Mixture of Neurons (MoN)**, with each serving as a distinct *expert*, as illustrated in Figure 2 (*Upper*).

Obs. 1 **Distinct neuron experts contribute significantly to failure modes of different levels.** Specifically, experts associated with intra-step and instance-level failures predominantly cluster within the FFN, whereas those corresponding to inter-step failures are concentrated within the attention heads.

2.2 Fluctuation Analysis

Following MoN identification, we further aim to analyze the underlying fluctuation patterns associated with the emergence of failure modes. To this end, we construct positive and negative sample pairs. Technically, targeting intra-step and inter-step failures, we pause generation at each step, clone the KV cache, and employ LLM-as-Judge to detect failure modes (Zheng et al., 2023). Upon detection, we perform repeated sampling from the preceding step to construct the pairs. For the instance level, we construct contrastive instances by selecting two questions with substantially different difficulty. Finally, we leverage the fourier transform to analyze fluctuation patterns. Figure 2 (*Lower*) illustrates the time- and frequency-domain profiles of MoN for sample pairs under different failure modes.

Obs. 2 **Corresponding experts within the MoN exhibit distinct fluctuation patterns across failure levels.** Specifically, experts associated with intra-step failures display sharp spikes, whereas those linked to inter-step failures exhibit periodic fluctuations; regarding instance level patterns, experts maintain sustained activation for complex instances but undergo rapid collapse for simpler ones.

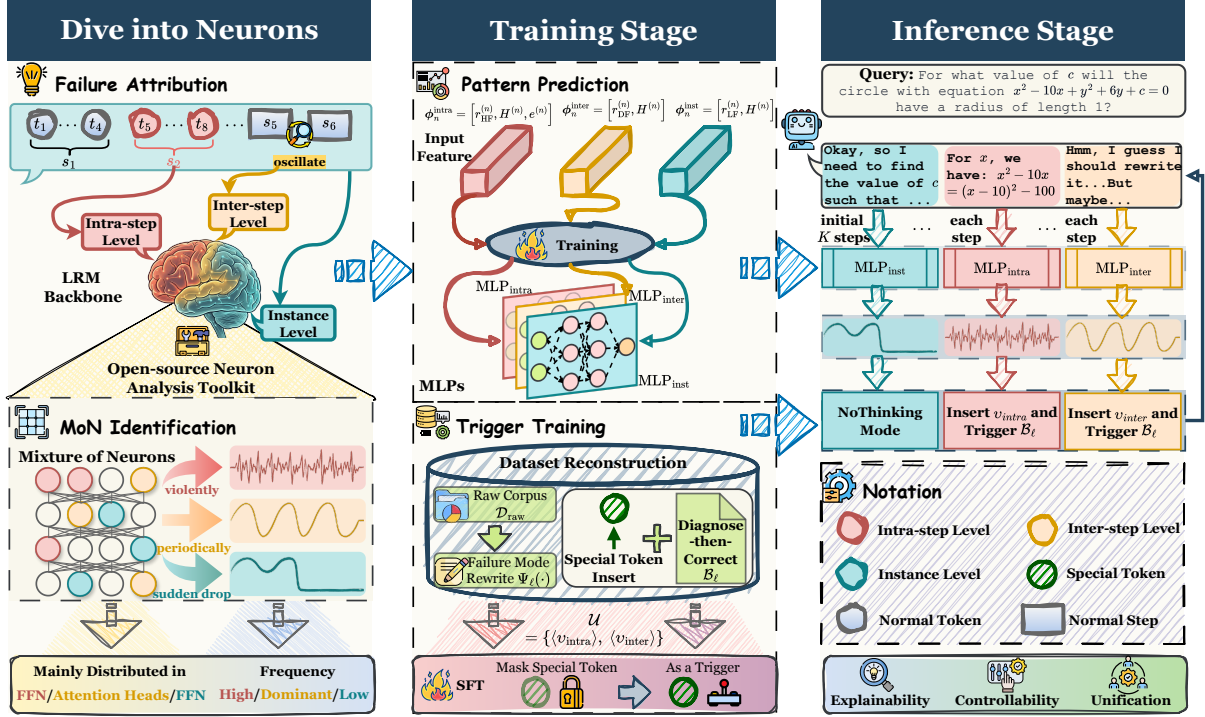


Figure 3: The overview of our proposed NeuReasoner.

3 Method

In this section, we elaborate on our proposed framework, NeuReasoner. Specifically, (i) **Pattern Prediction**, building upon the preceding in-depth qualitative analysis of neurons, we train a set of lightweight MLPs to quantify and predict distinct fluctuation patterns (► Section 3.1); (ii) **Trigger Training**, by reconstructing the original dataset according to failure modes, we employ SFT to enable the model to utilize special tokens as triggers to elicit specific behavioral patterns (► Section 3.2); and (iii) **Online Monitoring**, during inference, we deploy MLPs for parallel online monitoring, inserting special tokens upon the emergence of failure modes to induce specific behaviors, thereby achieving self-improvement (► Section 3.3).

3.1 Pattern Prediction

Having identified the fluctuation patterns of failure modes in Section 2.2, we aim to train lightweight MLPs to monitor and predict the fluctuations of MoN. Motivated by our observations and prior works (Hu et al., 2025; Yi et al., 2023), we employ the fourier transform to construct input features.

Preliminary. For an arbitrary neuron n , we denote its scalar activation values over a token sequence of length T as a_1, a_2, \dots, a_T . Let the activation magnitude be $x_t = |a_t|$, with the corresponding mean given by $\mu = \frac{1}{T} \sum_{t=1}^T x_t$. To prevent the mean from dominating the spectrum, we first per-

form mean removal to obtain the zero-centered sequence $y_t = x_t - \mu$. We apply the Discrete Fourier Transform (DFT) (Cooley and Tukey, 1965) to the real-valued sequence y_t . Let the number of non-redundant frequency points be $F = \lfloor \frac{T}{2} \rfloor + 1$. The frequency domain coefficients are defined as $Y(f) = \sum_{t=1}^T y_t \exp(-i\frac{2\pi}{T}(f-1)(t-1))$, where $f = 1, 2, \dots, F$ serves as the discrete frequency index. Consequently, the power spectrum is given by $P(f) = |Y(f)|^2$, where $f = 1$ corresponds to the Direct Current (DC) component. To construct length-invariant and comparable frequency-domain statistics, we exclude the DC component to obtain the power distribution:

$$\tilde{P}(f) = \frac{P(f)}{\sum_{j=2}^F P(j) + \varepsilon}, f = 2, \dots, F, \quad (2)$$

where $\varepsilon > 0$ is a constant for numerical stability. Finally, we have the normalized spectral entropy:

$$H = -\frac{1}{\log(F-1)} \sum_{f=2}^F \tilde{P}(f) \log(\tilde{P}(f)). \quad (3)$$

I) Intra-step level. Given that intra-step failure modes are characterized by intense fluctuations, manifesting as stronger high-frequency components and more dispersed spectra, we focus on extracting the high-frequency energy ratio, spectral entropy, and total variation energy. We first define the high-frequency set as the latter half of the

spectrum $\mathcal{F}_{\text{HF}} = \{f \mid \lfloor \frac{F}{2} \rfloor + 1 \leq f \leq F\}$. The high-frequency energy ratio is calculated as $r_{\text{HF}} = \sum_{f \in \mathcal{F}_{\text{HF}}} \tilde{P}(f)$. Additionally, we define the total variation energy $e = \log(\sum_{f=2}^F P(f) + \varepsilon)$. Consequently, the input feature is formed as $\phi_n^{\text{intra}} = [r_{\text{HF}}^{(n)}, H^{(n)}, e^{(n)}] \in \mathbb{R}^3$.

II) Inter-step level. Given that inter-step level failure modes are accompanied by periodic oscillations, signifying the prolonged dominance of energy at a specific non-zero frequency, we focus on extracting the dominant frequency energy ratio and spectral entropy. Specifically, the dominant frequency energy ratio is defined as $r_{\text{dom}} = \max_{2 \leq f \leq F} \tilde{P}(f)$. Formally, we have $\phi_n^{\text{inter}} = [r_{\text{dom}}^{(n)}, H^{(n)}] \in \mathbb{R}^2$.

III) Instance level. Regarding instance-level failure modes, compared to hard queries, easy queries exhibit a distinct pattern characterized by initial intense fluctuations followed by a sharp drop and subsequent stabilization. In the frequency domain, this manifests as a concentration of energy in low-frequency components and lower spectral entropy. To quantify this, we first define the low-frequency set as the former half of the spectrum $\mathcal{F}_{\text{LF}} = \{f \mid 2 \leq f \leq \lfloor \frac{F}{2} \rfloor\}$. The low-frequency energy ratio is calculated as $r_{\text{LF}} = \sum_{f \in \mathcal{F}_{\text{LF}}} \tilde{P}(f)$. Formally, we have $\phi_n^{\text{inst}} = [r_{\text{LF}}^{(n)}, H^{(n)}] \in \mathbb{R}^2$.

MLPs training. For *intra-step* and *inter-step* levels, we maintain a sliding window tracking the most recent k steps, which shifts to discard the earliest step upon detecting a new step separator (“\n \n”) during token generation. Notably, we optimize feature updates within this variable-length window to achieve an approximate $O(1)$ time complexity. For the *instance* level, consistent with Section 2.1, we directly utilize the initial K steps. Further details on MLPs training can be found in Appendix E.3.

3.2 Trigger Training

Leveraging the trained MLPs to localize failure modes within the sliding window, we introduce special tokens as triggers to enforce controllable self-correction. Specifically, for the *intra-step* and *inter-step* levels, we employ dataset reconstruction followed by SFT to condition the model to execute diagnose-then-correct behaviors upon encountering these tokens. In contrast, for the *instance* level, we adopt a direct, training-free strategy, which will be elaborated upon in the next subsection.

Data Reconstruction. To enable the model to learn the mechanism where special tokens serve as triggers to elicit specific behavioral patterns, we reconstruct the raw dataset $\mathcal{D}_{\text{raw}} = \{(\mathbf{u}^{(i)}, \mathbf{v}^{(i)})\}_{i=1}^{|\mathcal{D}_{\text{raw}}|}$, where \mathbf{u} and \mathbf{v} denote the input and output, respectively. We first apply a segmentation operation $\text{Seg}(\cdot)$ to divide \mathbf{v} into step-level subsequences $\mathbf{v} = \langle \pi_1 \parallel \dots \parallel \pi_K \rangle$. Let \mathcal{J} denote the set of critical steps to be perturbed. We define the failure mode levels as $\ell \in \{\text{intra}, \text{inter}\}$ and introduce a rewriting operation $\Psi_\ell(\cdot)$, yielding:

$$\tilde{\pi}_j^{(\ell)} = \Psi_\ell(\mathbf{u}, \langle \pi_1, \dots, \pi_K \rangle, j), j \in \mathcal{J}. \quad (4)$$

We introduce a set of mode-specific triggers $\mathcal{U} = \{\langle v_{\text{intra}}, v_{\text{inter}} \rangle\}$, which are inserted immediately following $\Psi_\ell(\cdot)$. For each level ℓ , we associate a diagnose-then-correct template $\mathcal{B}_\ell = \langle \mathbf{p}_\ell, \mathbf{d}_\ell, \mathbf{c}_\ell \rangle$, where \mathbf{p}_ℓ serves as a prompt signaling the detected failure mode, followed by the diagnostic (\mathbf{d}_ℓ) and corrective (\mathbf{c}_ℓ) behaviors. Consequently, we obtain the reconstructed dataset $\hat{\mathcal{D}} = \{(\mathbf{u}^{(i)}, \hat{\mathbf{v}}^{(i)})\}_{i=1}^{|\hat{\mathcal{D}}|}$, where the constructed output is:

$$\hat{\mathbf{v}}^{(\ell, j)} = \langle \pi_{<j}, \tilde{\pi}_j^{(\ell)}, \langle v_\ell \rangle, \mathcal{B}_\ell, \pi_{>j} \rangle. \quad (5)$$

See Appendix G for the detailed reconstructed data.

SFT. Our training objective maximizes the following factorized likelihood over step blocks:

$$\begin{aligned} & P_\theta(\pi_{<j} \mid \mathbf{u}) \cdot P_\theta(\tilde{\pi}_j^{(\ell)} \mid \mathbf{u}, \pi_{<j}) \\ & \cdot P_\theta(\mathcal{B}_\ell \mid \mathbf{u}, \pi_{<j}, \tilde{\pi}_j^{(\ell)}, \langle v_\ell \rangle) \\ & \cdot P_\theta(\pi_{>j} \mid \mathbf{u}, \pi_{<j}, \tilde{\pi}_j^{(\ell)}, \langle v_\ell \rangle, \mathcal{B}_\ell), \end{aligned} \quad (6)$$

where θ denotes the parameters. Training follows standard next-token prediction with a label mask:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(\mathbf{u}, \hat{\mathbf{v}}) \sim \hat{\mathcal{D}}} \sum_{t \in \mathcal{I}(\mathbf{u}, \hat{\mathbf{v}})} \log P_\theta(y_t \mid \mathbf{u}, y_{<t}), \quad (7)$$

where $\{y_t\}$ enumerates tokens in the concatenated sequence $\langle \mathbf{u}, \hat{\mathbf{v}} \rangle$ and $\mathcal{I}(\mathbf{u}, \hat{\mathbf{v}})$ indexes the tokens that contribute to the loss. Notably, we mask \mathbf{u} and $\langle v_\ell \rangle$ during training, ensuring the trigger operates solely as an inference-time control signal.

3.3 Online Monitoring

During inference, we run our trained lightweight MLPs in parallel to conduct online monitoring of the MoN over designated token sequences. For *intra-step* and *inter-step* levels, this sequence corresponds to the sliding window defined in Section

3.1. Upon detecting fluctuation patterns characteristic of failure modes at level $\ell \in \{\text{intra}, \text{inter}\}$, we intervene by forcibly inserting the trigger token $\langle v_\ell \rangle$. Formally, letting x denote the current decoding prefix with length $\tau = |x|$, we impose a hard constraint on the next token $y_{\tau+1} := \langle v_\ell \rangle$, and continue autoregressive decoding as:

$$y_{\tau+k} \sim P_\theta(\cdot | x, \langle v_\ell \rangle, y_{\tau+1:\tau+k-1}), k \geq 2. \quad (8)$$

This intervention activates the diagnose-then-correct behavior pattern learned during SFT. In contrast, for the *instance* level, the monitored sequence comprises the initial K steps of the reasoning process (consistent with Section 3.1). Upon detecting the corresponding fluctuations, we directly insert the prompt (“Okay, I have finished thinking.”), thereby triggering a transition to the *NoThinking* mode (Ma et al., 2025).

4 Experiments

In this section, we conduct extensive experiments to address the following research questions: (**RQ1**) Can *NeuReasoner* achieve dual superiority in terms of performance and cost? (**RQ2**) Does *NeuReasoner* demonstrate effective scalability? (**RQ3**) Specifically, how does *NeuReasoner* realize improvements during the reasoning process? (**RQ4**) What are the distinct contributions of the MLPs within *NeuReasoner*?

4.1 Experimental Setup

Backbones. We conduct experiments using representative open-source LRMs with diverse architectures from different families. To demonstrate the effectiveness and scalability of *NeuReasoner*, the selected models span a wide range of scales, from 7B to 70B: **I) Qwen family**, DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025), Qwen3-8B-thinking, and Qwen3-32B-thinking (Yang et al., 2025a); and **II) Llama family**, DeepSeek-R1-Distill-Llama-8B and DeepSeek-R1-Distill-Llama-70B.

Baselines. We compare *NeuReasoner* against a comprehensive set of representative baselines, categorized into three groups: **I) Vanilla Model**, referring to the original backbone LRM itself; **II) Training-free**, including Think or Not (Yong et al., 2025), AlphaOne (Zhang et al., 2025d), and Self-Consistency (Wang et al., 2023); and **III) RL-based**, including DAST (Shen et al., 2025b), RL + Length Penalty (Arora and Zanette, 2025), GRPO (Shao et al., 2024), S-GRPO (Dai et al., 2025) and DAPO (Yu et al., 2025).

Benchmarks. We conduct extensive evaluations of *NeuReasoner* on five benchmarks spanning three complex reasoning domains: **I) Mathematical Reasoning**, including GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2023), AIME24 and AIME25 (MAA Committees); **II) Scientific Reasoning**, including GPQA-Diamond (Rein et al., 2024); and **III) Code Reasoning**, including LiveCodeBench (Jain et al., 2025).

Implementation details. The sliding window length k for intra-step and inter-step levels, as well as the initial step count K for the instance level, are selected from the set $\{2, 4, 8\}$. The failure mode detector (Section 2.2), the rewriting operation $\Psi_\ell(\cdot)$ (Equation 4), and the diagnose-then-correct operator \mathcal{B}_ℓ (Equation 5), are consistently implemented by gpt-5. For the segmentation operator $\text{Seg}(\cdot)$, we implement step-level splitting using paragraph delimiters (“\n\n”). We use BS-17k (Labs, 2025) as the raw corpus for reconstruction and conduct full-parameter fine-tuning. Following (Guo et al., 2025), we use a sampling temperature of 0.6 and top- p of 0.95. We report Pass@1 by averaging results over 5 repeated runs. Detailed technical details are provided in Appendix E. All prompts utilized are detailed in Appendix G.

4.2 Main Results (RQ1)

To address RQ1, we conduct extensive evaluations on six benchmarks, comparing *NeuReasoner* against nine competitive baseline methods. Experimental results are reported in Table 1 and 5.

Obs.⊙ *NeuReasoner* achieves dual superiority in both performance and token cost. Overall, compared to the vanilla model, *NeuReasoner* delivers performance gains ranging from 0.3 ~ 7.8 (3.2% ~ 27.0%), while simultaneously reducing token consumption by 19.6% ~ 63.3%. Notably, on DeepSeek-R1-Distill-Qwen-7B + MATH500, *NeuReasoner* improves accuracy by 5.0 \uparrow and cuts token usage by 48.1% \downarrow , outperforming all baselines in both metrics. Even against DAPO, the most competitive baseline, *NeuReasoner* achieves a 0.9 \uparrow performance lead with a substantial 60.5% \downarrow reduction in token consumption.

Obs.⊗ *NeuReasoner* exhibits robust cross-task generalization. On DeepSeek-R1-Distill-Qwen-32B, while DAPO performs adequately on math and science tasks, it struggles on LiveCodeBench (showing a negligible 0.1 \uparrow improvement with a 22.1% \uparrow cost overhead). In stark contrast, *NeuReasoner* achieves a 1.4 \uparrow performance gain

Model	Method	AIME25		MATH500		GSM8K		GPQA-Diamond		LiveCodeBench	
		Pass@1 ↑	Token ↓	Pass@1 ↑	Token ↓	Pass@1 ↑	Token ↓	Pass@1 ↑	Token ↓	Pass@1 ↑	Token ↓
DeepSeek-R1-Distill-Qwen-7B	Vanilla	43.3	11454	91.8	2887	92.4	442	49.2	8016	36.9	9072
	DAST	41.1	7904	91.6	1330	91.8	301	48.8	3635	36.8	4280
	Think or Not	38.9	4760	92.0	1103	92.9	264	47.0	3390	36.4	7397
	AlphaOne	44.4	8921	92.5	3791	93.4	459	<u>50.5</u>	8591	37.5	8177
	RL + LP	43.3	6255	93.4	1322	92.5	291	50.3	<u>3209</u>	36.2	6293
	GRPO	45.6	12006	93.9	2873	92.1	275	49.8	8890	<u>39.2</u>	10084
	S-GRPO	44.4	5794	93.2	1204	<u>93.7</u>	297	49.5	3107	36.9	5449
	DAPO	<u>46.7</u>	12307	<u>94.2</u>	3033	92.9	301	50.2	9102	40.1	11407
	NeuReasoner	47.8	5446	95.1	1197	94.0	271	51.3	4011	39.0	5311
	Δ	↑4.5	↓52.5%	↑3.3	↓58.5%	↑1.6	↓38.7%	↑2.1	↓50.0%	↑2.1	↓41.5%
DeepSeek-R1-Distill-Qwen-32B	Vanilla	58.9	8906	93.3	2337	93.9	438	60.8	6027	55.8	7125
	DAST	54.4	6647	93.5	1421	93.9	266	60.3	4048	56.0	3419
	Think or Not	58.9	4711	92.9	1410	93.3	<u>247</u>	60.4	3706	54.8	5216
	AlphaOne	61.1	8994	94.4	3037	94.1	433	61.3	6771	<u>56.1</u>	8223
	RL + LP	<u>62.2</u>	6124	93.1	<u>1379</u>	94.7	239	60.4	3596	55.9	4729
	GRPO	<u>62.2</u>	8990	93.4	3012	<u>94.4</u>	420	60.4	7123	<u>56.1</u>	8974
	S-GRPO	60.0	5334	94.2	1556	94.7	269	61.3	3119	54.9	<u>4404</u>
	DAPO	63.3	9093	<u>94.6</u>	3551	94.3	443	<u>61.4</u>	7390	55.9	8700
	NeuReasoner	63.3	4702	96.3	1318	94.2	247	62.3	3498	57.2	4689
	Δ	↑4.4	↓47.2%	↑3.0	↓43.6%	↑0.3	↓43.6%	↑1.5	↓42.0%	↑1.4	↓34.2%
Qwen3-8B-thinking	Vanilla	61.1	12490	95.9	4486	94.1	1573	58.8	6638	54.5	8549
	DAST	60.0	6055	96.0	2158	94.1	483	58.4	3007	54.7	4987
	Think or Not	62.2	6674	96.3	2408	94.2	599	59.1	3678	53.9	6009
	AlphaOne	<u>63.3</u>	8607	96.7	4311	94.5	1095	<u>59.8</u>	6002	<u>56.2</u>	8607
	RL + LP	58.9	7639	96.6	2692	94.1	823	59.4	3139	55.9	6792
	GRPO	62.2	11098	96.7	4225	94.4	1373	59.1	7472	55.3	9073
	S-GRPO	61.1	7331	96.5	2576	<u>94.6</u>	775	59.6	3046	54.3	5998
	DAPO	62.2	12004	<u>96.9</u>	4609	94.7	1553	59.4	8008	56.3	9937
	NeuReasoner	64.4	7279	97.2	<u>2402</u>	<u>94.6</u>	<u>588</u>	61.4	5153	<u>56.2</u>	6002
	Δ	↑3.3	↓41.7%	↑1.3	↓46.5%	↑0.5	↓62.6%	↑2.6	↓22.4%	↑1.7	↓29.8%
Qwen3-32B-thinking	Vanilla	68.9	11589	96.8	4318	94.3	1435	65.3	5475	64.7	8725
	DAST	68.9	6504	96.8	2417	94.2	609	65.7	<u>2918</u>	64.6	5108
	Think or Not	65.6	6772	97.1	2179	94.5	641	64.5	1713	62.9	7337
	AlphaOne	71.1	8569	<u>97.8</u>	3170	94.4	1090	<u>66.8</u>	5591	<u>66.1</u>	9003
	RL + LP	66.7	6854	97.2	2701	94.7	822	66.0	4362	65.1	7382
	GRPO	70.0	12487	97.1	4641	94.5	1558	66.2	6151	65.4	9546
	S-GRPO	71.1	6389	97.3	2566	<u>94.6</u>	809	66.5	4151	63.9	7560
	DAPO	<u>72.2</u>	12403	97.7	5152	94.7	1970	65.8	6099	65.9	10124
	NeuReasoner	73.3	6308	97.9	<u>2398</u>	<u>94.6</u>	574	67.3	4043	66.9	<u>7012</u>
	Δ	↑4.4	↓45.6%	↑1.1	↓44.5%	↑0.3	↓60.0%	↑2.0	↓26.2%	↑2.2	↓19.6%
DeepSeek-R1-Distill-Llama-8B	Vanilla	28.9	11548	86.2	3635	92.3	606	46.3	8341	38.9	9588
	DAST	32.2	8438	87.0	2458	91.9	<u>388</u>	46.1	4410	38.4	4091
	Think or Not	30.0	7158	87.4	<u>1954</u>	92.5	257	46.8	3729	38.4	6916
	AlphaOne	<u>34.4</u>	9005	89.1	3804	93.1	598	<u>47.6</u>	8569	41.2	8421
	RL + LP	30.0	5897	89.4	2290	92.3	446	45.3	3299	39.8	<u>3491</u>
	GRPO	30.0	11987	89.6	3309	92.9	571	46.6	8783	40.0	10123
	S-GRPO	31.1	5426	89.1	2195	93.2	432	47.0	<u>3624</u>	39.2	3349
	DAPO	<u>34.4</u>	12399	<u>90.0</u>	3732	93.5	4004	47.8	9110	<u>41.7</u>	11394
	NeuReasoner	36.7	6076	91.2	1887	<u>93.4</u>	419	47.8	4539	42.5	3522
	Δ	↑7.8	↓47.4%	↑5.0	↓48.1%	↑1.1	↓30.9%	↑1.5	↓45.6%	↑3.6	↓63.3%
DeepSeek-R1-Distill-Llama-70B	Vanilla	47.8	8909	94.1	2433	94.0	432	64.5	5881	55.6	7258
	DAST	45.6	5932	94.2	1563	93.9	<u>239</u>	63.5	4026	56.3	3975
	Think or Not	46.7	4414	94.4	1360	93.7	241	64.6	3544	52.9	5668
	AlphaOne	50.0	8689	95.3	2009	94.2	427	65.3	4322	56.8	<u>3742</u>
	RL + LP	45.6	5967	93.8	<u>1149</u>	94.4	255	65.0	4101	56.2	5209
	GRPO	51.1	8715	94.5	3167	<u>94.7</u>	473	64.8	6274	56.4	7561
	S-GRPO	48.9	<u>5896</u>	95.1	1252	94.5	249	65.2	<u>4001</u>	54.8	4689
	DAPO	53.3	9054	<u>95.4</u>	3776	94.9	509	<u>65.7</u>	6968	57.3	8049
	NeuReasoner	<u>52.2</u>	5907	96.8	1125	94.9	208	66.8	4084	57.5	3308
	Δ	↑4.4	↓33.7%	↑2.7	↓53.8%	↑0.9	↓51.9%	↑2.3	↓30.6%	↑1.9	↓54.4%

Table 1: Main results. Best results are highlighted in **bold**, with runners-up underlined.

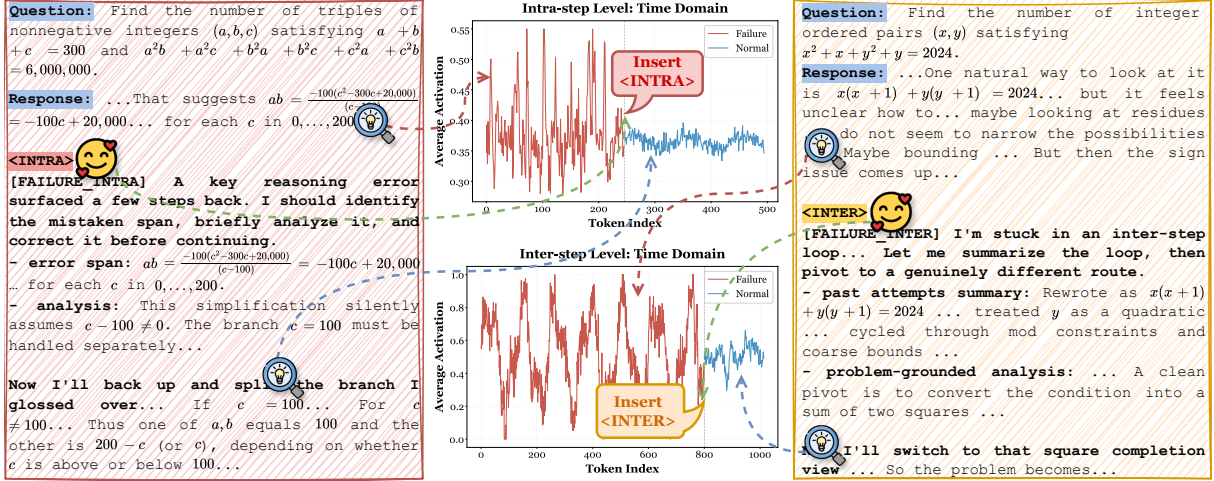


Figure 4: Case studies of NeuReasoner.

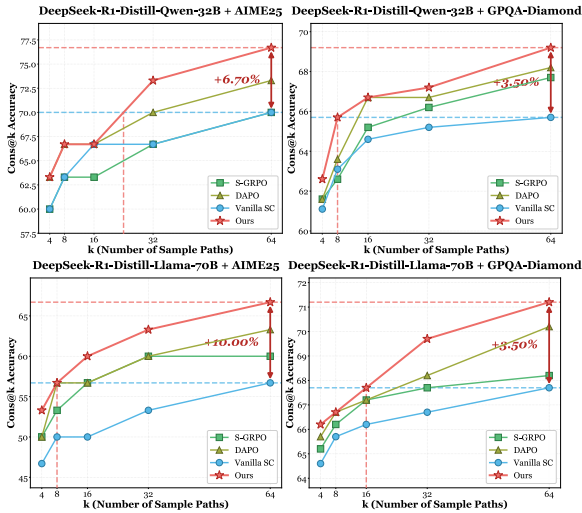


Figure 5: Test-time scalability under self-consistency.

with a 34.2% \downarrow cost reduction on this benchmark, while securing state-of-the-art results across AIME25, MATH500, and GPQA-Diamond.

4.3 Scalability Analysis ($\mathcal{R}Q2$)

We examine scalability across two dimensions: *model scale* and *test-time*. To evaluate test-time scalability, we benchmark NeuReasoner against baselines using Cons@ k , where $k \in \{4, 8, 16, 32, 64\}$, as illustrated in Figure 5.

Obs. 8 NeuReasoner demonstrates robust scalability across both model scale and test-time dimensions. Regarding model scale, Table 1 reveals that on DeepSeek-R1-Distill-Llama series + AIME25, while DAST suffers a performance decline, shifting from a 3.3 \uparrow gain at 8B to a 2.2 \downarrow drop at 70B relative to the vanilla, NeuReasoner maintains consistent gains of 7.8 \uparrow and 4.4 \uparrow at 8B and 70B, respectively. Regarding test-time scalability, Figure 5 illustrates that NeuReasoner achieves superior Cons@ k performance across nearly all k . On DeepSeek-R1-Distill-Qwen-32B + GPQA-

Model	Method	MATH500		GPQA	
		Pass@1 \uparrow	Token \downarrow	Pass@1 \uparrow	Token \downarrow
DS-R1-Distill-Qwen-32B	w/o MLP _{all}	93.5	2469	63.9	6010
	w/o MLP _{intra}	94.5	1412	64.4	3441
	w/o MLP _{inter}	94.7	1434	64.9	3219
	w/o MLP _{inst}	95.8	2389	65.4	6118
	NeuReasoner	96.3	1318	66.2	3274
DS-R1-Distill-Llama-70B	w/o MLP _{all}	94.2	2523	65.9	5721
	w/o MLP _{intra}	95.9	1293	66.8	4065
	w/o MLP _{inter}	96.3	1334	67.1	4007
	w/o MLP _{inst}	96.3	2476	67.6	5703
	NeuReasoner	96.8	1125	67.8	3936

Table 2: Ablation study of NeuReasoner.

Diamond, NeuReasoner at $k = 8$ matches the performance of the vanilla at $k = 64$. Furthermore, at $k = 64$, NeuReasoner outperforms the vanilla by 3.5% \sim 10.0%, underscoring its scalability.

4.4 Case Study ($\mathcal{R}Q3$)

Obs. 9 NeuReasoner demonstrates explainability and controllability. As visualized in Figure 4 for intra- and inter-step levels, NeuReasoner successfully detects failure mode fluctuations during inference and triggers the diagnose-then-correct behavior via special token insertion. Crucially, we observe corresponding shifts in MoN activation dynamics post-intervention. These findings collectively substantiate the explainability and controllability.

4.5 Framework Analysis ($\mathcal{R}Q4$)

To address $\mathcal{R}Q4$, we evaluate four variants by ablating specific components: (1) w/o MLP_{intra}, (2) w/o MLP_{inter}, (3) w/o MLP_{inst}, and (4) w/o MLP_{all}. As shown in Table 2, the removal of MLP_{intra} induces the most significant drop, corroborating the predominance of intra-step failures. While removing MLP_{inter} yields minor deficits, ablating MLP_{inst} triggers a drastic surge in token consumption, validating the necessity of difficulty-aware strategies. See Appendix F.3 for sensitivity analysis.

5 Conclusion

In this work, we categorize failure modes across different levels and highlight significant gaps in current research. To address these, we conduct a fine-grained neuronal analysis to identify MoN and their fluctuation patterns. Based on these insights, we propose **NeuReasoner**, an explainable, controllable, and unified framework driven by MoN, which demonstrates superior performance and cost-efficiency across multiple domains. We believe it paves the way towards human-like reasoning.

Limitations

While **NeuReasoner** demonstrates significant potential in explainability, controllability, and unification, achieving superior performance and token efficiency across diverse tasks and backbone models, we acknowledge certain limitations. The integration of monitoring MLPs incurs inference overhead. Despite implementation optimizations achieving approximate $\mathcal{O}(1)$ time complexity for feature updates (► Appendix E.3), marginal latency increases persist; a detailed runtime comparison is provided in Appendix F.2. Furthermore, **NeuReasoner** is not currently a fully automated, end-to-end pipeline. We leave the further optimization of this aspect as a direction for future research.

Acknowledgement

This work is supported by the State Key Laboratory of General Artificial Intelligence; the National Natural Science Foundation of China (Grant No. 625B2001); and the National Natural Science Foundation of China (Grant No. 62276006).

References

- Shengnan An, Xunliang Cai, Xuezhi Cao, Xiaoyu Li, Yehao Lin, Junlin Liu, Xinxuan Lv, Dan Ma, Xuanlin Wang, Ziwen Wang, and Shuang Zhou. 2025. [Amo-bench: Large language models still struggle in high school math competitions](#). *CoRR*, abs/2510.26768.
- Daman Arora and Andrea Zanette. 2025. [Training language models to reason efficiently](#). In *The Thirtieth Annual Conference on Neural Information Processing Systems*.
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. [Graph of thoughts: Solving elaborate problems with large language models](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.
- Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2025. [Forest-of-thought: Scaling test-time compute for enhancing LLM reasoning](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.
- Yupeng Chang, Yi Chang, and Yuan Wu. 2026. [BA-loRA: Bias-alleviating low-rank adaptation to mitigate catastrophic inheritance in large language models](#). In *The Fourteenth International Conference on Learning Representations*.
- Yupeng Chang, Chenlu Guo, Yi Chang, and Yuan Wu. 2025a. [Lora-mgpo: Mitigating double descent in low-rank adaptation via momentum-guided perturbation optimization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 648–659. Association for Computational Linguistics.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. [A survey on evaluation of large language models](#). *ACM Trans. Intell. Syst. Technol.*, 15(3):39:1–39:45.
- Yurui Chang, Bochuan Cao, Yujia Wang, Jinghui Chen, and Lu Lin. 2025b. [Jopa: Explaining large language model’s generation via joint prompt attribution](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 22106–22122. Association for Computational Linguistics.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé 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](#). *CoRR*, abs/2107.03374.
- Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025a. [Towards reasoning era: A survey of long chain-of-thought for reasoning large language models](#). *CoRR*, abs/2503.09567.
- Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2024a. [Do NOT think that much for 2+3=? on the overthinking of o1-like llms](#). *CoRR*, abs/2412.21187.

- Yezeng Chen, Zui Chen, and Yi Zhou. 2024b. [Brain-inspired two-stage approach: Enhancing mathematical reasoning by imitating human thought processes](#). *CoRR*, abs/2403.00800.
- Zhaorun Chen, Zhuokai Zhao, Zhihong Zhu, Ruiqi Zhang, Xiang Li, Bhiksha Raj, and Huaxiu Yao. 2024c. [Autoprm: Automating procedural supervision for multi-step reasoning via controllable question decomposition](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1346–1362. Association for Computational Linguistics.
- Zhiwei Chen, Yupeng Hu, Zhiheng Fu, Zixu Li, Jiale Huang, Qinlei Huang, and Yinwei Wei. 2026. [INTENT: invariance and discrimination-aware noise mitigation for robust composed image retrieval](#). In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 20463–20471. AAAI Press.
- Zhiwei Chen, Yupeng Hu, Zixu Li, Zhiheng Fu, Xueming Song, and Liqiang Nie. 2025b. [OFFSET: segmentation-based focus shift revision for composed image retrieval](#). In *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pages 6113–6122. ACM.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. [What you can cram into a single \$\\$&!#*\$ vector: Probing sentence embeddings for linguistic properties](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.
- James W Cooley and John W Tukey. 1965. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301.
- Mz Dai, Chenxu Yang, and Qingyi Si. 2025. [S-GRPO: Early exit via reinforcement learning in reasoning models](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Haonan Dong, Wenhao Zhu, Guojie Song, and Liang Wang. 2025. [AuroRA: Breaking low-rank bottleneck of loRA with nonlinear mapping](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Jeremias Ferrao, Ezgi Basar, Khondoker Ittehadul Islam, and Mahrokh Hassani. 2025. [What really counts? examining step and token level attribution in multi-lingual cot reasoning](#). *Preprint*, arXiv:2511.15886.
- Andrey V. Galichin, Alexey Dontsov, Polina Druzhinina, Anton Razzhigaev, Oleg Y. Rogov, Elena Tutubalina, and Ivan V. Oseledets. 2025. [I have covered all the bases here: Interpreting reasoning features in large language models via sparse autoencoders](#). *CoRR*, abs/2503.18878.
- Çağlar Gülçehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. [Reinforced self-training \(rest\) for language modeling](#). *CoRR*, abs/2308.08998.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.
- Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. 2025. [Circuit-tracer: A new library for finding feature circuits](#). In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 239–249, Suzhou, China. Association for Computational Linguistics.
- Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. [Training large language models to reason in a continuous latent space](#). *CoRR*, abs/2412.06769.
- Yanji He, Yuxin Jiang, Yiwen Wu, Bo Huang, Jiaheng Wei, and Wei Wang. 2026. [Idea: An interpretable and editable decision-making framework for llms via verbal-to-numeric calibration](#). *Preprint*, arXiv:2604.12573.
- 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](#). *Preprint*, arXiv:2103.03874.
- John Hewitt and Christopher D. Manning. 2019. [A structural probe for finding syntax in word representations](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4129–4138. Association for Computational Linguistics.
- Xiangyu Hong, Che Jiang, Kai Tian, Biqing Qi, Youbang Sun, Ning Ding, and Bowen Zhou. 2025.

- Depass: Unified feature attributing by simple decomposed forward pass. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Yifan Hu, Jie Yang, Tian Zhou, Peiyuan Liu, Yujin Tang, Rong Jin, and Liang Sun. 2025. Bridging past and future: Distribution-aware alignment for time series forecasting. *CoRR*, abs/2509.14181.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Live-codebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*.
- Yiyang Jiang, Guangwu Qian, Jiabin Wu, Qi Huang, Qing Li, Yongkang Wu, and Xiao-Yong Wei. 2026. Self-paced learning for images of antinuclear antibodies. *IEEE Trans. Medical Imaging*, 45(4):1661–1672.
- Yiyang Jiang, Wengyu Zhang, Xulu Zhang, Xiaoyong 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, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 7249–7258. ACM.
- Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, Fan Yang, Mengnan Du, and Yongfeng Zhang. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 558–573. Association for Computational Linguistics.
- Philip N Johnson-Laird. 2010. Mental models and human reasoning. *Proceedings of the National Academy of Sciences*, 107(43):18243–18250.
- Nancy Kanwisher. 2010. Functional specificity in the human brain: a window into the functional architecture of the mind. *Proceedings of the national academy of sciences*, 107(25):11163–11170.
- Deqian Kong, Minglu Zhao, Dehong Xu, Bo Pang, Shu Wang, Edouardo Honig, Zhangzhang Si, Chuan Li, Jianwen Xie, Sirui Xie, and Ying Nian Wu. 2025. Scalable language models with posterior inference of latent thought vectors. *CoRR*, abs/2502.01567.
- Bespoke Labs. 2025. Bespoke-stratos: The unreasonable effectiveness of reasoning distillation. <https://www.bespokelabs.ai/blog/bespoke-stratos-the-unreasonable-effectiveness-of-reasoning-distillation>. Accessed: 2025-01-22.
- Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Chengpeng Li, Zhengyang Tang, Ziniu Li, Mingfeng Xue, Keqin Bao, Tian Ding, Ruoyu Sun, Benyou Wang, Xiang Wang, Junyang Lin, and Dayiheng Liu. 2025a. Cort: Code-integrated reasoning within thinking. *CoRR*, abs/2506.09820.
- Jierui Li, Hung Le, Yingbo Zhou, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. 2025b. Codetree: Agent-guided tree search for code generation with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 3711–3726. Association for Computational Linguistics.
- Junlong Li, Daya Guo, Dejian Yang, Runxin Xu, Yu Wu, and Junxian He. 2025c. Code/o: Condensing reasoning patterns via code input-output prediction. *CoRR*, abs/2502.07316.
- Junxian Li, Di Zhang, Xunzhi Wang, Zeying Hao, Jingdi Lei, Qian Tan, Cai Zhou, Wei Liu, Yaotian Yang, Xinrui Xiong, Weiyun Wang, Zhe Chen, Wenhai Wang, Wei Li, Mao Su, Shufei Zhang, Wanli Ouyang, Yuqiang Li, and Dongzhan Zhou. 2025d. Chemvlm: Exploring the power of multimodal large language models in chemistry area. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 415–423. AAAI Press.
- Mengdi Li, Jiaye Lin, Xufeng Zhao, Wenhao Lu, Peilin Zhao, Stefan Wermter, and Di Wang. 2025e. Curriculum-rlaif: Curriculum alignment with reinforcement learning from AI feedback. *CoRR*, abs/2505.20075.
- Songze Li, Xiaoke Guo, Tianqi Liu, Biao Yi, Zhaoyan Gong, Zhiqiang Liu, Huajun Chen, and Wen Zhang. 2026a. What’s missing in screen-to-action? towards a ui-in-the-loop paradigm for multimodal gui reasoning. *Preprint*, arXiv:2604.06995.
- Yang Li. 2025. Policy guided tree search for enhanced LLM reasoning. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.
- Zixu Li, Yupeng Hu, Zhiwei Chen, Qinlei Huang, Guozhi Qiu, Zhiheng Fu, and Meng Liu. 2026b.

- Retrack: Evidence-driven dual-stream directional anchor calibration network for composed video retrieval. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 23373–23381. AAAI Press.
- Zixu Li, Yupeng Hu, Zhiwei Chen, Shiqi Zhang, Qinlei Huang, Zhiheng Fu, and Yinwei Wei. 2026c. **HABIT: chrono-synergia robust progressive learning framework for composed image retrieval**. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 6762–6770. AAAI Press.
- Huanxuan Liao, Shizhu He, Yupu Hao, Xiang Li, Yuanzhe Zhang, Jun Zhao, and Kang Liu. 2025. **Skintern: Internalizing symbolic knowledge for distilling better cot capabilities into small language models**. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 3203–3221. Association for Computational Linguistics.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. **Let’s verify step by step**. *Preprint*, arXiv:2305.20050.
- Jiaye Lin, Yifu Guo, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, Mingguang Chen, Hongzhang Liu, Ronghao Chen, Yangfan He, Daxin Jiang, Binxing Jiao, Chen Hu, and Huacan Wang. 2025. **Se-agent: Self-evolution trajectory optimization in multi-step reasoning with llm-based agents**. *CoRR*, abs/2508.02085.
- Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. **Deductive verification of chain-of-thought reasoning**. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Junlin Liu, Shengnan An, Shuang Zhou, Dan Ma, Shixiong Luo, Ying Xie, Yuan Zhang, Wenling Yuan, Yifan Zhou, Xiaoyu Li, Ziwen Wang, Xuezhi Cao, and Xunliang Cai. 2026. **General365: Benchmarking general reasoning in large language models across diverse and challenging tasks**. *Preprint*, arXiv:2604.11778.
- Peiyang Liu, Sen Wang, Xi Wang, Wei Ye, and Shikun Zhang. 2021. **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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3734–3739. Association for Computational Linguistics.
- Peiyang Liu, Xi Wang, Ziqiang Cui, and Wei Ye. 2025a. **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, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pages 874–883. ACM.
- 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, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pages 4099–4104. ACM.
- Zheng Liu, Hao Liang, Xijie Huang, Wentao Xiong, Qinhan Yu, Linzhuang Sun, Chong Chen, Conghui He, Bin Cui, and Wentao Zhang. 2024. **Synthvlm: High-efficiency and high-quality synthetic data for vision language models**. *CoRR*, abs/2407.20756.
- Zheng Liu, Mengjie Liu, Siwei Wen, Mengzhang Cai, Bin Cui, Conghui He, and Wentao Zhang. 2025b. **From uniform to heterogeneous: Tailoring policy optimization to every token’s nature**. *CoRR*, abs/2509.16591.
- Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. **O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning**. *CoRR*, abs/2501.12570.
- Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. 2025. **Reasoning models can be effective without thinking**. *Preprint*, arXiv:2504.09858.
- MAA Committees. Aime problems and solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.
- Shuaiyi Nie, Siyu Ding, Wenyuan Zhang, Linhao Yu, Tianmeng Yang, Yao Chen, Tingwen Liu, Weichong Yin, Yu Sun, and Hua Wu. 2026. **ATTNPO: attention-guided process supervision for efficient reasoning**. *CoRR*, abs/2602.09953.
- OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Ifimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. **Openai o1 system card**. *Preprint*, arXiv:2412.16720.
- Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2024. **REFINER: reasoning feedback on intermediate representations**. In *Proceedings of the*

- 18th Conference of the European Chapter of the Association for Computational Linguistics, *EACL 2024 - Volume 1: Long Papers*, St. Julian's, Malta, March 17-22, 2024, pages 1100–1126. Association for Computational Linguistics.
- Leonardo Ranaldi, Marco Valentino, and André Freitas. 2025. [Improving chain-of-thought reasoning via quasi-symbolic abstractions](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, *ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 17222–17240. Association for Computational Linguistics.
- David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. [Gpqa: A graduate-level google-proof q&a benchmark](#). In *First Conference on Language Modeling*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- Xuan Shen, Yizhou Wang, Xiangxi Shi, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. 2025a. [Efficient reasoning with hidden thinking](#). *CoRR*, abs/2501.19201.
- Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. 2025b. [DAST: Difficulty-adaptive slow-thinking for large reasoning models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2322–2331, Suzhou (China). Association for Computational Linguistics.
- Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. [The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity](#). *CoRR*, abs/2506.06941.
- Oscar Skea, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. [Layer by layer: Uncovering hidden representations in language models](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.
- Zhongxiang Sun, Qipeng Wang, Weijie Yu, Xiaoxue Zang, Kai Zheng, Jun Xu, Xiao Zhang, Yang Song, and Han Li. 2025. [Rearter: Retrieval-augmented reasoning with trustworthy process rewarding](#). In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pages 1251–1261. ACM.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2023. [Challenging big-bench tasks and whether chain-of-thought can solve them](#). In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13003–13051. Association for Computational Linguistics.
- Gemini Team. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *CoRR*, abs/2507.06261.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4593–4601. Association for Computational Linguistics.
- Hanlin Wang, Jian Wang, Chak Tou Leong, and Wenjie Li. 2025a. [Steca: Step-level trajectory calibration for LLM agent learning](#). In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 11597–11614. Association for Computational Linguistics.
- Jiayu Wang, Yifei Ming, Zixuan Ke, Caiming Xiong, Shafiq Joty, Aws Albarghouthi, and Frederic Sala. 2025b. [Beyond accuracy: Dissecting mathematical reasoning for llms under reinforcement learning](#). *CoRR*, abs/2506.04723.
- 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)*, *ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9426–9439. Association for Computational Linguistics.
- Tongxi Wang. 2026. [Fbs: Modeling native parallel reading inside a transformer](#). *Preprint*, arXiv:2601.21708.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. 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 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

- Siye Wu, Jian Xie, Yikai Zhang, Aili Chen, Kai Zhang, Yu Su, and Yanghua Xiao. 2025a. [ARM: Adaptive reasoning model](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Zhenhe Wu, Jian Yang, Jiaheng Liu, Xianjie Wu, Changzai Pan, Jie Zhang, Yu Zhao, Shuangyong Song, Yongxiang Li, and Zhoujun Li. 2025b. [Table-r1: Region-based reinforcement learning for table understanding](#). *CoRR*, abs/2505.12415.
- Xi Xiao, Chenrui Ma, Yunbei Zhang, Chen Liu, Zhuxuanzi Wang, Yanshu Li, Lin Zhao, Guosheng Hu, Tianyang Wang, and Hao Xu. 2026a. [Not all directions matter: Toward structured and task-aware low-rank adaptation](#). *CoRR*, abs/2603.14228.
- Xi Xiao, Yunbei Zhang, Xingjian Li, Tianyang Wang, Xiao Wang, Yuxiang Wei, Jihun Hamm, and Min Xu. 2025. [Visual instance-aware prompt tuning](#). In *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pages 2880–2889. ACM.
- Xi Xiao, Yunbei Zhang, Lin Zhao, Yiyang Liu, Xiaoying Liao, Zheda Mai, Xingjian Li, Xiao Wang, Hao Xu, Jihun Hamm, Xue Lin, Min Xu, Qifan Wang, Tianyang Wang, and Cheng Han. 2026b. [Prompt-based adaptation in large-scale vision models: A survey](#). *Trans. Mach. Learn. Res.*, 2026.
- Can Xie, Ruotong Pan, Xiangyu Wu, Yunfei Zhang, Jiayi Fu, Tingting Gao, and Guorui Zhou. 2025. [Unlocking exploration in RLVR: uncertainty-aware advantage shaping for deeper reasoning](#). *CoRR*, abs/2510.10649.
- Fangzhi Xu, Qiushi Sun, Kanzhi Cheng, Jun Liu, Yu Qiao, and Zhiyong Wu. 2025. [Interactive evolution: A neural-symbolic self-training framework for large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 12975–12993. Association for Computational Linguistics.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Cheng Yang, Chufan Shi, Siheng Li, Bo Shui, Yujiu Yang, and Wai Lam. 2025b. [LLM2: let large language models harness system 2 reasoning](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 2: Short Papers, Albuquerque, New Mexico, April 29 - May 4, 2025*, pages 168–177. Association for Computational Linguistics.
- Xiao-Wen Yang, Xuan-Yi Zhu, Wen-Da Wei, Dingchu Zhang, Jie-Jing Shao, Zhi Zhou, Lan-Zhe Guo, and Yufeng Li. 2025c. [Step back to leap forward: Self-backtracking for boosting reasoning of language models](#). *CoRR*, abs/2502.04404.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Zijun Yao, Yantao Liu, Yanxu Chen, Jianhui Chen, Junfeng Fang, Lei Hou, Juanzi Li, and Tat-Seng Chua. 2025. [Are reasoning models more prone to hallucination?](#) *CoRR*, abs/2505.23646.
- Haoran Ye, Xuning He, Vincent Arak, Haonan Dong, and Guojie Song. 2026. [Meta context engineering via agentic skill evolution](#). *Preprint*, arXiv:2601.21557.
- Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. 2023. [Frequency-domain mlps are more effective learners in time series forecasting](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Xixian Yong, Xiao Zhou, Yingying Zhang, Jinlin Li, Yefeng Zheng, and Xian Wu. 2025. [Think or not? exploring thinking efficiency in large reasoning models via an information-theoretic lens](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Dian Yu, Baolin Peng, Ye Tian, Linfeng Song, Haitao Mi, and Dong Yu. 2024. [Siam: Self-improving code-assisted mathematical reasoning of large language models](#). *CoRR*, abs/2408.15565.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gao-hong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. [DAPO: An open-source LLM reinforcement learning system at scale](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Haohan Yuan, Sukhwa Hong, and Haopeng Zhang. 2026. [Strucsum: Graph-structured reasoning for long document extractive summarization with llms](#). In *Findings of the Association for Computational Linguistics: EACL 2026, Rabat, Morocco, March 24-29, 2026*, Findings of ACL, pages 3708–3721. Association for Computational Linguistics.
- Haohan Yuan and Haopeng Zhang. 2025a. [Domainsum: A hierarchical benchmark for fine-grained domain shift in abstractive text summarization](#). In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, Findings of ACL, pages 2219–2231. Association for Computational Linguistics.

- Haohan Yuan and Haopeng Zhang. 2025b. [Understanding LLM reasoning for abstractive summarization](#). *CoRR*, abs/2512.03503.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. [Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model?](#) In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. [Star: Bootstrapping reasoning with reasoning](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025a. [Reasoning models know when they're right: Probing hidden states for self-verification](#). *CoRR*, abs/2504.05419.
- 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](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024b. [Accessing GPT-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b](#). *CoRR*, abs/2406.07394.
- Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025b. [Adaptthink: Reasoning models can learn when to think](#). *CoRR*, abs/2505.13417.
- Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. 2025c. [LightThinker: Thinking step-by-step compression](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13318–13339, Suzhou, China. Association for Computational Linguistics.
- Junyu Zhang, Runpei Dong, Han Wang, Xuying Ning, Haoran Geng, Peihao Li, Xialin He, Yutong Bai, Jitendra Malik, Saurabh Gupta, and Huan Zhang. 2025d. [AlphaOne: Reasoning models thinking slow and fast at test time](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11340–11365, Suzhou, China. Association for Computational Linguistics.
- Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Bqing Qi, and Bowen Zhou. 2025e. [OpenPRM: Building open-domain process-based reward models with preference trees](#). In *The Thirteenth International Conference on Learning Representations*.
- Shaowei Zhang and Deyi Xiong. 2025. [Backmath: Towards backward reasoning for solving math problems step by step](#). In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025 - Industry Track, Abu Dhabi, UAE, January 19-24, 2025*, pages 466–482. Association for Computational Linguistics.
- Xiaoyun Zhang, Jingqing Ruan, Xing Ma, Yawen Zhu, Haodong Zhao, Hao Li, Jiansong Chen, Ke Zeng, and Xunliang Cai. 2025f. [When to continue thinking: Adaptive thinking mode switching for efficient reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 5808–5828. Association for Computational Linguistics.
- Ziqi Zhao, Zhaochun Ren, Jiahong Zou, Liu Yang, Zhiwei Xu, Xuri Ge, Zhumin Chen, Xinyu Ma, Daiting Shi, Shuaiqiang Wang, Dawei Yin, and Xin Xin. 2026. [Reinforced efficient reasoning via semantically diverse exploration](#). *CoRR*, abs/2601.05053.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Keyang Zhong, Junlin Xie, Hefeng Wu, Haofeng Li, and Guanbin Li. 2026. [Collaborative multi-agent scripts generation for enhancing imperfect-information reasoning in murder mystery games](#). *Preprint*, arXiv:2604.11741.
- Yixiao Zhou, Ziyu Zhao, Dongzhou Cheng, Zhiliang Wu, Jie Gui, Yi Yang, Fei Wu, Yu Cheng, and Hehe Fan. 2025. [Dropping experts, recombining neurons: Retraining-free pruning for sparse mixture-of-experts llms](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 15169–15186. Association for Computational Linguistics.

A Notation

Table 3 summarizes the notations used throughout this paper.

B Algorithm Workflow

The algorithm framework of **NeuReasoner** is presented in Algorithm 1.

C Related Work

LLM Reasoning. According to (Chen et al., 2025a; An et al., 2025; Liu et al., 2026), LLM reasoning can be grouped into three families: (1) Deep Reasoning, (2) Feasible Reflection, and (3)

Notation	Description
L	The total number of layers in the Large Language Model (LLM).
l^*	The index of the target intermediate layer for analysis, set to $l^* = L/2$.
\mathcal{C}	The set of all neurons, including both FFN neurons and attention heads, within the target layer l^* .
y_t	The target token generated at time step t .
\mathbf{w}_{y_t}	The vector representation of the Language Model (LM) head corresponding to token y_t .
$\mathbf{h}_{dec}^{(l^*)}(t, c)$	The decomposed hidden state associated with neuron c at time step t in layer l^* .
$\phi(c, t)$	The attribution score quantifying the independent contribution of neuron c toward the logit of y_t .
T	The set of time steps defining the observation window, varying by failure mode level (intra, inter, or instance).
\mathcal{N}^*	The set of significant neurons identified as MoN via the intersection of top-k attribution scores.
\mathbf{a}_n	The sequence of scalar activation values for a neuron n over the time window T .
μ	The mean activation value of the sequence used for zero-centering.
$P(f)$	The power spectrum derived from the Discrete Fourier Transform (DFT) of the activation sequence.
$\tilde{P}(f)$	The normalized power distribution excluding the Direct Current (DC) component.
H	The normalized spectral entropy quantifying the dispersion of the power spectrum.
\mathcal{F}_{HF}	The set of high-frequency indices, defined as the latter half of the spectrum.
r_{HF}	High-frequency energy ratio, a key feature for detecting intra-step failure modes.
e	Total variation energy representing the overall intensity of fluctuations.
r_{dom}	Dominant frequency energy ratio, indicating the strength of periodic oscillations (inter-step level).
r_{LF}	Low-frequency energy ratio used to characterize instance-level collapse patterns.
ϕ_n	The concatenated input feature vector for neuron n (specific to level ℓ) used for MLP training.
\mathcal{D}_{raw}	The raw dataset used for reconstruction, consisting of input-output pairs (\mathbf{u}, \mathbf{v}) .
$\hat{\mathcal{D}}$	The reconstructed dataset containing injected failure modes and trigger mechanisms.
$\text{Seg}(\cdot)$	The segmentation operator that divides the output \mathbf{v} into step-level subsequences π .
ℓ	The level of failure mode, where $\ell \in \{\text{intra}, \text{inter}\}$.
$\Psi_\ell(\cdot)$	The rewriting operation that injects a specific failure pattern into a critical step.
$\langle v_\ell \rangle$	The special trigger token associated with failure level ℓ , serving as a control signal.
\mathcal{B}_ℓ	The diagnose-then-correct template containing the prompt \mathbf{p}_ℓ , diagnosis \mathbf{d}_ℓ , and correction \mathbf{c}_ℓ .
θ	The trainable parameters of the model optimized via SFT.
$\mathcal{I}(\mathbf{u}, \hat{\mathbf{v}})$	The set of token indices that contribute to the loss function (masking inputs and triggers).
x	The current decoding prefix sequence during inference.
τ	The length of the current decoding prefix x .

Table 3: Comprehensive summary of notations and definitions used in this paper.

Extensive Exploration. **❶ Deep Reasoning** emphasizes sustaining deep logical processing over long reasoning traces, and is often realized by improving the reasoning format (natural-language, structured/program-like, or latent-space reasoning) and/or by learning long reasoning behaviors from self-generated trajectories. Representative directions include natural-language CoT-style decomposition (e.g., CoT (Wei et al., 2022), Natural Program (Ling et al., 2023), CodeI/O (Li et al., 2025c), CoRT (Li et al., 2025a)), structured and symbolic/programmable reasoning (e.g., Brain (Chen et al., 2024b), SIaM (Yu et al., 2024), ENVISIONS (Xu et al., 2025), SKIntern (Liao et al., 2025), QuaSAR (Ranaldi et al., 2025), MCE (Ye et al., 2026)), and latent reasoning mechanisms (e.g., Coconut (Hao et al., 2024), Heima (Shen et al., 2025a), LTM (Kong et al., 2025)), as

well as self-learning and tree-search-based training signals (e.g., STaR (Zelikman et al., 2022), ReST (Gülçehre et al., 2023), PGTS (Li, 2025)). **❷ Feasible Reflection** equips models with iterative self-correction by generating feedback and then performing refinement over earlier reasoning states; typical instantiations range from prompt-based backtracking/refinement to SFT/RL-based (Dong et al., 2025; Lin et al., 2025; Li et al., 2025e; Wu et al., 2025b; Zhang et al., 2025f) reflective learning, with representative methods such as Self-Backtracking (Yang et al., 2025c), Refiner (Paul et al., 2024), BackMath (Zhang and Xiong, 2025), MCTSr (Zhang et al., 2024b), LLM2 (Yang et al., 2025b), ReARTeR (Sun et al., 2025) and ReST-MCTS* (Zhang et al., 2024a). **❸ Extensive Exploration** targets robustness under uncertainty by branching and searching over multiple candidate

Algorithm 1: Algorithm workflow of NeuReasoner

Input : Raw dataset \mathcal{D}_{raw} , LLM parameters θ , Failure Detection MLPs $\mathcal{M} = \{f_\ell\}_\ell$, Time window k

Output : Optimized LRM parameters θ^*

```
1 /* Phase 1: Pattern Prediction & Trigger Training */
2 for failure level  $\ell \in \{\text{intra}, \text{inter}, \text{inst}\}$  do
3   Construct feature vectors  $\phi^\ell$  via Fourier Transform; ▷ Sec. 3.1
4   Train lightweight MLP  $f_\ell$  to predict fluctuation patterns;
5 Initialize reconstructed dataset  $\hat{\mathcal{D}} \leftarrow \emptyset$ ;
6 for  $(\mathbf{u}, \mathbf{v}) \in \mathcal{D}_{\text{raw}}$  do
7   Segment output  $\mathbf{v}$  into steps  $\langle \pi_1, \dots, \pi_K \rangle$ ;
8   Identify critical steps  $\mathcal{J}$  for perturbation;
9   for  $j \in \mathcal{J}$  and level  $\ell$  do
10    Apply rewriting  $\tilde{\pi}_j^{(\ell)} = \Psi_\ell(\mathbf{u}, \pi_{<j}, j)$ ; ▷ Eq. 4
11    Insert trigger  $\langle v_\ell \rangle$  and diagnose-then-correct template  $\mathcal{B}_\ell$ ;
12    Construct  $\hat{\mathbf{v}}$  and add  $(\mathbf{u}, \hat{\mathbf{v}})$  to  $\hat{\mathcal{D}}$ ; ▷ Eq. 5
13 /* Supervised Fine-Tuning (SFT) */
14 Update  $\theta$  by maximizing likelihood on  $\hat{\mathcal{D}}$ :
15  $\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(\mathbf{u}, \hat{\mathbf{v}}) \sim \hat{\mathcal{D}}} \sum \log P_\theta(y_t | \mathbf{u}, y_{<t})$ ; ▷ Eq. 6
16 /* Phase 2: Inference with Online Monitoring */
17 Given input query  $x$ , current context  $C \leftarrow x$ ;
18 while generation not finished do
19   Update sliding window features  $\phi$  from recent tokens;
20   if  $\exists \ell, f_\ell(\phi)$  detects failure pattern then
21     /* Trigger Intervention */
22     Force next token  $y_{\text{next}} \leftarrow \langle v_\ell \rangle$ ;
23     Activate learned behavior  $\mathcal{B}_\ell$  (Diagnose & Correct);
24   else
25     Sample next token  $y_{\text{next}} \sim P_\theta(\cdot | C)$ ;
26   Append  $y_{\text{next}}$  to  $C$ ;
27 return Final generated reasoning path
```

reasoning trajectories (Liu et al., 2021, 2025a, 2023; Yuan et al., 2026; Yuan and Zhang, 2025a,b), spanning exploration scaling (sequential vs. parallel), internally learned exploration policies (e.g., STeCa (Wang et al., 2025a)), and externally scaffolded search/tool frameworks (e.g. ToT (Yao et al., 2023), CodeTree (Li et al., 2025b), Forest-of-Thought (Bi et al., 2025)). However, most existing works suffer from a lack of explainability, controllability, and unification. In contrast, NeuReasoner charts a promising course for future developments in this field.

Interpretability for LLMs. Interpretability techniques for LRMs span several fronts. ❶

Attribution-based methods assign credit to input features or intermediate steps, adapting token-level saliency to trace how each part of a prompt or chain-of-thought contributes to the model’s answer (Ferraro et al., 2025; Chang et al., 2025b; Hanna et al., 2025). ❷ **Probing techniques** analyze internal representations by training diagnostic classifiers or intervening on hidden states, revealing whether models encode latent reasoning variables (Conneau et al., 2018; Tenney et al., 2019; Hewitt and Manning, 2019; Zhang et al., 2025a). ❸ **Mechanistic interpretability** directly examines network weights and activations to reverse-engineer the model’s reasoning process, identifying sub-circuits responsible for specific reasoning skills (He et al.,

2026; Zhong et al., 2026; Xie et al., 2025; Li et al., 2026a; Zhou et al., 2025; Nie et al., 2026; Wang, 2026). However, there remains a lack of systematic and in-depth mechanistic analysis regarding why LRMs fail during reasoning trajectories. Our work effectively sheds light on these underlying mechanisms.

D Dataset

MATH. MATH (Hendrycks et al., 2021) is a competition-level mathematics benchmark containing 12,500 problems paired with step-by-step solutions, designed to evaluate multi-step mathematical reasoning beyond routine K–12 exercises. It covers seven subject areas (e.g., algebra, geometry, number theory) and is commonly used for answer exact-match evaluation on the final result.

MATH500. MATH500 (Lightman et al., 2023) is a curated subset of 500 representative problems drawn from the MATH benchmark and popularized as a lightweight yet challenging test set for LLM math reasoning. It is widely adopted for standardized comparisons across models.

GSM8K. GSM8K (Cobbe et al., 2021) is a dataset of 8.5K human-written grade-school arithmetic word problems, split into 7.5K training and 1K test instances. Problems typically require a short chain of elementary operations (multi-step arithmetic), and performance is usually measured by exact-match accuracy on the final answer.

AIME24. AIME24 (MAA Committees) is an olympiad-style math benchmark constructed from the 2024 American Invitational Mathematics Examination (AIME) I and II, totaling 30 problems. Each question expects an integer answer in $[0,999]$, enabling reliable automatic evaluation via exact match while still demanding substantial symbolic and combinatorial reasoning.

AIME25. AIME25 (MAA Committees) similarly aggregates 30 problems from the 2025 AIME I and II into an evaluation set with integer answers in $[0,999]$. Owing to its “fresh” annual release and contest difficulty, AIME25 is frequently used as a stringent test of advanced mathematical reasoning and generalization under minimal ambiguity in answer format.

GPQA-Diamond. GPQA (Rein et al., 2024) is a graduate-level, “Google-proof” multiple-choice QA benchmark authored and validated by domain

experts across biology, chemistry, and physics; the Diamond split is a higher-quality, more challenging subset containing 198 questions. It is commonly used to probe scientific reasoning under expert-level knowledge demands, with accuracy computed over 4-option multiple-choice answers.

LiveCodeBench. LiveCodeBench (Jain et al., 2025) is a continuously updated coding benchmark explicitly designed to mitigate test-set contamination by collecting newly released competitive-programming problems over time. Beyond code generation, it emphasizes holistic coding abilities (e.g., self-repair, execution, test-output prediction) and provides time-stamped releases (e.g., hundreds of problems spanning May 2023 onward) for reproducible evaluation.

Bespoke-Stratos-17k. Bespoke-Stratos-17k (Labs, 2025) is a reasoning distillation dataset (17K examples) consisting of questions paired with reasoning traces and final answers, created by replicating and improving the Berkeley Sky-T1 pipeline using distillation data from DeepSeek-R1. It is used as supervised fine-tuning data to induce long-form, explicit reasoning behaviors across domains including math and coding.

E Technical Details

E.1 Data Preprocessing

Following LightThinker (Zhang et al., 2025c), we utilize BS-17k (Labs, 2025) as the training dataset for both MLPs and SFT. To prevent potential data leakage, we explicitly exclude samples overlapping with our evaluation benchmarks (MATH500, GSM8K, AIME24, AIME25, GPQA-Diamond, and LiveCodebench) via text matching.

E.2 Elaboration on MoN

As discussed in the Limitations, **NeuReasoner** is not currently a fully automated, end-to-end framework, primarily due to variations in layer depth and architecture across different backbones. However, we emphasize that while the specific constituent neurons of MoN vary, they adhere to universal patterns established in Section 2. Specifically: ① intra-step and instance-level MoN are predominantly located in FFNs, whereas inter-step MoN reside in attention heads; and ② intra-step failures exhibit significant fluctuation amplitudes, inter-step failures display periodicity, and instance-level failures are characterized by sudden collapse. In prac-

Algorithm 2: Approximate $O(1)$ Update of Variable-Length Sliding-Window Fourier Features

Input : Level $\ell \in \{\text{intra}, \text{inter}\}$; monitored channels $n \in \{1, \dots, N_\ell\}$; new time index t with activations $\{a_{n,t}^{(\ell)}\}$; pop count $r_t \geq 0$; fixed probes $\Omega = \{\omega_k\}_{k=1}^K$ (K is a small constant); $\varepsilon > 0$.

Output : Features $\{\phi_n^\ell(t)\}_{n=1}^{N_\ell}$, where $\phi_n^{\text{intra}}(t) = [r_{\text{HF}}^{(n)}(t), H^{(n)}(t), e^{(n)}(t)]$ and $\phi_n^{\text{inter}}(t) = [r_{\text{dom}}^{(n)}(t), H^{(n)}(t)]$.

```
1 /* State (persistent across decoding steps). Use  $y_{n,i} = |a_{n,i}^{(\ell)}|$ . */
   Maintain window left index  $s$ , length  $L$ ; FIFO queue  $\mathcal{Q}$  storing vectors  $\mathbf{y}_i = [y_{1,i}, \dots, y_{N_\ell,i}]$ ;
   vectors  $\mathbf{U} = \sum_{i=s}^t \mathbf{y}_i$ ,  $\mathbf{V} = \sum_{i=s}^t \mathbf{y}_i \odot \mathbf{y}_i$ ; matrix  $\mathbf{A} \in \mathbb{C}^{N_\ell \times K}$  with  $\mathbf{A}_{:,k} = \sum_{i=s}^t \mathbf{y}_i q_k^i$ .
   Maintain phases  $\alpha_k = q_k^t$ ,  $\beta_k = q_k^s$ ,  $\rho_k = q_k^L$  and constants  $q_k = e^{-i\omega_k}$ ,  $\eta_k = (1 - q_k)^{-1}$ .

2 /* 1) Push one token (always). */
3  $\mathbf{y}_t \leftarrow [|a_{1,t}^{(\ell)}|, \dots, |a_{N_\ell,t}^{(\ell)}|]$ ; Enqueue  $\mathbf{y}_t$  into  $\mathcal{Q}$ ;  $L \leftarrow L + 1$ ;
4  $\mathbf{U} \leftarrow \mathbf{U} + \mathbf{y}_t$ ;  $\mathbf{V} \leftarrow \mathbf{V} + \mathbf{y}_t \odot \mathbf{y}_t$ ;
5 for  $k \leftarrow 1$  to  $K$  do
6    $\alpha_k \leftarrow \alpha_k q_k$ ;  $\rho_k \leftarrow \rho_k q_k$ ;  $\mathbf{A}_{:,k} \leftarrow \mathbf{A}_{:,k} + \mathbf{y}_t \alpha_k$ ;

7 /* 2) Pop  $r_t$  oldest tokens (optional, can be multiple). */
8 while  $r_t > 0$  do
9   Dequeue  $\mathbf{y}_s$  from  $\mathcal{Q}$ ;  $L \leftarrow L - 1$ ;
10   $\mathbf{U} \leftarrow \mathbf{U} - \mathbf{y}_s$ ;  $\mathbf{V} \leftarrow \mathbf{V} - \mathbf{y}_s \odot \mathbf{y}_s$ ;
11  for  $k \leftarrow 1$  to  $K$  do
12     $\mathbf{A}_{:,k} \leftarrow \mathbf{A}_{:,k} - \mathbf{y}_s \beta_k$ ;  $\beta_k \leftarrow \beta_k q_k$ ;  $\rho_k \leftarrow \rho_k q_k^{-1}$ ;
13   $s \leftarrow s + 1$ ;  $r_t \leftarrow r_t - 1$ ;

14 /* 3) Compute features (no traversal over window). */
15 for  $k \leftarrow 1$  to  $K$  do
16    $B_k \leftarrow \beta_k (1 - \rho_k) \eta_k$ ; //  $B_k = \sum_{i=s}^t q_k^i$ 
17 for  $n \leftarrow 1$  to  $N_\ell$  do
18    $\mu_n \leftarrow U_n / L$ ;
19   for  $k \leftarrow 1$  to  $K$  do
20      $S_{n,k} \leftarrow A_{n,k} - \mu_n B_k$ ;  $P_{n,k} \leftarrow |S_{n,k}|^2$ ;
21      $Z_n \leftarrow \sum_{k=1}^K P_{n,k} + \varepsilon$ ;  $\tilde{P}^{(n)}(k) \leftarrow P_{n,k} / Z_n$ ;
22      $H^{(n)}(t) \leftarrow -\frac{1}{\log K} \sum_{k=1}^K \tilde{P}^{(n)}(k) \log \tilde{P}^{(n)}(k)$ ;
23   if  $\ell = \text{intra}$  then
24      $r_{\text{HF}}^{(n)}(t) \leftarrow \sum_{k > \lfloor K/2 \rfloor} \tilde{P}^{(n)}(k)$ ;  $E^{(n)}(t) \leftarrow V_n - \frac{U_n^2}{L}$ ;  $e^{(n)}(t) \leftarrow \log(E^{(n)}(t) + \varepsilon)$ ;
25      $\phi_n^{\text{intra}}(t) \leftarrow [r_{\text{HF}}^{(n)}(t), H^{(n)}(t), e^{(n)}(t)]$ ;
26   else
27      $r_{\text{dom}}^{(n)}(t) \leftarrow \max_{1 \leq k \leq K} \tilde{P}^{(n)}(k)$ ;
28      $\phi_n^{\text{inter}}(t) \leftarrow [r_{\text{dom}}^{(n)}(t), H^{(n)}(t)]$ ;

29 return  $\{\phi_n^\ell(t)\}_{n=1}^{N_\ell}$ 
```

tice, this necessitates independent MoN extraction and MLP training for each backbone. Fortunately, these processes remain lightweight. For instance, training the MLPs requires labeling only ~ 20 reasoning trajectories, which yields approximately

$\sim 5,000$ token-level training samples. Extending this framework to achieve cross-backbone alignment represents a pivotal direction for our future work. And we also plan to explore the application of our method to multimodal reasoning (Jiang et al.,

Failure Mode	Accuracy	Recall
MLP _{intra}	0.871 ± 0.010	0.848 ± 0.012
MLP _{inter}	0.862 ± 0.011	0.936 ± 0.008
MLP _{inst}	0.944 ± 0.006	0.951 ± 0.007

Table 4: Experimental results across different MLPs.

2026, 2024; Xiao et al., 2025, 2026a,b; Li et al., 2026b,c; Chen et al., 2025b, 2026; Liu et al., 2024, 2025b; Li et al., 2025d; Zhao et al., 2026).

E.3 MLPs Training

Datasets. We utilize the BS-17k dataset for MLP training, randomly partitioned into training and test sets at an 8 : 2 ratio. To maximize the diversity of erroneous reasoning paths, we perform repeated sampling using DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Llama-8B to collect reasoning trajectories with associated neuron activations. We then filter for failure instances and employ an LLM-as-a-Judge approach to annotate the specific token segments corresponding to these failures.

Training Details. We adopt a unified architecture for MLPs across all levels: a simple three-layer fully connected network, with input dimensions determined by the respective input features. Training employs the GELU activation function, Dropout regularization, binary cross-entropy loss, and the AdamW optimizer.

Results. Table 4 presents the experimental results on the test set, reporting accuracy and recall. The results demonstrate the excellent performance of our trained MLPs.

Feature Update. Algorithm 2 presents our optimized sliding-window feature update procedure, achieving $\mathcal{O}(1)$ time complexity.

E.4 Fine-tuning

We employ full-parameter fine-tuning for 1 epoch, utilizing a cosine warmup schedule with a ratio of 0.05. The learning rate is set to $1e-5$, and the batch size is 64 (Chang et al., 2024, 2026, 2025a).

E.5 Baseline Settings

For all baselines, we strictly adhere to the settings detailed in their respective original papers. To ensure fair comparison, we standardize generation hyperparameters across all methods: temperature is set to 0.6, top- p to 0.95, and the maximum token limit to 16,000. We report the average Pass@1

accuracy over five independent runs. Additionally, for RL-based baselines, experiments are conducted on A100 GPUs utilizing bf16 mixed precision and DeepSpeed ZeRO-3.

F Additional Results

F.1 AIME24 Results

Table 5 presents additional experimental results on the AIME24 dataset.

F.2 Runtime Comparison

Table 6 presents the inference runtime comparison between **NeuReasoner** and the Vanilla model, measured on A100 GPU. It is evident that while **NeuReasoner** introduces marginal overhead, it remains within an acceptable range.

F.3 Hyperparameter Sensitivity

We present the results of the hyperparameter sensitivity experiments in Table 7.

G Prompt & Example Data

We provide all prompts utilized in this paper, including those for dataset reconstruction and failure mode detection, alongside examples of reconstructed data.

INTRA-STEP LEVEL RECONSTRUCTION

```

### SYSTEM
You are a data-construction
assistant for SFT. Your task is
to generate reasoning
trajectories containing a
specific "Intra-step Level"
failure and diagnose-then-
correct pattern.

### INPUT FORMAT
1. <PROBLEM> ... </PROBLEM>
2. <REFERENCE_REASONING> ... </
REFERENCE_REASONING>
3. <CONTROL>
error_length: {short|medium|
long}
error_type: {dropped_case|
invalid_division|sign_error|
algebra_simplification_error|
mistaken_assumption|
domain_violation}
style: LRM_natural_first_person
</CONTROL>

### INSTRUCTIONS
Produce a single reasoning trace
following these steps strictly:
1. Context: Copy the first 2-4
steps of the <
REFERENCE_REASONING> verbatim.

```

Method	R1-Qwen-7B		R1-Qwen-32B		Qwen3-8B-thinking		Qwen3-32B-thinking		R1-Llama-8B		R1-Llama-70B	
	Pass@1	Token	Pass@1	Token	Pass@1	Token	Pass@1	Token	Pass@1	Token	Pass@1	Token
Vanilla	54.4	10438	70.0	7873	70.0	11125	<u>77.8</u>	10677	45.6	10798.9	68.9	7766
DAST	55.6	7258	70.0	5802	67.8	<u>5964</u>	<u>77.8</u>	5981	45.6	8246	67.8	5115
Think or Not	52.2	4341	67.8	3993	68.9	5387	76.7	6173	44.4	6761	71.1	4005
AlphaOne	55.6	8224	72.2	8210	<u>73.3</u>	8343	78.9	8007	<u>47.8</u>	8339	72.2	7873
RL + LP	52.2	5693	71.1	5492	71.1	6986	76.7	6238	45.6	5333	66.7	5304
GRPO	<u>56.7</u>	11673	<u>73.3</u>	8389	72.2	10931	78.9	11934	44.4	11312	72.2	8109
S-GRPO	54.4	5094	70.0	4906	<u>73.3</u>	6771	<u>77.8</u>	6040	45.6	4809	70.0	<u>5002</u>
DAPO	57.8	11908	<u>73.3</u>	8817	71.1	11781	78.9	12038	46.7	12079	<u>73.3</u>	8589
NeuReasoner	57.8	4997	74.4	4456	74.4	6609	78.9	<u>6029</u>	50.0	5116	74.4	5237
Δ	$\uparrow+3.4$	$\downarrow52.1\%$	$\uparrow+4.4$	$\downarrow43.4\%$	$\uparrow+4.4$	$\downarrow40.6\%$	$\uparrow+1.1$	$\downarrow43.5\%$	$\uparrow+4.4$	$\downarrow52.6\%$	$\uparrow+5.5$	$\downarrow32.6\%$

Table 5: Additional results on the AIME24 dataset. Best results are **bold**, runners-up underlined.

Model	Method	AIME24	AIME25	MATH500	GSM8K	GPQA-Diamond	LiveCodeBench
DeepSeek-R1-Qwen-7B	Vanilla	02:13	02:12	04:11	02:26	04:52	24:20
	NeuReasoner	02:24	02:23	04:29	02:38	05:13	25:51
DeepSeek-R1-Qwen-32B	Vanilla	04:00	04:10	07:06	01:45	07:29	36:36
	NeuReasoner	04:17	04:28	07:35	01:54	07:59	38:51
Qwen3-8B-thinking	Vanilla	02:26	02:28	06:24	05:47	04:12	24:56
	NeuReasoner	02:38	02:40	06:50	06:11	04:30	26:29
Qwen3-32B-thinking	Vanilla	04:25	04:34	10:04	08:09	06:21	44:15
	NeuReasoner	04:44	04:53	10:43	08:41	06:47	46:57
DeepSeek-R1-Llama-8B	Vanilla	02:10	02:06	04:49	02:38	04:59	25:21
	NeuReasoner	02:21	02:17	05:09	02:50	05:20	26:55
DeepSeek-R1-Llama-70B	Vanilla	06:41	06:37	10:47	03:42	10:25	57:56
	NeuReasoner	07:08	07:04	11:29	03:58	11:06	61:28

Table 6: Inference efficiency comparison (Time in MM:SS).

Hyper	Value	Pass@1	Token
k_{intra}	2	96.0	1360
	4	96.3	1318
	8	96.2	1335
k_{inter}	2	96.1	1345
	4	96.3	1318
	8	96.4	1328
K_{inst}	2	95.9	1180
	4	96.3	1318
	8	96.5	1575

Table 7: Hyperparameter sensitivity on MATH500 with DeepSeek-R1-Distill-Qwen-32B.

- Error Injection:** Identify the next critical step and rewrite it to be plausibly *wrong* based on the ``error_type``. Do not introduce multiple errors.
- Trigger Insertion:** Immediately after the wrong step, insert this exact block:

```
<INTRA>
[FAILURE_INTRA] A key reasoning error surfaced a few steps back. I should identify the mistaken span, briefly analyze it, and correct it before
```

- ```
continuing.
- **error span**: "<quote the wrong text span>"
- **analysis**: <1-3 sentences explaining why it is wrong>
```
- Correct & Continue:** After the block, "back up" logically, apply the correction, and continue reasoning to the correct final answer.
  - Format:** Use natural first-person reasoning. Return ONLY the reconstructed reasoning trace (no meta commentary, no JSON, no extra headers).

### INTER-STEP LEVEL RECONSTRUCTION

```
SYSTEM
You are a data-construction assistant for SFT. Your task is to generate reasoning trajectories containing an "Inter-step Level" stagnation-and-repair pattern.
```

```
INPUT FORMAT
1. <PROBLEM> ... </PROBLEM>
2. <REFERENCE_REASONING> ... </REFERENCE_REASONING>
```

```

3. <CONTROL>
loop_length_paragraphs: {3|4|5}
loop_theme: {mod_checks|bounds|
symmetry_observations|
equivalent_reformulations}
style: LRM_natural_first_person
</CONTROL>

INSTRUCTIONS
Produce a single reasoning trace
following these steps strictly:
1. Context: Copy the first 2-4
steps of the <REFERENCE_REASONING> verbatim.
2. Loop Generation: Create a
realistic stagnation loop of `
loop_length_paragraphs`.
- The model must oscillate
between strategies related to `
loop_theme`.
- It should sound logical but
fail to make decisive progress
(spinning wheels).
- Do not generate nonsense;
simulate a model trying but
failing to break through.
3. Trigger Insertion:
Immediately after the loop,
insert this exact block:
<INTER>
[FAILURE_INTER] I'm stuck in an
inter-step loop: I keep
revisiting the same near
equivalent checks and switching
between them whenever one
stalls, without any progress.
Let me summarize the loop, then
pivot to a genuinely different
route.
- past attempts summary:
<2-4 bullet-worthy clauses
summarizing the repeated
approaches>
- problem-grounded analysis
: <1-3 sentences explaining
why those moves aren't breaking
the core constraint>
4. Pivot & Finish: Define a
pivot plan (1-2 sentences
naming a different framework
and the next concrete step).
Insert this plan into the block
above, then immediately
execute it and continue
reasoning to the correct final
answer.
5. Format: Use natural first-
person reasoning. Return ONLY
the reconstructed reasoning
trace (no meta commentary, no
JSON, no extra headers).

```

### INTRA-STEP LEVEL FAILURE DETECTION

```

SYSTEM
You are an expert Logic Verifier.
Your goal is to strictly

```

```

evaluate a reasoning trace for
Local Validity.

```

You will be given a <PROBLEM> and a <REASONING\_TRACE>. You must verify the trace step-by-step.

```

DETECTION CRITERIA
Flag a step as an [ERROR] if it
contains any of the following "
Intra-step" failures:
1. Calculation/Algebra: Sign
errors, invalid simplification,
arithmetic mistakes (e.g.,
2+2=5).
2. Domain Violation: Dividing
by zero, taking the log of a
negative, applying a theorem
outside its valid conditions.
3. Logic Non-sequitur: The
conclusion of step N does not
logically follow from step N-1.
4. Hallucination: Inventing
constraints or values not
present in the problem context.
5. Dropped Case: Arbitrarily
narrowing the scope (e.g.,
assuming x is positive without
proof).

```

```

CRITICAL NOTE:
- Ignore "inefficient" steps or "
circular" reasoning (that is a
separate check).
- Focus ONLY on whether the
specific statement is *
factually* or *mathematically*
false.

```

```

OUTPUT FORMAT
Return valid JSON only.

```

```

{
 "has_error": boolean,
 "first_error_step": string |
 null, // Quote the specific
 sentence/equation
 "error_type": "calculation" | "
 logic_non_sequitur" | "
 domain_violation" | "
 hallucination" | "dropped_case"
 | "none",
 "explanation": "Brief analysis
 of why this step is invalid
 given the prior context."
}

```

```

INPUT
<PROBLEM>
{{PROBLEM_TEXT}}
</PROBLEM>

```

```

<REASONING_TRACE>
{{MODEL_OUTPUT}}
</REASONING_TRACE>

```

## INTER-STEP LEVEL FAILURE DETECTION

```
SYSTEM
You are an expert Metacognitive
Strategy Evaluator. Your goal
is to analyze the Global
Information Flow of a
reasoning trace to detect
Stagnation or Loops. You will
be given a <PROBLEM> and a <
REASONING_TRACE>. Do not check
for minor calculation errors.
Instead, check if the reasoning
is "spinning its wheels."
```

```
DETECTION CRITERIA
Flag the trace as [STAGNANT] if it
exhibits these "Inter-step"
patterns for 3+ consecutive
paragraphs/steps:
```

1. **Equivalent Reformulations**:  
Rewriting the same equation in  
different forms without  
isolating new variables (e.g.,  
 $x=5-y \rightarrow y=5-x \rightarrow x+y=5$ ).
2. **Strategy Oscillation**:  
Switching back and forth  
between two approaches (e.g.,  
Modular Arithmetic  $\rightarrow$  Bounds  $\rightarrow$   
Modular Arithmetic) without  
ruling anything out.
3. **Empty Verbosefication**: Long  
explanations that restate the  
goal or definitions without  
deriving new data.
4. **Repetitive Checks**: Testing  
values or cases that were  
already implicitly or  
explicitly handled.

```
CRITICAL NOTE:
- A long derivation is NOT a loop
if it is making progress
towards a solution.
- A "stagnation" means the *
Information Entropy* is not
decreasing (the search space
isn't shrinking).
```

```
OUTPUT FORMAT
Return valid JSON only.
```

```
{
 "is_stagnant": boolean,
 "loop_detected_at": "Approximate
start of the loop (e.g., '
Paragraph 4')",
 "loop_theme": "modular_checks" |
"bounds_restatement" | "
equation_shuffling" | "none",
 "severity": "low" | "medium" | "
high" | "none",
 "explanation": "Brief analysis
of the repetitive pattern
observed."
}
```

```
INPUT
<PROBLEM>
```

```
{{PROBLEM_TEXT}}
</PROBLEM>

<REASONING_TRACE>
{{MODEL_OUTPUT}}
</REASONING_TRACE>
```

## EXAMPLE OF INTRA-STEP LEVEL DATA

```
[Question]:
How many positive integers n
satisfy that $n^2 + 2n + 2017$ is
a perfect square?
```

```
[Response]:
Let $n^2 + 2n + 2017 = m^2$ for some
integer m.
```

```
Complete the square on n: $n^2 + 2n
+ 2017 = (n+1)^2 + 2016$, so we
need
 $m^2 = (n+1)^2 + 2016$.
```

```
Rearrange as a difference of
squares:
 $m^2 - (n+1)^2 = 2016$
 $(m-(n+1))(m+(n+1)) = 2016$.
```

```
Let $u = m-(n+1)$ and $v = m+(n+1)$.
Then u and v are positive
integers with $uv = 2016$ and $v >
u$, and we can recover
 $n+1 = (v-u)/2$.
```

```
At first glance it seems each
factor pair (u,v) with $uv=2016$
and $u < v$ gives a valid n, so the
number of solutions should
just be the number of positive
factor pairs of 2016, which is
 $d(2016)/2 = 36/2 = 18$.
```

```
<INTRA>
[FAILURE_INTRA] A key reasoning
error occurred in the counting
step; I need to pinpoint it and
fix it before continuing.
- error span: "each factor
pair (u,v) with $uv=2016$ and $u < v$
gives a valid n ... so ... $d
(2016)/2 = 18$."
- analysis: The recovery
formula $n+1 = (v-u)/2$ requires
(v-u) to be even, i.e., u and v
must have the same parity.
This was not checked. Since $uv
=2016$ is even, u and v cannot
both be odd; the only way to
have the same parity is that
both are even. Counting all
factor pairs overcounts pairs
with opposite parity, which do
not yield an integer n.
```

```
Back up and apply the parity
constraint properly. Because $n
+1 = (v-u)/2$ must be an integer
```

,  $u$  and  $v$  must have the same parity. With  $uv=2016$  even, "same parity" forces  $u$  and  $v$  to both be even.

So write  $u = 2u'$  and  $v = 2v'$ . Then  $uv = 2016$  becomes  $4u'v' = 2016$ , hence  $u'v' = 504$ , and still  $v > u$  corresponds to  $v' > u'$ .

Now each factor pair  $(u', v')$  with  $u'v' = 504$  and  $u' < v'$  gives  $n+1 = (v-u)/2 = (2v' - 2u')/2 = v' - u'$ , so  $n = (v' - u') - 1$ .

We need  $n > 0$ , i.e.  $v' - u' > 1$ . The excluded case  $v' - u' = 1$  would require  $u'(u'+1) = 504$ , but 504 is not a product of consecutive integers (the discriminant  $1+4 \cdot 504 = 2017$  is not a perfect square), so every valid factor pair with  $u' < v'$  gives  $n > 0$ .

Therefore the number of valid  $n$  equals the number of factor pairs  $u' < v'$  of 504.

Factor  $504 = 2^3 \cdot 3^2 \cdot 7$ , so the number of divisors is  $(3+1)(2+1)(1+1) = 4 \cdot 3 \cdot 2 = 24$ . Since 504 is not a perfect square, the number of unordered factor pairs is  $24/2 = 12$ , and that equals the number of pairs with  $u' < v'$ .

Hence there are 12 positive integers  $n$  satisfying the condition.

Final Answer: 12.

### EXAMPLE OF INTER-STEP LEVEL DATA

[Question]:

How many ordered pairs of integers  $(x, y)$  satisfy  $x^2 + xy + y^2 = 91$ ?

[Response]:

This is a positive definite quadratic form, so there will only be finitely many integer solutions. I'll start by bounding the size.

Since  $x^2 + xy + y^2 \geq (1/2)(x^2 + y^2)$  (because the smallest eigenvalue of the symmetric form is  $1/2$ ), we get  $(1/2)(x^2 + y^2) \leq 91$

so  $x^2 + y^2 \leq 182$ , hence  $|x|, |y| \leq 13$ .

A direct search in that box is feasible in principle, but I'd like a cleaner counting method.

Treat the equation as a quadratic in  $x$ :

$$x^2 + yx + (y^2 - 91) = 0.$$

For  $x$  to be an integer, the discriminant must be a perfect square:

$$\Delta = y^2 - 4(y^2 - 91) = 364 - 3y^2.$$

So I need  $364 - 3y^2$  to be a nonnegative square. That already bounds  $y$ :

$$3y^2 \leq 364 \Rightarrow |y| \leq 11.$$

At this point I could try listing  $y = -11, \dots, 11$  and checking which values make  $\Delta$  a square, but I'm also tempted to see if modular arithmetic can prune first.

Modulo 3,  $x^2 + xy + y^2 \equiv 91 \equiv 1 \pmod{3}$ . The expression can be 0 or 1 mod 3 depending on  $(x, y)$ , so that's not restrictive. Mod 4 doesn't help much either since squares are 0 or 1 and  $xy$  mixes parities.

Maybe I should exploit the discriminant form:  $\Delta = 364 - 3y^2$ . If I set  $\Delta = k^2$ , then  $3y^2 + k^2 = 364$ . That resembles a sum of squares condition, but I'm not sure how to count it cleanly without still enumerating  $y$ .

I can also swap roles: solving for  $y$  gives the same discriminant condition, so it doesn't change the nature of the search.

Let me try more modular filtering: since  $\Delta$  is a square,  $\Delta \equiv 0$  or  $1 \pmod{4}$ . But  $364 - 3y^2 \equiv 0 - 3y^2 \pmod{4} \equiv -3y^2 \pmod{4}$ . If  $y$  is even,  $y^2 \equiv 0$  so  $\Delta \equiv 0$  (ok). If  $y$  is odd,  $y^2 \equiv 1$  so  $\Delta \equiv -3 \equiv 1 \pmod{4}$  (ok). So parity doesn't prune.

Trying mod 8: if  $y$  is odd,  $3y^2 \equiv 3 \pmod{8}$ , so  $\Delta \equiv 364 - 3 \equiv 1 \pmod{8}$ , which is a square residue. If  $y$  even,  $\Delta \equiv 4 \pmod{8}$ , also a square residue. So again permissive.

I'm going in circles: each "check" restates the same discriminant condition or gives permissive residue constraints. The cleanest might just be to enumerate  $y$  and count  $x$  from the quadratic formula, but I'd like a method that feels more structural.

<INTER>

[FAILURE\_INTER] I'm stuck in an inter-step loop: I keep revisiting the same near equivalent checks and switching between them whenever one stalls, without any progress. Let me summarize the loop, then pivot to a genuinely different route.

- **\*\*past attempts summary:\*\*** I bounded  $|x|, |y|$  and rewrote as a quadratic in  $x$  with discriminant  $\Delta = 364 - 3y^2$ ; then I repeatedly tried modular residue checks (mod 3/4/8) and symmetry swaps, but each move either restated the same discriminant constraint or stayed too permissive to reduce the search.
- **\*\*problem-grounded analysis:\*\*** The form  $x^2 + xy + y^2$  has a standard linear change of variables that diagonalizes it:  $4(x^2 + xy + y^2) = (2x + y)^2 + 3y^2$ . Converting the problem to  $u^2 + 3v^2 = 364$  (with a simple parity condition) turns it into a clean "square checking" enumeration over a much smaller set with no looping.

I'll take that diagonalization route and keep it consistent.

Start from  $x^2 + xy + y^2 = 91$  and multiply by 4:  
 $4x^2 + 4xy + 4y^2 = 364$ .

Notice  $(2x + y)^2 = 4x^2 + 4xy + y^2$ ,  
 so  
 $4x^2 + 4xy + 4y^2 = (2x + y)^2 + 3y^2$ .

Let  $u = 2x + y$  and  $v = y$ . Then the equation becomes  
 $u^2 + 3v^2 = 364$ .

Also, since  $x = (u - v)/2$ , we need  $u - v$  to be even (equivalently  $u \equiv v \pmod{2}$ ) so that  $x$  is an integer.

Now bound  $v$ :  
 $3v^2 \leq 364 \Rightarrow |v| \leq 11$ .

I can now check  $v = -11, \dots, 11$  and see when  $364 - 3v^2$  is a perfect square  $u^2$ , while also ensuring  $u \equiv v \pmod{2}$ .

Compute:

- $v = +/-11$ :  $364 - 3 \cdot 121 = 364 - 363 = 1 \Rightarrow u = +/-1$  (odd; matches  $v$  odd).
- $v = +/-10$ :  $364 - 3 \cdot 100 = 64 \Rightarrow u = +/-8$  (even; matches  $v$  even).
- $v = +/-9$ :  $364 - 3 \cdot 81 = 121 \Rightarrow u = +/-11$  (odd; matches  $v$  odd).
- $v = +/-6$ :  $364 - 3 \cdot 36 = 256 \Rightarrow u = +/-16$  (even; matches  $v$  even).
- $v = +/-5$ :  $364 - 3 \cdot 25 = 289 \Rightarrow u = +/-17$  (odd; matches  $v$  odd).
- $v = +/-1$ :  $364 - 3 \cdot 1 = 361 \Rightarrow u = +/-19$  (odd; matches  $v$  odd).

All other  $v$  in  $-11 \dots 11$  give non-squares, and in every listed case the parity condition  $u \equiv v \pmod{2}$  is satisfied automatically.

For each absolute value  $|v|$  in  $\{1, 5, 6, 9, 10, 11\}$ , there are:

- 2 choices for sign of  $v$ ,
  - 2 choices for sign of  $u$ ,
- giving 4 solutions  $(u, v)$ .

That's  $6 \cdot 4 = 24$  solutions  $(u, v)$ , and each corresponds to a unique integer pair  $(x, y)$  via  $y = v, x = (u - v)/2$ .

Therefore, the number of ordered integer pairs  $(x, y)$  is 24.

Final Answer: 24