INTERPRETING REASONING OF LARGE LANGUAGE MODELS VIA PARTIAL INFORMATION DECOMPOSITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large reasoning models (LRMs) have achieved substantial improvements in solving complex mathematical problems, but often produce lengthy, repetitive, or erroneous reasoning trajectories. In this work, we introduce a new interpretability framework SLIDER to evaluate the quality of the reasoning process. SLIDER leverages an emerging body of work from information theory called Partial Information Decomposition to disentangle the information about the final answer between two consecutive reasoning steps into non-negative components: unique information (in preceding steps or current step), redundant information, and synergistic information. Building on this decomposition, we propose the *Step-wise Repetitive Reasoning Index (Step-RRI)*, a theoretically grounded measure that assesses whether the answer-relevant information in the current step S_i is predominantly redundant with the past steps $S_{<i}$, relative to its unique and synergistic contributions. To evaluate the effectiveness of Step-RRI in detecting repetitiveness, we apply SLIDER to the redundancy class of the PRMBench dataset where Step-RRI improves step-level redundancy identification accuracy by over 10 points compared to embedding-similarity and information-gain baselines. Next, we define *Trajectory-RRI*, an aggregate measure of repetitiveness for an individual reasoning trajectory. To demonstrate its practical relevance, we show that average Trajectory-RRI strongly correlates with actual reasoning length across QwQ-32B, DeepSeek-R1-Distill-Qwen-32B, and GPT-4.1, motivating its use as a signal for improving reasoning efficiency. Finally, we introduce *Trajectory-RRI-guided data selection for fine-tuning*, demonstrating that selecting training data based on Trajectory-RRI can improve a fine-tuned model's reasoning efficiency while largely preserving its task performance. Our results establish RRI measures as both a step-level diagnostic of repetitive reasoning and a practical signal for improving reasoning efficiency.

1 INTRODUCTION

The emergence of large reasoning models (LRMs), such as OpenAI's o1 (OpenAI., 2024), DeepSeek-R1 (Guo et al., 2025), and QwQ-32B (Team, 2025), has marked a significant leap in complex problem solving. These models typically engage in extensive reasoning or thinking before producing a final answer, often generating long chain-of-thought (CoT) reasoning trajectories (Wei et al., 2022). However, such reasoning can be excessively verbose and repetitive, frequently revisiting information that has already been established in earlier steps. This redundant reasoning increases computational cost without necessarily improving the correctness of the final answer. Moreover, as reasoning trajectories become longer and more convoluted, they become increasingly difficult for humans to follow and evaluate, making it challenging to determine whether individual steps are correct, informative, or meaningfully contribute to the final answer. *Thus, there is a pressing need for automated methods to evaluate the quality of individual reasoning steps without human supervision.*

To address this challenge, we propose SLIDER (**SL**iding Window **I**nformation **D**ecomposition **E**xplainer for **R**easoning), a unified interpretability framework that leverages Partial Information Decomposition (PID) to analyze the quality of individual reasoning steps. Building on PID, we introduce a novel metric *Step-wise Repetitive Reasoning Index (Step-RRI)*, which uses the relative contributions of redundant, unique, and synergistic answer-relevant information to characterize whether an individual reasoning step is repetitive with respect to the preceding reasoning. Fig. 1 illustrates an example of reasoning steps exhibiting different degrees of repetitiveness, and hence varying Step-RRI values.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To quantify reasoning repetitiveness at the trajectory level, we further define *Trajectory-RRI* by aggregating repetitive behavior across the individual steps of a reasoning trajectory. Together, these measures characterize reasoning repetitiveness at the step and trajectory levels without requiring reference reasoning traces.

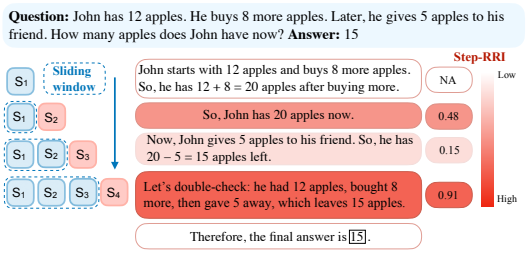

Figure 1: *Example illustrating the role of* SLIDER *in identifying repetitiveness.* The sliding window compares each reasoning step with the preceding steps. A step that adds answer-relevant information, such as computing the remaining apples, has a low expected Step-RRI; a step that repeats the same calculation has a high expected Step-RRI.

Recent works (Ton et al., 2024; Yong et al., 2025) have analyzed reasoning using classical information-theoretic quantities such as mutual information and conditional mutual information. However, classical information-theoretic measures only quantify whether the current step is informative or not, without distinguishing whether the steps contribute jointly or individually to the final answer. Such aggregate information gains fail to capture whether a step contributes uniquely, has redundancy, or has complementary information with the preceding reasoning. In contrast, SLIDER characterizes repetitiveness directly by decomposing the information between the current step and its past reasoning steps about the final answer, enabling reference-free, step-level analysis of repetitive reasoning. Our main contributions can be summarized as follows:

Disentangling Step-wise Reasoning through PID: We propose a principled interpretability framework called SLIDER to analyze when and how individual reasoning steps contribute to the final answer. SLIDER is rooted in Partial Information Decomposition (PID), which decomposes the information about the final answer between two reasoning steps into unique, redundant, and synergistic components (see Fig. 2 for an illustrative example). Given the reasoning trajectories, we move across the steps in a sliding-window fashion and project them onto a meaningful embedding space. SLIDER discretizes the embeddings of the reasoning steps and the final answer and estimates the PID measures. To examine the effect of noisy reasoning steps on these PID components, we consider two consecutive reasoning steps, S_{i-1} and S_i, and inject independent synthetic noise into an individual step. We show, both theoretically (Theorem 1) and empirically on synthetic arithmetic data, that the unique information contributed by the perturbed step consistently decreases.

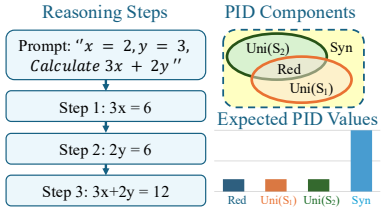

Figure 2: SLIDER *in a simple arithmetic problem:* Demonstrates high synergistic information between Step 1 (3x) and Step 2 (2y) because they jointly provide complete information about (3x+2y).

Quantifying Repetitive Reasoning through RRI: We introduce the *Step-wise Repetitive Reasoning Index (Step-RRI)*, an information-theoretic measure for characterizing repetitive reasoning at the step level. *Step-RRI* characterizes whether the answer-relevant information contributed by the current step is predominantly redundant with the preceding reasoning relative to its unique and synergistic contributions (Proposition 2). Furthermore, to quantify the overall repetitiveness of a reasoning trajectory, we define *Trajectory-RRI* (Definition 3) by aggregating repetitive behavior across individual steps. We theoretically demonstrate (see Theorem 2) how Step-RRI effectively captures repetitiveness. Our proof leverages monotonicity properties of the PID measure. We empirically validate the effectiveness of Step-RRI in detecting actual repetitiveness using the annotated redundancy class of PRMBench (Song et al., 2025). Step-RRI achieves a 10-point improvement in accuracy for step-level redundancy detection, outperforming embedding-similarity and information-gain baselines.

Linking Trajectory-RRI to Reasoning Efficiency: To assess the practical utility of Trajectory-RRI for capturing reasoning efficiency, we examine its relationship with actual reasoning length across multiple LLMs (see Table 1). We evaluate reasoning trajectories on GSM8K (Cobbe et al., 2021) and AIME2025 (Zhang & Team, 2025) from three models QwQ-32B (Team, 2025), DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025), and GPT-4.1 (OpenAI, 2025a). For these models, Trajectory-RRI is strongly correlated with reasoning length (Spearman's correlation coefficient, $\rho = 0.91$, 0.95, and 0.73, respectively, on GSM8K). We find a similarly strong association between Trajectory-RRI and reasoning length on AIME2025 (Zhang & Team, 2025). This consistent relationship indicates that

reasoning trajectories that are more repetitive, as determined by our measure, also tend to be longer and thus less efficient, highlighting a practical utility of Trajectory-RRI.

Leveraging Trajectory-RRI for Efficient Data Selection: Beyond post-hoc analysis, we demonstrate that Trajectory-RRI can be directly incorporated into reasoning data selection for supervised fine-tuning (SFT). Using GSM8K (Cobbe et al., 2021), we construct training datasets with different levels of average Trajectory-RRI by selecting reasoning trajectories generated by QwQ-32B (Team, 2025) and DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025). We use these datasets to fine-tune Qwen2.5-7B-Instruct (Team, 2024) with Low-Rank Adaptation (LoRA) (Hu et al., 2021). We find that the efficiency of the fine-tuned models, evaluated using an LLM judge, is strongly negatively correlated ($\rho = -0.97$) with the average Trajectory-RRI of the training data, while task performance remains largely preserved. These findings establish Trajectory-RRI as an actionable signal for selecting reasoning trajectories during fine-tuning, providing a practical approach to training models to reason more efficiently without sacrificing task performance.

1.1 Related Works

Classical information theoretic measures offer a lens to theoretically analyze and evaluate different aspects of LLMs (Farquhar et al., 2024; Kossen et al., 2024; Ali et al., 2025; Ton et al., 2024; Gan et al., 2025; Yong et al., 2025). Recent works Gan et al. (2025); Ton et al. (2024); Gan et al. (2025); Yong et al. (2025) use information-theoretic measures to evaluate LLM reasoning. Gan et al. (2025) explore slow-thinking in the multi-step reasoning of LLMs using information theory. Ton et al. (2024) use conditional mutual information to quantify stepwise information gain and identify failure modes in CoT reasoning. Yong et al. (2025) introduce two metrics, namely InfoGain and InfoBias, to quantify stepwise information gain and divergence from the original reasoning path, respectively, utilizing classical information-theoretic measures. *Although these works demonstrate the utility of information-theoretic measures for analyzing LLM reasoning, the structural interactions between reasoning steps, particularly their joint and individual contributions to the final answer, remain less explored. Also, theory-grounded approaches for detecting step-level repetitiveness without reference reasoning trajectories remain underexplored.*

Partial Information Decomposition (PID) (Williams & Beer, 2010; Venkatesh et al., 2024; Goswami et al., 2023; Pakman et al., 2021; Lyu et al., 2024) is an emerging area of research of information theory, with growing applications across diverse domains, including representation analysis, fairness, and multimodal learning, (Tax et al., 2017; Dutta et al., 2020; Hamman & Dutta, 2023; Ehrlich et al., 2022; Liang et al., 2023; Wollstadt et al., 2023; Mohamadi et al., 2023; Dewan et al., 2024; Dissanayake et al., 2024; Halder et al., 2025; Xiu et al., 2026). For instance, Liang et al. (2023) employ PID to analyze multimodal interactions, while Dewan et al. (2024) leverage PID to interpret diffusion models. Xiu et al. (2026) measure the "information spectrum" of Large vision-language models (LVLMs) by decomposing decision-relevant information into redundant, unique, and synergistic components. *Despite these advances, the application of PID to reasoning processes, particularly to analyze the informational structure of intermediate reasoning steps, remains largely unexplored. In this work, we extend PID to the study of structured reasoning, introducing a principled framework to quantify repetitiveness at the step level and then at the trajectory level.*

1.2 Background on PID

The classical measure of the total information about a target variable A that is contained in two random variables X and Y is given by mutual information $\mathrm{I}(A; X, Y)$ (Cover & Thomas, 2012). However, mutual information $\mathrm{I}(A; X, Y)$ does not disentangle what is uniquely contributed by each or shared by both. An emerging body of work called Partial Information Decomposition (PID) goes beyond classical measures and decomposes the joint information about a target A among multiple random variables X and Y into four non-negative measures (see Fig. 3). Here, unique information $\mathrm{Uni}(A{:}X|Y)$ and $\mathrm{Uni}(A{:}Y|X)$ capture the information that is exclusively provided by X or Y. Redundant information $\mathrm{Red}(A{:}X, Y)$ is the shared information about A in X and Y, and synergistic information

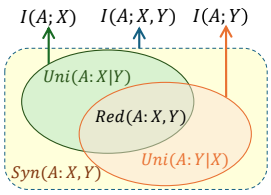

Figure 3: PID of $\mathrm{I}(A; X, Y)$

$\mathrm{Syn}(A{:}X, Y)$ emerges only when X and Y are both present together (complementary). We now formally define these quantities.

Definition 1 (Unique information (Bertschinger et al., 2014)). *Let Δ be the set of all joint distributions on (A, X, Y) and $\Delta_P = \{Q_{AXY} \in \Delta\colon Q_{AX} = P_{AX}$ and $Q_{AY} = P_{AY}\}$ be the set of joint distributions with same marginals on (A, X) and (A, Y) as the true distribution P_{AXY}. Then, $\mathrm{Uni}(A{:}X|Y) := \min_{Q \in \Delta_P} \mathrm{I}_Q(A; X|Y)$. Here, $\mathrm{I}_Q(A; X|Y)$ is the conditional mutual information under joint distribution Q_{AXY} instead of P_{AXY}.*

Defining any one of the PID terms is sufficient to derive the others, due to the following relationship:

$$\mathrm{I}(A; X) = \mathrm{Uni}(A{:}X|Y) + \mathrm{Red}(A{:}X, Y). \tag{1}$$

Intuitively, $\mathrm{Red}(A{:}X, Y)$ can be interpreted as the overlapping portion between $\mathrm{I}(A; Y)$ and $\mathrm{I}(A; X)$ (see Fig. 3). Hence, $\mathrm{Red}(A{:}X, Y) = \mathrm{I}(A; X) - \mathrm{Uni}(A{:}X|Y)$. Finally, the synergy corresponds to the remaining information: $\mathrm{Syn}(A{:}X, Y) = \mathrm{I}(A; X, Y) - \mathrm{Uni}(A{:}X|Y) - \mathrm{Uni}(A{:}Y|X) - \mathrm{Red}(A{:}X, Y)$, which can be computed once the unique and redundant information terms have been obtained. We also note an additional complementary relationship with conditional mutual information as follows:

$$\mathrm{I}(A; X|Y) = \mathrm{I}(A; X, Y) - \mathrm{I}(A; Y) = \mathrm{Uni}(A{:}X|Y) + \mathrm{Syn}(A{:}X, Y). \tag{2}$$

Toy Example: Let $Z{=}(Z_1, Z_2, Z_3)$ with each $Z_i \sim$ i.i.d. Bern(1/2). Let $X = (Z_1, Z_2, Z_3 \oplus N)$, $Y = (Z_2, N)$, and $N \sim$ Bern(1/2) which is independent of Z. Here, $\mathrm{I}(Z; X, Y) = 3$ bits. The unique information about Z only in X and not in Y is effectively in Z_1. Thus, $\mathrm{Uni}(Z{:}X|Y) = \mathrm{I}(Z; Z_1) = 1$ bit. Redundant information about Z that is in both X and Y is effectively in Z_2 and is given by $\mathrm{Red}(Z{:}X, Y) = \mathrm{I}(Z; Z_2) = 1$ bit. Synergistic information about Z that is not in either X or Y alone, but is in both of them together is effectively in the tuple $(Z_3 \oplus N, N)$, and is given by $\mathrm{Syn}(Z{:}X, Y){=}\mathrm{I}(Z; (Z_3 \oplus N, N)) = 1$ bit. This accounts for the 3 bits in $\mathrm{I}(Z; X, Y)$.

2 Proposed Interpretability Framework: SLIDER

Notations. Given an input question Q, we denote the LRM-generated reasoning trajectory by $S = \{S_1, S_2, \ldots, S_T\}$, where each S_i represents an intermediate reasoning step. We use $S_{<i} = \{S_1, S_2, ...S_{i-1}\}$ to denote all reasoning steps prior to step i. The final answer is denoted by A. We segment the reasoning path into paragraph-level units and consider each segment as a distinct reasoning step toward obtaining the final answer A, following Yong et al. (2025).

PID in reasoning trajectories. The total information of two consecutive reasoning steps S_{i-1} and S_i about the final answer A can be decomposed into four non-negative components as follows:

$$\mathrm{I}(A; S_{i-1}, S_i){=}\mathrm{Red}(A{:}S_{i-1}, S_i) + \mathrm{Uni}(A{:}S_{i-1}|S_i) + \mathrm{Uni}(A{:}S_i|S_{i-1}) + \mathrm{Syn}(A{:}S_{i-1}, S_i). \tag{3}$$

Each term in equation 3 interprets how the reasoning steps contribute toward the final answer A.

Redundant information $\mathrm{Red}(A{:}S_{i-1}, S_i)$ captures the overlapping information preserved in both steps. For example, consider an arithmetic problem where the first two reasoning steps are, $S_1 = \frac{x}{2}$ and $S_2 = \frac{x}{2} \times \frac{3}{5}$, and the final answer $A = x - \frac{x}{2} - \frac{x}{2} \times \frac{3}{5} = \frac{x}{5}$. In this case, the term $\frac{x}{2}$ computed in S_1 is reused in S_2 for subsequent computation. As a result, the two steps share overlapping information about the final answer, which results in redundant information.

Unique information $\mathrm{Uni}(A{:}S_{i-1}|S_i)$ or $\mathrm{Uni}(A{:}S_i|S_{i-1})$ captures the information in one step that cannot be obtained from the other. In the previous example, step S_2 involves an additional computation, taking $\frac{3}{5}$ of $\frac{x}{2}$ which is not present in S_1 and therefore unique to S_2. This implies that S_2 provides unique information about the final answer that cannot be obtained from S_1.

Synergistic information $\mathrm{Syn}(A{:}S_{i-1}, S_i)$ arises when both steps jointly provide complementary information necessary for determining the final answer, which cannot be obtained from either step alone. For example, consider an arithmetic problem where the first two reasoning steps are $S_1 = 3x$ and $S_2 = 2y$, and the final answer $A = 3x + 2y$. Here, neither S_1 nor S_2 is sufficient to determine the final answer A. A is obtained only when they are considered together, resulting in synergy.

PID under errors. Intuitively, if a reasoning step contains errors, its task-relevant contribution toward the final answer is diminished. As a result, we expect a reduction in its unique information compared to the previous, non-erroneous case. We formalize this argument in the following theorem.

Theorem 1 (Uniqueness under noise). *Consider an erroneous step $S_i' = S_i + N$ where N is independent additive noise, $N \perp (A, S_i, S_{i-1})$. Then,* $\mathrm{Uni}(A{:}S_i|S_{i-1}) \geq \mathrm{Uni}(A{:}S_i'|S_{i-1})$.

Theorem 1 suggests that introducing synthetic noise into a reasoning step reduces its predictive contribution to the final answer. For example, consider the arithmetic problem in Fig. 2 with reasoning steps $S_1 = 3x$ and $S_2 = 2y$, and the final answer $A = 3x + 2y$. If the calculation in S_2 becomes erroneous due to injected noise, $S_2' = 2y + N$, where N is independent random noise, then $\mathrm{Uni}(A{:}S_2|S_1) \geq \mathrm{Uni}(A{:}S_2'|S_1)$. This indicates that the predictive power of S_2 towards A cannot increase under noise. The proof is given in Appendix B.1.

PID to identify repetitive reasoning. If the current step S_i revisits prior computations in $S_{<i} = \{S_1, S_2, ...S_{i-1}\}$ or presents an alternative formulation to reach the same conclusion, we would consider S_i as repetitive. To identify repetitive reasoning, we propose the following decomposition:

Proposition 1 (Decomposition for repetitive reasoning). *The total information of $S_{<i}$ and S_i about the final answer A can be decomposed into four non-negative components:*

$$\mathrm{I}(A; S_{<i}, S_i) = \mathrm{Red}(A{:}S_{<i}, S_i) + \mathrm{Uni}(A{:}S_{<i}|S_i) + \mathrm{Uni}(A{:}S_i|S_{<i}) + \mathrm{Syn}(A{:}S_{<i}, S_i).$$

When S_i is repetitive, we expect redundant information $\mathrm{Red}(A{:}S_{<i}, S_i)$ to be high and dominant over other terms. This leads to our next proposition, a step-wise measure of repetitiveness that contrasts the redundant contribution of S_i with its unique and synergistic contributions.

Definition 2 (Step-wise Repetitive Reasoning Index (Step-RRI)). *For some $\eta \geq 1$ and $i > 1$, our proposed measure of repetitiveness in a reasoning step S_i is defined as,*

$$\textit{Step-RRI} := \mathrm{RRI}_i = \mathrm{Red}(A{:}S_i, S_{<i}) - \eta \cdot \max\{\mathrm{Uni}(A{:}S_i|S_{<i}), \mathrm{Syn}(A{:}S_i, S_{<i})\}. \tag{4}$$

For intuition, consider the simplified case where $S_{<i}{=}S_i$. *We will now show how the uniqueness in S_i and $S_{<i}$ and synergy become zero, leaving redundancy as the only positive term.* First note that the conditional mutual information $\mathrm{I}(A; S_{<i}|S_i) = \mathrm{I}(A; S_i|S_{<i}) = 0$ since $\mathrm{I}(A; S_i|S_{<i}) \overset{(a)}{=} H(S_i|S_{<i}) - H(S_i|A, S_{<i}) = 0$. where (a) is from definition. Then, $\mathrm{Uni}(A{:}S_i|S_{<i}) \overset{(b)}{=} \mathrm{I}(A; S_i|S_{<i}) - \mathrm{Syn}(A{:}S_i, S_{<i}) \overset{(c)}{\leq} \mathrm{I}(A; S_i|S_{<i}) = 0$, where (b) follows from equation 2 and (c) is from the non-negativity of PID terms. Similarly, $\mathrm{Uni}(A{:}S_{<i}|S_i) \leq \mathrm{I}(A; S_{<i}|S_i) = 0$. Then, $\mathrm{Syn}(A{:}S_{<i}, S_i) = \mathrm{I}(A; S_{<i}|S_i) - \mathrm{Uni}(A{:}S_{<i}|S_i)$ is also 0. Lastly, $\mathrm{Red}(A{:}S_i, S_{<i}) = \mathrm{I}(A; S_i) - \mathrm{Uni}(A{:}S_i|S_{<i}) = \mathrm{I}(A; S_i) = H(A) - H(A|S_i)$, which is positive as long as there is some dependence between A and S_i. As a result, Step-RRI will also be positive. We formalize this intuition in the next theorem. The proof (see Appendix B.2) leverages interesting monotonicity properties of the PID measures. In particular, we leverage the fact that $\mathrm{Uni}(A{:}B|C \cup C') \leq \mathrm{Uni}(A{:}B|C)$ (Monotonicity under adversarial side information; see Lemma 2 in Appendix B).

Theorem 2 (Step-RRI under Repetitive Reasoning). *Let the current step $S_i = (S_A, S_N)$ where S_A is relevant to the answer A and S_N is additional independent text, i.e., $\mathrm{I}(A; S_A) > 0$ and $S_N \perp A, S_{<i}$ (noise). Suppose the current step is repetitive with the logic already contained in past steps $S_{<i}$, i.e., the relevant part $S_A \subseteq S_{<i}$. Then, we have: (i)* $\mathrm{Uni}(A{:}S_i|S_{<i})$ *and* $\mathrm{Syn}(A{:}S_i, S_{<i})$ *are both zero; (ii) Step-RRI* $= \mathrm{Red}(A{:}S_i, S_{<i}) - \eta \cdot \max\{\mathrm{Uni}(A{:}S_i|S_{<i}), \mathrm{Syn}(A{:}S_i, S_{<i})\} > 0$.

Theorem 2 highlights that when a reasoning step S_i only repeats information already contained in $S_{<i}$, its unique and synergistic information contributions vanish, and the step becomes purely redundant with respect to the answer, resulting in a positive Step-RRI.

Next, to characterize reasoning repetitiveness across the entire reasoning trajectory at the individual-problem level, we define *Trajectory-RRI*. Trajectory-RRI quantifies the overall repetitiveness of a reasoning trajectory for a given question by counting the number of reasoning steps identified as repetitive by our measure Step-RRI.

Definition 3 (Trajectory Repetitive Reasoning Index (Trajectory-RRI)). *For a given question Q with a reasoning trajectory of length T, we define the Trajectory-RRI as, Trajectory-RRI* $:= \mathrm{RRI}(Q) = \sum_{i=2}^{T} \mathbb{I}\left[\mathrm{RRI}_i > 0\right]$, *where $\mathbb{I}[\cdot]$ denotes the indicator function.*

Implementation of SLIDER. The estimation of PID measures for high-dimensional data with continuous target variables is a non-trivial task. We propose a tractable approach for PID estimation

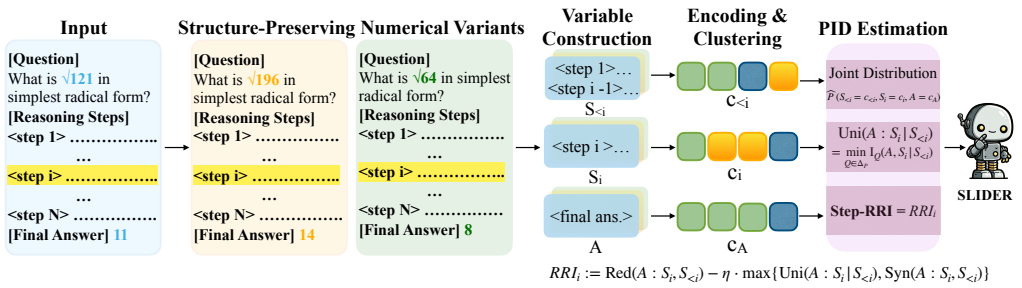

Figure 4: *Our proposed framework* SLIDER. Given a question and its reasoning trajectory, the sample generator creates multiple variants by changing only the numerical values while preserving the underlying problem structure. Each generated reasoning trajectory is then segmented into individual reasoning steps using the `<step i>` tags, from which we construct the preceding reasoning $S_{<i}$ and the current step S_i in text form. Then, we obtain embeddings of the current step S_i, preceding steps $S_{<i}$, and final answer A using an encoder-only language model and discretize the resulting embeddings via k-means clustering. Finally, we estimate the joint distribution from the discrete representations, compute the PID measures, and Step-RRI.

at the step level in reasoning trajectories. Our objective is to obtain stable PID estimates that are both intuitive and interpretable. SLIDER consists of several key steps (see Fig. 4) as follows:

Structure-Preserving Sample Generation. To estimate PID at the individual reasoning-step level, we require multiple samples to estimate the underlying joint distributions. A single reasoning trajectory provides insufficient samples for reliable estimation, while independently generated trajectories may not match exactly in their reasoning structure and step correspondence. To address this challenge, we construct multiple *structure-preserving numerical variants* of each problem, ensuring exact semantic and structural alignment of corresponding steps across trajectories. These aligned samples enable the estimation of step-level joint distributions and, subsequently, the PID measures and RRI measures.

Specifically, for each original question and its associated reasoning trajectory, we use GPT-4o-Mini (OpenAI, 2024) (the prompt template is in Appendix C.1), unless otherwise stated, to iteratively generate numerical variants. The generator modifies only the numerical values while preserving the question format, reasoning logic, step ordering, and `<step i>` structure. A generated sample is accepted only if its trajectory contains the same number of reasoning steps as its input, ensuring correspondence between S_i across samples. Each accepted sample is then used as the reference for the next generation step, forming a recursive chain that increases numerical diversity while preserving the underlying reasoning structure. If the generated trajectory does not preserve the step count, the sample is discarded, and generation is reset to the original example. We generate up to 50 variants per problem and terminate generation after five consecutive structural mismatches. This procedure yields a collection of aligned reasoning trajectories from which we estimate the joint distributions of the step-level representations required for PID.

Encoding and PID Estimation. Each generated reasoning trajectory is first segmented into individual reasoning steps using the `<step i>` tags. For each step i, we define the tagged segment as the current-step variable S_i and construct the preceding-reasoning variable $S_{<i}$ by concatenating all steps before S_i in their original textual form. Next, we use an encoder-only LLM, all-MiniLM-L6-v2 (Reimers & Gurevych, 2019), unless otherwise stated, to obtain embeddings of the preceding steps $S_{<i}$, the current step S_i, and the final answer A, respectively. We then discretize these continuous embeddings using k-means clustering with 10 clusters to obtain categorical variables (refer to Table 2, Appendix C.2 for a sensitivity analysis of the number of clusters). Based on the resulting cluster assignments, we estimate the joint distribution, $\widehat{P}(S_{<i} = c_{<i},\ S_i = c_i,\ A = c_A)$, with a slight abuse of notation where we still use S_i to denote the discrete embeddings of the corresponding steps. Then, according to Definition 1, we compute the PID measures: redundant information $\mathrm{Red}(A{:}S_{<i}, S_i)$, unique information $\mathrm{Uni}(A{:}S_{<i}|S_i)$ and $\mathrm{Uni}(A{:}S_i|S_{<i})$, and synergistic information $\mathrm{Syn}(A{:}S_{<i}, S_i)$ from the estimated joint distribution. Specifically, we utilize the CVX estimator proposed by Liang et al. (2023) for PID estimation. Finally, following Definition 2 and 3, we compute Step-RRI and Trajectory-RRI, respectively. For further implementation details, refer to Appendix C.1.

3 EXPERIMENTAL RESULTS

In this section, we evaluate SLIDER across synthetic arithmetic problems, PRMBench (Song et al., 2025), GSM8K (Cobbe et al., 2021), and AIME2025 (Zhang & Team, 2025). We examine the effect of injected errors on PID components, evaluate Step-RRI for detecting step-level ground-truth redundancy against cosine similarity and InfoGain (Yong et al., 2025), and assess the relationship between Trajectory-RRI and reasoning length across reasoning and non-reasoning models. Finally, we use Trajectory-RRI to guide data selection for fine-tuning Qwen2.5-7B-Instruct (Team, 2024) toward more efficient reasoning. Additional dataset details are provided in Appendix D.

SLIDER uncovers interpretable information dynamics under synthetic noise. We consider the following problem: *Arithmetic problem:* "x = {x}, y = {y}. Please calculate the following: 1. $3x$, 2. $2y$, 3. $3x + 2y$." We sample two integers x and y independently and uniformly from $[1, 10^5)$. We generate 500 samples in exact format, each containing exactly three reasoning steps. The intermediate steps are $S_1 = 3x$, $S_2 = 2y$, and $S_3 = 3x + 2y$, and the final answer $A = 3x + 2y$ (see Fig. 5(d)). To introduce errors, we perturb an individual step with probability p by adding an integer sampled uniformly from $[-1000, 10^6]$ to its value, while keeping the remaining steps and the final answer unchanged. We vary p to analyze how increasing the probability of errors affects the PID components.



(a) Erroneous S_1 **(b) Erroneous S_2** **(c) Erroneous S_3** **(d) Reasoning Steps**

Figure 5: *Interpretable trends of PID measures under errors for synthetic arithmetic problems.* (a,b) Increasing error probability p reduces synergy and the unique information of erroneous steps, while increasing the unique contribution of correct steps. (c) Perturbing S_3 similarly decreases $\mathrm{Uni}(S_3)$ as p increases. (d) Reference example for the reasoning steps.

Fig. 5 illustrates how the information contributed by intermediate reasoning steps toward the final answer evolves under synthetic noise. We first consider $S_1 = 3x$ and $S_2 = 2y$ in Fig. 5(a,b). For this arithmetic problem, individually, neither $S_1 = 3x$ nor $S_2 = 2y$ is sufficient to determine the target A. However, when S_1 and S_2 are considered jointly, their combined information fully specifies A (considering no error in the reasoning steps). This implies that the information about A is predominantly synergistic/complementary across S_1 and S_2. As the error probability increases in either S_1 or S_2, synergy decreases because the error limits the inference of the final answer, even when considering both steps together. At the same time, the unique information of the erroneous step decreases, while that of the corresponding correct step increases, as the correct step carries answer-relevant information that is increasingly unavailable from its corrupted counterpart. In Fig. 5(c), we consider the same problem with the last two reasoning steps, $S_2 = 2y$ and $S_3 = 3x + 2y$, and the final answer $A = 3x + 2y$. When S_3 is correct, it directly determines the final answer A and therefore contributes substantial unique information relative to $S_2 = 2y$. However, if there is an error in the calculation in S_3, its uniqueness decreases, since some examples can no longer provide useful information about A that is not already available from S_2. We can also use SLIDER to identify incorrectness for the GSM8K word problem (Cobbe et al., 2021) (see details in Appendix D.1).

SLIDER identifies step-wise repetitiveness in reasoning trajectories. To assess whether Step-RRI aligns with the actual repetitiveness between reasoning steps, we analyze reasoning trajectories generated by QwQ-32B on three GSM8K word problems. Implementation details and additional experiments using reasoning trajectories generated by DeepSeek-R1-Distill-Qwen-32B (see Fig. 11) are provided in Appendix D.2. Fig. 6 presents the corresponding PID and Step-RRI estimates, illustrating how answer-relevant information evolves across reasoning steps and how repetitive steps emerge within each trajectory based on Step-RRI. We observe that for Problem 1, synergistic information is initially dominant, indicating that the early reasoning steps provide complementary information required to reach the final answer. As the reasoning progresses toward the answer, the unique infor-

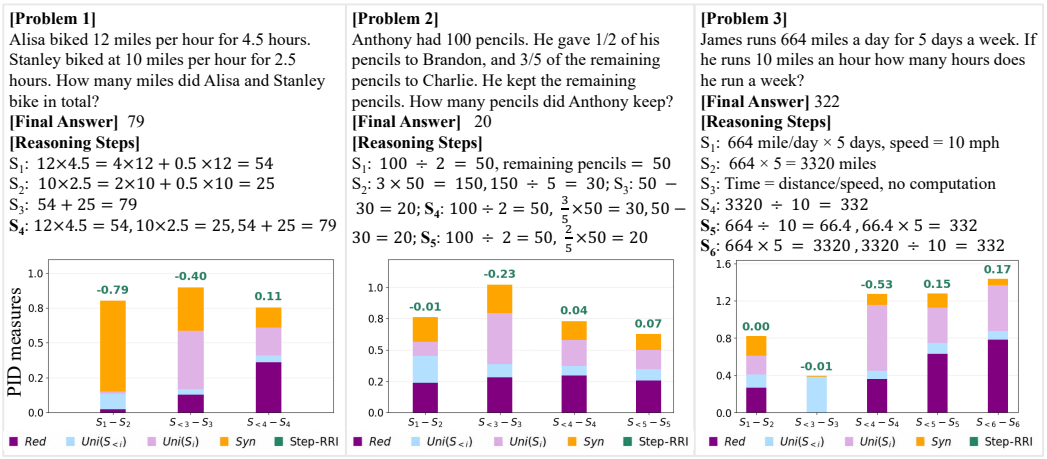

Figure 6: *PID estimation across reasoning trajectories.* Redundant information becomes dominant compared to uniqueness in S_i and synergistic information when the current step S_i revisits answer-relevant information already established in the preceding steps $S_{<i}$, resulting in positive Step-RRI.

mation of the current step increases. Specifically, $\mathrm{Uni}(S_3)$ is maximized at $S_{<3}$–S_3 for Problems 1 and 2, while $\mathrm{Uni}(S_4)$ is maximized at $S_{<4}$–S_4 for Problem 3. This suggests that this reasoning step S_i contributes information that is largely unavailable from previous steps, as it directly contains the final answer. SLIDER further localizes repetitive reasoning to individual steps using Step-RRI. The redundant contribution dominates the unique contribution of S_i and its synergistic contribution with $S_{<i}$ when the model either repeats earlier computations or performs alternative re-computations that recover answer-relevant information already contained in $S_{<i}$. As a result, Step-RRI becomes positive. Accordingly, with $\eta = 1.25$, Step-RRI becomes positive at S_4 in Problem 1, S_4 and S_5 in Problem 2, and S_5 and S_6 in Problem 3. Thus, Step-RRI enables SLIDER to localize repetitive reasoning to specific steps within each problem.

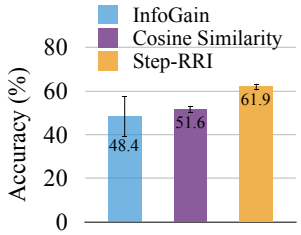

Figure 7: *Step-level redundancy detection.* Step-RRI achieves higher accuracy than baselines on PRMBench.

Next, we quantitatively evaluate the effectiveness of Step-RRI for identifying true repetitive reasoning steps using PRMBench (Song et al., 2025) since this dataset has annotations for step-level redundancy. We focus specifically on the redundancy error class and compare Step-RRI against two baselines: InfoGain (Yong et al., 2025) and Cosine Similarity. For each method, we calibrate the decision threshold on a held-out set by training a linear classifier and use the resulting threshold for step-level ground-truth redundancy detection on the test set. Fig. 7 reports the detection performance. Step-RRI achieves the highest accuracy of 61.91%, improving over InfoGain and Cosine Similarity by 13.53 and 10.31 percentage points. More details on the baselines are in Appendix D.2.

SLIDER reveals a strong correlation between Trajectory-RRI and reasoning length. To demonstrate the practical utility of Trajectory-RRI, we examine its relationship with actual reasoning length across multiple models. We consider the QwQ-32B (Team, 2025), DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025), and and GPT-4.1 (OpenAI, 2025a). For QwQ-32B and DeepSeek-R1-Distill-Qwen-32B, each response contains a thinking trajectory followed by the final response, separated by the `</think>` tag; we use the text preceding this tag as the reasoning trajectory. For GPT-4.1, we consider the entire generated response as the reasoning trajectory. See Appendix D.3 for additional details on the experimental setup.

We evaluate on GSM8K (Cobbe et al., 2021) and AIME2025 (Zhang & Team, 2025). For GSM8K, we consider 1,000 randomly selected problems with correct final answers, evaluating QwQ-32B, DeepSeek-R1-Distill-Qwen-32B, and GPT-4.1. For each trajectory, we compute Trajectory-RRI using SLIDER and measure reasoning length as the number of tokens in the corresponding reasoning trajectory. To mitigate variability from problem-specific step segmentation, we group trajectories by reasoning length and report the average Trajectory-RRI and reasoning length within each bin. As shown in Fig. 8 and Table 1, average Trajectory-RRI is positively associated with reasoning

Table 1: Spearman correlation between average Trajectory-RRI and reasoning length.

	DeepSeek-R1		QwQ-32B		GPT-4.1	
Dataset	ρ	p-val	ρ	p-val	ρ	p-val
GSM8K	0.95	0.00	0.91	0.00	0.73	0.00
AIME2025	0.89	0.01	0.75	0.01	0.82	0.03

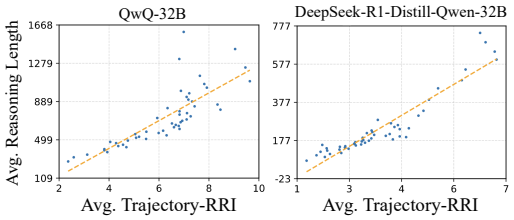

Figure 8: *Trajectory-RRI strongly correlates with reasoning length.* Avg. Trajectory-RRI exhibits a positive monotonic relationship with average thinking tokens for both models (left; $\rho = 0.91$, right; $\rho = 0.95$) on GSM8K, indicating higher RRI is associated with longer trajectories.

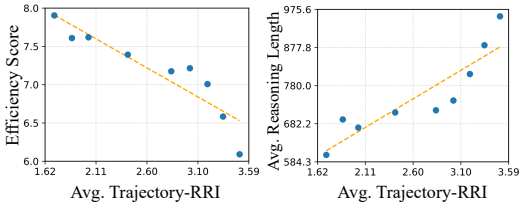

Figure 9: *Trajectory-RRI guided fine-tuning improves reasoning efficiency.* As the average Trajectory-RRI of the training data increases, the efficiency score of the fine-tuned model on GSM8K decreases (left; $\rho = -0.97$), while its average tokens increase (right; $\rho = 0.98$).

length across the evaluated settings. On GSM8K, the Spearman correlations (see Appendix D for its definition) are $\rho = 0.95$ for DeepSeek-R1-Distill-Qwen-32B, $\rho = 0.91$ for QwQ-32B, and $\rho = 0.73$ for GPT-4.1. On AIME2025, DeepSeek-R1-Distill-Qwen-32B, QwQ-32B, and GPT-4.1, yield Spearman correlations of $\rho = 0.89, 0.75$, and 0.82, respectively. The corresponding p-values reported in Table 1 are also small, indicating that these correlations are statistically significant. *The consistently strong positive correlations across these models indicate that trajectories identified as more repetitive by our measure also tend to require more reasoning tokens.*

Trajectory-RRI serves as a signal for SFT data selection. Motivated by the strong positive correlation between average Trajectory-RRI and reasoning length, we investigate whether average Trajectory-RRI can be used to guide data selection for supervised fine-tuning toward more efficient reasoning while preserving task performance. We use 1,000 GSM8K problems, along with their corresponding reasoning trajectories generated by QwQ-32B and DeepSeek-R1-Distill-Qwen-32B, yielding two candidate reasoning trajectories for each problem. Using SLIDER, we estimate the Trajectory-RRI of each trajectory and construct training sets with varying average Trajectory-RRI by selecting, for each problem, the trajectory with the higher Trajectory- RRI with probability p. We then fine-tune Qwen2.5-7B-Instruct (Team, 2024) on each dataset using Low-Rank Adaptation (LoRA) Hu et al. (2021) and evaluate the resulting models on the GSM8K test set in terms of accuracy, reasoning length, and an LLM-as-a-judge efficiency score. As shown in Fig. 9, the average Trajectory-RRI of the training data exhibits a strong negative correlation with an LLM-as-a-judge efficiency score of the fine-tuned models (Spearman's correlation $\rho = -0.97$, $p - \text{val} = 2.2 \times 10^{-5}$) and a strong positive correlation with their actual reasoning length, measured by the total number of tokens generated during inference ($\rho = 0.98$, $p - \text{val} = 1.9 \times 10^{-6}$), while test accuracy remains within a comparable range (see Table 3 in Appendix D.4). *These results demonstrate that Trajectory-RRI can serve as an actionable signal for curating fine-tuning data: training on trajectories with lower Trajectory-RRI leads to more concise and efficient reasoning without a substantial degradation in task performance.* See Appendix D.4 and D.5 for additional implementation and computational details, respectively.

Conclusion. This work introduces SLIDER, a novel information-theoretic framework for interpreting reasoning trajectories using Partial Information Decomposition. We propose *Step-RRI* to identify repetitive reasoning at the individual-step level and *Trajectory-RRI* to quantify overall repetitiveness within a reasoning trajectory for a specific problem. Our results demonstrate that RRI measures effectively identify true repetitive reasoning, strongly correlate with actual reasoning length across LLMs, and can guide data selection for fine-tuning toward more efficient reasoning while largely preserving task performance. We discuss limitations and future work in Appendix A.

4 AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic datasets. We did not use generative AI tools to develop the core methodology or formulate or prove the theoretical results; the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to assist with drafting and editing the manuscript, developing initial code, and brainstorming ideas for figures. All AI-assisted work was reviewed by the authors, and LLM-generated code was verified and tested for correctness by one author. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Riccardo Ali, Francesco Caso, Christopher Irwin, and Pietro Liò. Entropy-lens: The information signature of transformer computations. *arXiv preprint arXiv:2502.16570*, 2025.

Pradeep Kr Banerjee, Eckehard Olbrich, Jürgen Jost, and Johannes Rauh. Unique informations and deficiencies. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 32–38, 2018.

Nils Bertschinger, Johannes Rauh, Eckehard Olbrich, Jürgen Jost, and Nihat Ay. Quantifying unique information. *Entropy*, 16(4):2161–2183, 2014.

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

Thomas M Cover and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

Shaurya Dewan, Rushikesh Zawar, Prakanshul Saxena, Yingshan Chang, Andrew Luo, and Yonatan Bisk. Diffusion pid: Interpreting diffusion via partial information decomposition. *Advances in Neural Information Processing Systems*, 37:2045–2079, 2024.

Pasan Dissanayake, Faisal Hamman, Barproda Halder, Ilia Sucholutsky, Qiuyi Zhang, and Sanghamitra Dutta. Quantifying knowledge distillation using partial information decomposition. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Sanghamitra Dutta, Praveen Venkatesh, Piotr Mardziel, Anupam Datta, and Pulkit Grover. An information-theoretic quantification of discrimination with exempt features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3825–3833, 2020.

David A Ehrlich, Andreas C Schneider, Michael Wibral, Viola Priesemann, and Abdullah Makkeh. Partial information decomposition reveals the structure of neural representations. *arXiv preprint arXiv:2209.10438*, 2022.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Zeyu Gan, Yun Liao, and Yong Liu. Rethinking external slow-thinking: From snowball errors to probability of correct reasoning. *arXiv preprint arXiv:2501.15602*, 2025.

Chaitanya Goswami, Amanda Merkley, and Pulkit Grover. Computing unique information for poisson and multinomial systems. *arXiv preprint arXiv:2305.07013*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Barproda Halder, Faisal Hamman, Pasan Dissanayake, Qiuyi Zhang, Ilia Sucholutsky, and Sanghamitra Dutta. Towards formalizing spuriousness of biased datasets using partial information decomposition. *Transactions on Machine Learning Research*, 2025.

Faisal Hamman and Sanghamitra Dutta. Demystifying local and global fairness trade-offs in federated learning using partial information decomposition. *arXiv preprint arXiv:2307.11333*, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*, 2024.

Paul Pu Liang, Chun Kai Ling, Yun Cheng, Alex Obolenskiy, Yudong Liu, Rohan Pandey, Alex Wilf, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal learning without labeled multimodal data: Guarantees and applications. *arXiv preprint arXiv:2306.04539*, 2023.

Aobo Lyu, Andrew Clark, and Netanel Raviv. Explicit formula for partial information decomposition. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pp. 2329–2334, 2024. doi: 10.1109/ISIT57864.2024.10619369.

Salman Mohamadi, Gianfranco Doretto, and Donald A Adjeroh. More synergy, less redundancy: Exploiting joint mutual information for self-supervised learning. *arXiv preprint arXiv:2307.00651*, 2023.

OpenAI. Learning to reason with LLMs, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, July 2024. Accessed: 2026-08-06.

OpenAI. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/, April 2025a. Accessed: 2026-09-25.

OpenAI. GPT-5 mini. https://developers.openai.com/api/docs/models/gpt-5-mini, 2025b. OpenAI API model documentation.

Ari Pakman, Amin Nejatbakhsh, Dar Gilboa, Abdullah Makkeh, Luca Mazzucato, Michael Wibral, and Elad Schneidman. Estimating the unique information of continuous variables. *Advances in neural information processing systems*, 34:20295–20307, 2021.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. Prmbench: A fine-grained and challenging benchmark for process-level reward models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25299–25346, 2025.

Tycho Tax, Pedro Mediano, and Murray Shanahan. The partial information decomposition of generative neural network models. *Entropy*, 19(9):474, 2017.

Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, 2025.

Jean-Francois Ton, Muhammad Faaiz Taufiq, and Yang Liu. Understanding chain-of-thought in llms through information theory. *arXiv preprint arXiv:2411.11984*, 2024.

Praveen Venkatesh, Corbett Bennett, Sam Gale, Tamina Ramirez, Greggory Heller, Severine Durand, Shawn Olsen, and Stefan Mihalas. Gaussian partial information decomposition: Bias correction and application to high-dimensional data. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Paul L Williams and Randall D Beer. Nonnegative decomposition of multivariate information. *arXiv preprint arXiv:1004.2515*, 2010.

Patricia Wollstadt, Sebastian Schmitt, and Michael Wibral. A rigorous information-theoretic definition of redundancy and relevancy in feature selection based on (partial) information decomposition. *J. Mach. Learn. Res.*, 24:131–1, 2023.

Lixin Xiu, Xufang Luo, and Hideki Nakayama. A comprehensive information-decomposition analysis of large vision-language models. *arXiv preprint arXiv:2603.29676*, 2026.

Xixian Yong, Xiao Zhou, Yingying Zhang, Jinlin Li, Yefeng Zheng, and Xian Wu. Think or not? exploring thinking efficiency in large reasoning models via an information-theoretic lens. *arXiv preprint arXiv:2505.18237*, 2025.

Jerrold H Zar. Spearman rank correlation. *Encyclopedia of biostatistics*, 7, 2005.

Yifan Zhang and Math-AI Team. American invitational mathematics examination (aime) 2025, 2025. URL `https://huggingface.co/datasets/math-ai/aime25`.

Appendix

A Limitations and Broader Impacts **14**

B Appendix to Theoretical Results **14**

B.1 Proof to Theorem 1 14

B.2 Proof to Theorem 2 15

C Appendix to Implementation of SLIDER **16**

C.1 Structure-Preserving Sample Generation 16

C.2 Sensitivity Analysis 18

D Appendix to Experiments **18**

D.1 PID measures under synthetic noise 19

D.2 Step-wise Repetitiveness 19

D.3 Trajectory-RRI versus Reasoning Length 22

D.4 Trajectory-RRI Guided Data Selection 22

D.5 Runtime and Resource Usage 23

A Limitations and Broader Impacts

Limitations. Our framework estimates PID from encoded and discretized representations of reasoning steps. This process may introduce information loss and make the estimated PID components sensitive to the choice of encoder, clustering procedure, and other estimation parameters. Future work could investigate continuous PID estimators and alternative representations to reduce this dependence. In addition, our structure-preserving sample generation relies on an LLM to generate aligned numerical variants, and the quality of the resulting PID estimates depends on how well these variants preserve the underlying reasoning structure. Evaluating step-level incorrectness under synthetic noise also requires knowledge of the expected problem structure or an approximate correspondence to correct reasoning steps, which may limit its applicability to domains where such structure is difficult to obtain. Finally, our experiments primarily focus on mathematical reasoning tasks. Evaluating SLIDER and RRI measures across broader reasoning domains and model families remains an important direction for future work.

Broader Impacts. Quantifying how individual steps contribute to a model's reasoning has broader implications for understanding and improving large language models. By characterizing repetitive reasoning at both the individual-step and trajectory levels, Step-RRI and Trajectory-RRI can support more interpretable auditing of reasoning behavior and provide signals for selecting more efficient reasoning trajectories during post-training. In particular, RRI measures have the potential to serve as theoretically grounded metrics for data selection when the goal is to promote efficient reasoning. While commonly used proxies such as reasoning length provide a coarse measure for curating efficient reasoning data, RRI directly characterizes repetitiveness based on the information contributed by reasoning steps, providing a principled basis for selecting less repetitive trajectories. More broadly, SLIDER provides an information-theoretic perspective for studying when reasoning steps contribute new, shared, or complementary answer-relevant information. Such tools may help researchers diagnose inefficient or unreliable reasoning and develop models that produce more concise and interpretable reasoning trajectories.

B Appendix to Theoretical Results

First, we include two results on PID from (Banerjee et al., 2018).

Lemma 1 (Monotonicity under local operations on B (Banerjee et al., 2018))**.** *Let $B' = f(B)$ where $f(\cdot)$ is a deterministic function. Then,*

$$\mathrm{Uni}(A{:}B|C) \geq \mathrm{Uni}(A{:}B'|C).$$

This result is derived in (Banerjee et al., 2018, Lemma 31).

Lemma 2 (Monotonicity under adversarial side information (Banerjee et al., 2018))**.** *For all (A, B, C, C'), we have:*

$$\mathrm{Uni}(A{:}B|C \cup C') \leq \mathrm{Uni}(A{:}B|C).$$

This result is derived in (Banerjee et al., 2018, Lemma 32).

B.1 Proof to Theorem 1

Proof of Theorem 1. The proof follows from monotonicity under local operations, Lemma 1, since $S_i' = S_i + N$ is a local operation on S_i.

We provide the detailed proof here for completeness. Since $N \perp (A, S_i, S_{i-1})$, the random variables (A, S_{i-1}), S_i, and S_i' form a Markov chain, $(A, S_{i-1}) - S_i - S_i'$. Let P' be the marginal distribution of (S_i', A, S_{i-1}), and P be the marginal distribution of (S_i, A, S_{i-1}). Let $Q^* = \arg\min_{Q \in \Delta_P} I_Q(S_i; A \mid S_{i-1})$, and let

$$Q^{*'}(s_i', a, s_{i-1}) = \sum_{s_i} P'(s_i' \mid s_i) Q^*(s_i, a, s_{i-1}).$$

Then $Q^{*'} \in \Delta_{P'}$. By definition,

$$\begin{aligned}\mathrm{Uni}(S_i{:}A|S_{i-1}) &= I_{Q^*}(S_i; A \mid S_{i-1}) \\ &\overset{(a)}{\geq} I_{Q^{*'}}(S_i'; A \mid S_{i-1}) \\ &\geq \min_{Q' \in \Delta_{P'}} I_{Q'}(S_i'; A \mid S_{i-1}) \\ &= \mathrm{Uni}(S_i'{:}A|S_{i-1}),\end{aligned}$$

Here, (a) uses the conditional form of the data processing inequality. Therefore, $\mathrm{Uni}(S_i'{:}A \mid S_{i-1}) \leq \mathrm{Uni}(S_i{:}A \mid S_{i-1})$, i.e., the unique information contributed by the noisy step S_i' about A cannot exceed that of the original step S_i.

□

B.2 Proof to Theorem 2

Proof of Theorem 2. To begin with,

$$\begin{aligned}\mathrm{Uni}(A{:}S_i|S_{<i}) &\overset{(a)}{=} \mathrm{I}(A; S_i|S_{<i}) - \mathrm{Syn}(A{:}S_i, S_{<i}) \\ &\overset{(b)}{\leq} \mathrm{I}(A; S_i|S_{<i}) \\ &\overset{(c)}{=} \mathrm{I}(A; S_A, S_N|S_{<i}) \\ &\overset{(d)}{=} \mathrm{I}(A; S_N|S_{<i}) + \mathrm{I}(A; S_A|S_{<i}, S_N) \\ &\overset{(e)}{=} 0 + \mathrm{I}(A; S_A|S_{<i}, S_N) \\ &\overset{(f)}{=} 0.\end{aligned}$$

Here, step (a) holds from equation 2 and step (b) holds from the non-negativity of PID terms, $\mathrm{Syn}(A{:}S_i, S_{<i}) \geq 0$. Step (c) holds because $S_i = (S_A, S_N)$. Step (d) follows directly from the chain rule of mutual information. Step (e) holds because $S_N \perp A, S_{<i}$ and step (f) holds because $S_A \subseteq S_{<i}$.

Similarly, $\mathrm{Syn}(A{:}S_i, S_{<i}) = \mathrm{I}(A; S_i|S_{<i}) - \mathrm{Uni}(A{:}S_i|S_{<i}) \leq \mathrm{I}(A; S_i|S_{<i}) = 0$.

Next, since Step-RRI $:= \mathrm{Red}(A{:}S_i, S_{<i}) - \eta \cdot \max\{\mathrm{Uni}(A{:}S_i|S_{<i}), \mathrm{Syn}(A{:}S_i, S_{<i})\}$, and $\max\{\mathrm{Uni}(A{:}S_i|S_{<i}), \mathrm{Syn}(A{:}S_i, S_{<i})\} = 0$, we have,

$$\begin{aligned}\text{Step-RRI} = \mathrm{Red}(A{:}S_i, S_{<i}) &\overset{(a)}{=} \mathrm{I}(A; S_{<i}) - \mathrm{Uni}(A{:}S_{<i}|S_i) \\ &\overset{(b)}{\geq} \mathrm{I}(A; S_{<i}) - \mathrm{Uni}(A{:}S_{<i}|S_A) \\ &\overset{(c)}{=} \mathrm{Red}(A{:}S_A, S_{<i}) \\ &\overset{(d)}{=} \mathrm{I}(A; S_A) - \mathrm{Uni}(A{:}S_A|S_{<i}) \\ &\overset{(e)}{\geq} \mathrm{I}(A; S_A) - \mathrm{I}(A; S_A|S_{<i}) \\ &\overset{(f)}{=} \mathrm{I}(A; S_A) \\ &\overset{(g)}{>} 0.\end{aligned}$$

Here, step (a) holds from equation 1, step (b) holds from Lemma 2 (Monotonicity under adversarial side information) for unique information since $S_i = (S_A, S_N)$, step (c) and (d) hold from equation 1, step (e) holds from equation 1, $\mathrm{Uni}(A{:}S_{<i}|S_A) \leq \mathrm{I}(A; S_A|S_{<i})$ and non-negativity of PID terms, step (f) holds because $S_A \subseteq S_{<i}$, and step (g) holds since the mutual information of the relevant part S_A is positive.

□

C Appendix to Implementation of Slider

C.1 Structure-Preserving Sample Generation

A key component of Slider is the *Structure-Preserving Sample Generation* procedure, which constructs multiple samples while preserving the underlying reasoning structure. For responses generated by QwQ-32B (Team, 2025) and DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025) on *GSM8K* (Cobbe et al., 2021) and *AIME2025* (Zhang & Team, 2025), before sample generation, we segment each reasoning trajectory and insert explicit `<step i>` tags to facilitate consistent step tracking across generated variants. We first segment the reasoning trajectory at double-line breaks when the resulting number of steps is fewer than 11. For trajectories with more than 10 steps, we instead use Qwen2.5-72B-Instruct (Team, 2024) as an automated reasoning-trace segmenter to produce more balanced and logically coherent steps. This prevents highly uneven segmentation, where an individual current step may be substantially shorter than the accumulated preceding reasoning. Because both are encoded into representations of the same dimensionality, such an imbalance may cause the much longer preceding reasoning to lose potentially relevant information during encoding. The automated segmenter therefore aims to produce more semantically coherent reasoning steps while preserving the underlying reasoning structure. We use the following prompt for this segmentation:

[System Prompt]
You are an expert reasoning-trace segmenter. Your only task is to split a raw reasoning trace into sequential logical segments or steps by inserting markup tags. CRITICAL REGULATION FOR

CONTENT PRESERVATION:

- Do NOT rewrite, paraphrase, summarize, or alter a single word of the original trace.
- Do NOT compress or skip text. Every sentence, equation, repetition, error, and alternative reasoning method must remain word-for-word exactly as it appeared originally.
- Keep all redundant loops, repetitive checks, and alternate thinking trajectories completely intact.

Formatting Rules:

- Output the full, unchanged text prefixed strictly by step markers like this:
  <step1>: [First block of original text, entirely unchanged]
  <step2>: [Second block of original text, entirely unchanged]
  ...
  <stepN>: [Final block of original text, entirely unchanged]
- Maximum number of steps: 10

Output must strictly contain ONLY the segmented text following this exact format without any conversational intro or markdown fences.
[User Prompt]

QUESTION: {question}
REASONING TRACE: {trace}
Task: Segment the reasoning trace into logical steps using the required format. Do NOT modify, delete, or rephrase any text. Return ONLY the segmented trace.

For the redundancy class of the *PRMBench* (Song et al., 2025) dataset, the reasoning steps are already segmented. We prepend `<step i>` tags to each step to format the input text for sample generation. Next, we generate 50 numerical variants of each problem and its corresponding reasoning trajectory using GPT-4o-mini (OpenAI, 2024) for GSM8K and PRMBench, and GPT-5-mini (OpenAI, 2025b) for AIME2025 to accommodate its more challenging problems. The prompts require each variant to preserve the original reasoning structure and the inserted `<step i>` tags. This allows corresponding reasoning steps to remain aligned across generated samples, facilitating subsequent step-level information estimation. We use the following prompt for structure-preserving sample generation for GSM8K and PRMBench:

[System Prompt]

You are an expert synthetic math dataset generator.
Rules:

1. Change ONLY numerical values in the problem. Do NOT modify any other text.
2. Preserve the Chain-of-Thought structure EXACTLY:
   - Same number of steps where steps are given as: <step1>.... <step2> .... <stepN>....
   - Same step order
   - Same reasoning logic per step (even if repetitive, redundant, or unnecessary)
   - Preserve step formatting exactly as written in the input, including: <step1>.... <stepN>....
3. Return valid JSON only.

[User Prompt]

Given the reference data input:

Problem:
`{Original Question}`

Solution:
`{Original Reasoning Trace}`

Generate exactly 1 numerical variation of the reference problem and its corresponding solution.
Return exactly this JSON structure:

```
{
  "problem1": "<Insert problem variant 1>",
  "solution1": "<Insert full solution variant 1>",
  "answer1": "<Insert final numerical answer as number>"
}
```

Return VALID JSON ONLY.

Given the greater difficulty of AIME2025 problems, we adopt a two-stage generation process. First, we use the following prompt to generate valid numerical variants of the original problem.

[System Prompt]

Given a math problem, generate ONE numerical variant by changing only numerical values while preserving the original wording and problem structure.
Solve the generated problem carefully. If it is valid and solvable, end with boxed{your answer}. If it is invalid or unsolvable, end with boxed{no solution}.
Output:
<variant> [generated new problem] </variant>
<solution> [solution of the new problem] </solution>
You must end your response with boxed{your answer} every time!

[User Prompt]

Original problem: question Remember to box your final answer via boxed{your answer}.

Then, using the generated solvable problems, we use the following prompt to generate structure-preserving samples.

[System Prompt]

You are given (1) a reference math problem, (2) its reference reasoning trajectory, and (3) a numerical variant of that problem.
Solve the numerical variant by adapting the REFERENCE trajectory as conservatively as possible.
Requirements:

1. Change only numerical values, calculations, symbols, or words necessary for the changed numerical values.
2. Preserve the Chain-of-Thought structure EXACTLY:
- Same number of steps: <step1>... <step2>... <stepN>... - Same step order
- Same reasoning logic per step
- Same step formatting
3. Solve the numerical variant correctly.
4. Return only the aligned trajectory followed by boxed{your answer}.

[User Prompt]

REFERENCE PROBLEM: <question>
REFERENCE TRAJECTORY: <thinking_trajectory>
NUMERICAL VARIANT TO SOLVE: <variant_problem>

C.2 Sensitivity Analysis

A key component of SLIDER is the discretization step, for which we employ k-means clustering on the embeddings. To evaluate how sensitive the PID estimates are to the choice of cluster count, we compute PID values for Problem 3 in Fig. 6 using 10, 15, and 20 clusters (Table 2). The results show that increasing the number of clusters generally yields higher PID values, though the magnitude of this increase differs across components. Crucially, the overall trends remain stable: the patterns of redundancy, uniqueness, and synergy are largely preserved across different cluster choices. For example, Table 2 shows that in the row $S_{<4} - S_4$, uniqueness in S_4 consistently dominates across all cluster counts n_{cluster}. In contrast, for $S_{<5} - S_5$ and $S_{<6} - S_6$, redundant information remains the dominant component regardless of the number of clusters n_{cluster}.

Table 2: Sensitivity of PID components for $S_{<i}$ and S_i to the number of clusters.

#Clusters	Redundancy	Unique ($S_{<i}$)	Unique (S_i)	Synergy
		$S_{<4}$ and S_4		
10	0.3621	0.0846	**0.7101**	0.1181
15	0.4502	0.1319	**0.6737**	0.2166
20	0.5025	0.2378	**0.6656**	0.3353
		$S_{<5}$ and S_5		
10	**0.6284**	0.1178	0.3822	0.1515
15	**0.7101**	0.1486	0.4619	0.2233
20	**0.6985**	0.2426	0.5653	0.2418
		$S_{<6}$ and S_6		
10	**0.7851**	0.0907	0.4895	0.0713
15	**0.7736**	0.1507	0.5002	0.1382
20	**0.8389**	0.2268	0.5108	0.1692

D Appendix to Experiments

Datasets. We conduct our experiments on GSM8K (Cobbe et al., 2021), PRMBench (Song et al., 2025), and AIME2025 (Zhang & Team, 2025).

GSM8K (Cobbe et al., 2021) is a benchmark consisting of grade-school mathematical word problems that require multi-step reasoning. It contains approximately 7.47K training and $1.32k$ test mathematics word problems. Solving each problem typically requires multiple steps of numerical reasoning, including intermediate arithmetic operations, making GSM8K a widely used benchmark for evaluating the chain-of-thought reasoning capabilities of language models. We use GSM8K to analyze step-wise reasoning patterns, study the relationship between RRI and reasoning length, and conduct our RRI-guided fine-tuning experiments.

PRMBench (Song et al., 2025) is a process-level reasoning benchmark designed to evaluate the ability to identify errors occurring at individual steps of a reasoning trajectory. The benchmark contains

fine-grained annotations covering different categories of reasoning errors. In our experiments, we specifically focus on the *redundancy* category to evaluate the ability of RRI to identify repetitive reasoning steps. This class has around 750 problems. We treat steps annotated as redundant as the positive class and non-redundant steps as the negative class when computing the evaluation metric.

AIME2025 (Zhang & Team, 2025) is a benchmark containing 30 problems from the 2025 American Invitational Mathematics Examination I and II. Compared with GSM8K, its problems are more complex and require greater algebraic manipulation, geometric insight, and symbolic reasoning.

Spearman Correlation. We use Spearman's rank correlation coefficient (Zar, 2005) to quantify the relationship between reasoning repetitiveness and reasoning length. For two variables X and Y, Spearman's correlation is defined as the Pearson correlation between their ranks:

$$\rho = \frac{\mathrm{Cov}(\mathrm{rank}(X), \mathrm{rank}(Y))}{\sigma_{\mathrm{rank}(X)}\sigma_{\mathrm{rank}(Y)}}, \tag{5}$$

where $\mathrm{rank}(X)$ and $\mathrm{rank}(Y)$ denote the rank-transformed variables, $\mathrm{Cov}(\cdot,\cdot)$ denotes covariance, and σ denotes standard deviation. The value ranges from -1 (perfect negative monotonicity) to 1 (perfect positive monotonicity), with 0 indicating no monotonic relationship.

D.1 PID MEASURES UNDER SYNTHETIC NOISE



(a) Erroneous S_4 **(b) Erroneous S_5** **(c) Reasoning Steps**

Figure 10: Interpretable trends of PID measures under errors for word problem: (a,b) Under errors, uniqueness $\mathrm{Uni}(S_i)$ corresponding to a correct step increases with p, reflecting that correct steps contribute increasingly distinct information relative to their erroneous counterparts. Conversely, $\mathrm{Uni}(S_i)$ associated with the erroneous step decreases as error probability p increases. (c) Reference example for the reasoning steps in concise form.

We consider a word problem from the GSM8K dataset: *"Darren decides to do body exercises for a whole week. He does $\{x\}$ pushups, $\{y\}$ squats, and $\{z\}$ dumbbell presses on the first day. On the second day, he does 20 more pushups than on the first day, ten fewer squats, and doubles the number of dumbbell presses. What's the total count of the activities he's done in the two days?"* We prompt DeepSeek-R1-Distill-Qwen-32B to generate a model response for the problem and retain the text preceding the `</think>` tag as the reasoning trajectory for our analysis. We manually segment each reasoning trajectory at double-newline (`\n\n`) boundaries. We then generate 1,000 problem instances by uniformly sampling $x, y \in [10, 500]$ and $z \in [10, 100]$. To study the effect of reasoning errors, we perturb the total push-up calculation in step S_4 with probability p by adding uniformly sampled integer noise from $[0, 1000]$. We then analyze the resulting PID measures using SLIDER, as shown in Fig. 10. Consistent with Theorem 1, under errors, the uniqueness $\mathrm{Uni}(S_i)$ corresponding to a correct step increases with p, indicating that correct steps provide increasingly distinct information relative to their erroneous counterparts. In contrast, the uniqueness associated with the erroneous step decreases as the error probability p increases. Fig. 10(b) shows that introduced errors propagate toward the final step S_5. For that step, the drop in $\mathrm{Uni}(S_5)$ is significant since this step directly includes the final answer A. Thus, SLIDER can help identify incorrectness in reasoning.

D.2 STEP-WISE REPETITIVENESS

Setup. To assess whether Step-RRI captures true repetitiveness in a reasoning trajectory, we consider three word problems from the GSM8K dataset. For each problem, we first obtain reasoning trajectories

from DeepSeek-R1-Distill-Qwen-32B and QwQ-32B. We then generate 1,000 structure-preserving samples by parameterizing selected numerical values in the problem and updating the corresponding quantities in the reasoning trajectory while preserving its reasoning structure. For Problem 1, 'Alisa biked {x} miles per hour for 4.5 hours. Stanley biked at {y} miles per hour for 2.5 hours. How many miles did Alisa and Stanley bike in total?", we independently sample x and y uniformly from the integer range $[10, 100]$ and update the corresponding calculations in the trajectory. For Problem 2, ''Anthony had x pencils. He gave 1/2 of his pencils to Brandon, and 3/5 of the remaining pencils to Charlie. He kept the remaining pencils. How many pencils did Anthony keep?", and for Problem 3, "James runs x miles a day for 5 days a week. If he runs 10 miles an hour, how many hours does he run a week?", we similarly sample x uniformly from the integer range $[10, 1000]$ and update all dependent quantities in the reasoning trajectory. Then, we apply SLIDER to get the RRI measures.

Step-RRI for DeepSeek-R1-Distill-Qwen-32B. Fig. 11 shows the step-wise PID decomposition together with the resulting Step-RRI values (with $\eta = 1$) for reasoning trajectories generated by DeepSeek-R1-Distill-Qwen-32B. For Problems 1 and 2, Step-RRI remains negative or close to zero across the reasoning steps, indicating that the current steps primarily contribute new or complementary answer-relevant information relative to the preceding reasoning. In Problem 3, Step-RRI becomes positive at the final step, indicating that the current step increasingly recovers answer-relevant information already established by the preceding reasoning. These examples illustrate how Step-RRI summarizes the relative balance of redundant, unique, and synergistic information to localize true repetitive reasoning within a trajectory.

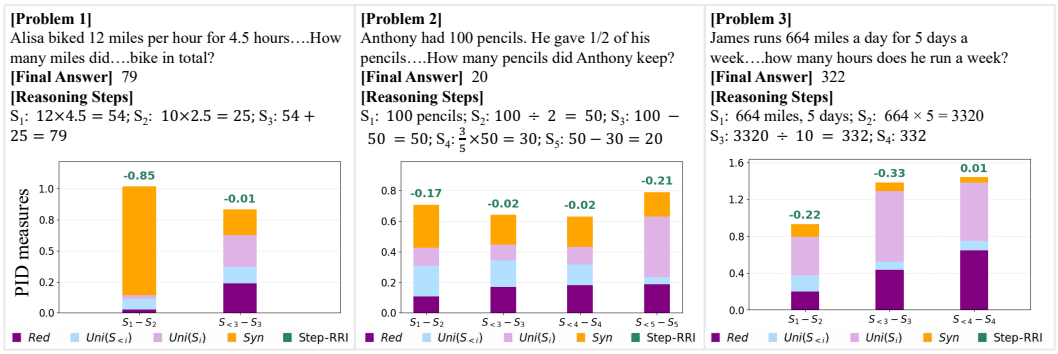

Figure 11: **Step-RRI for DeepSeek-R1-Distill-Qwen-32B reasoning trajectories.** Step-wise PID components and the corresponding Step-RRI values for three representative GSM8K reasoning trajectories. Negative Step-RRI indicates that the current step contributes primarily new or complementary answer-relevant information, whereas positive Step-RRI indicates increased repetitiveness with respect to the preceding reasoning.

Next, we provide one representative full reasoning trajectory for each model, QwQ-32B and DeepSeek-R1-Distill-Qwen-32B. These examples show the original natural-language reasoning traces used in our experiments.

[Question]

James runs 664 miles a day for 5 days a week. If he runs 10 miles an hour how many hours does he run a week?

[Reasoning of QwQ-32B]

1. Okay, let's see here. James runs 664 miles a day for 5 days a week. And we need to find out how many hours he runs each week, given that his speed is 10 miles per hour.
2. First, I think I should figure out the total distance he runs in a week. Since he does 664 miles each day and runs 5 days a week, I can multiply 664 by 5. Let me do that: 664 times 5 equals... 3320. So he runs 3320 miles in a week.

3. Now, the question is asking for the time he spends running, not the distance. His speed is 10 miles per hour. Hmm, so if he runs at 10 mph, then the time taken would be distance divided by speed. Right? Because time equals distance over speed.
4. So the total time per week would be the total miles divided by his speed. That would be 3320 miles divided by 10 miles per hour. Let me calculate that: 3320 divided by 10 is 332.0. So that gives 332.0 hours.
5. Wait, let me double-check. If he runs 664 miles a day at 10 mph, how long does each day's run take? For one day: 664 miles divided by 10 mph would be 66.4 hours per day. Then, multiplying that by 5 days: 66.4 times 5 equals 332.0 hours. Yep, same answer. So that seems right.
6. I don't think there's any trick here. Maybe I'm missing something? Like, does he take rest days or anything? The problem says he runs 5 days a week, so the other two days aren't counted. The calculation for total miles is correct. Then dividing by speed gives total hours. Both methods give 332.0, so I think that's solid. Alright, the answer should be 332.0 hours per week.

[Reasoning of DeepSeek-R1-Distill-Qwen-32B]

1. First, I need to determine how many miles James runs each week. He runs 664 miles per day and does this for 5 days.
2. So, multiplying the daily distance by the number of days gives me the weekly total: 664 miles/day * 5 days = 3320 miles/week.
3. Next, since James maintains a running speed of 10 miles per hour, I can calculate the total time he spends running each week by dividing the total weekly distance by his speed: 3320 miles / 10 miles/hour = 332.0 hours.
4. Therefore, James runs a total of 332.0 hours each week.

Baselines. We adapt two complementary measures as baselines: Cosine Similarity, which captures semantic overlap between reasoning steps, and InfoGain, which measures the reduction in predictive uncertainty contributed by a reasoning step. Intuitively, if the current reasoning step repeats or conveys the same information as the preceding reasoning, we expect it to exhibit high cosine similarity due to greater semantic overlap, while providing little additional information and therefore exhibiting lower information gain.

Cosine Similarity. As a similarity-based baseline, we measure the semantic overlap between the current reasoning step S_i and the preceding reasoning $S_{<i}$. We encode S_i and $S_{<i}$ separately using all-MiniLM-L6-v2 (Reimers & Gurevych, 2019) and normalize the resulting embeddings. Their cosine similarity is then given by

$$\mathrm{CosSim}(S_i, S_{<i}) = \mathbf{e}_{S_i}^{\top} \mathbf{e}_{S_{<i}}, \tag{6}$$

where $\mathbf{e}_{S_i}$ and $\mathbf{e}_{S_{<i}}$ denote the normalized embeddings of the current and preceding reasoning, respectively. A high cosine similarity indicates substantial semantic overlap with previously established reasoning and therefore provides a natural proxy for repetitiveness.

InfoGain. As an information-gain baseline, we adapt the information-gain principle of Yong et al. (2025), where an informative reasoning step is expected to reduce predictive uncertainty. At each step S_i, we use QwQ-32B to obtain the top-K next-token probabilities conditioned on the question Q and the reasoning prefix $S_{1:i}$. Since only the top-K probabilities are retained, we renormalize them as

$$\tilde{P}(v_k \mid Q, S_{1:i}) = \frac{P(v_k \mid Q, S_{1:i})}{\sum_{j=1}^{K} P(v_j \mid Q, S_{1:i})}, \tag{7}$$

where v_k denotes the k-th most probable next token. We then compute the entropy of the normalized top-K distribution as

$$H_i = -\sum_{k=1}^{K} \tilde{P}(v_k \mid Q, S_{1:i}) \log_2 \tilde{P}(v_k \mid Q, S_{1:i}), \tag{8}$$

and define the information gain associated with S_i as

$$\Delta I_i = H_{i-1} - H_i. \tag{9}$$

We use $K = 10$ in our experiments. A large positive ΔI_i indicates that incorporating S_i substantially reduces next-token uncertainty, whereas a small or negative ΔI_i suggests that the step provides limited additional predictive information. We therefore use low information gain as a proxy for repetitive reasoning, as a repetitive step is expected to contribute limited new predictive information.

D.3 TRAJECTORY-RRI VERSUS REASONING LENGTH

Setup. For our analysis on GSM8K (Cobbe et al., 2021), we first identify the problems from the training set for which both QwQ-32B and DeepSeek-R1-Distill-Qwen-32B reach the correct final numerical answer. From this common set of correctly solved problems, we sample 1,000 problem indices and collect the corresponding responses from both models, ensuring that our analysis focuses on repetitiveness among successful reasoning trajectories. Each model response consists of a thinking trajectory followed by the final response, separated by the `</think>` tag. We extract the text preceding this tag as the reasoning trajectory and use only this portion throughout the analysis. For GPT-4.1, we use the same 1,000 problems and treat its generated responses as the corresponding reasoning trajectories. For AIME2025 (Zhang & Team, 2025), we select problems for which all three models produce the correct final answer.

We apply SLIDER independently to the extracted reasoning trajectories of each model to estimate Step-RRI for individual reasoning steps and subsequently compute Trajectory-RRI for each problem. To analyze the relationship between Trajectory-RRI and the true reasoning length, we group the trajectories into quantile bins according to their reasoning length. Within each bin, we report the average Trajectory-RRI and the average reasoning length. For GSM8K, we use 50 bins, and for AIME2025, we use 7 bins. Importantly, the reasoning length is computed from the original reasoning trajectory generated by the LLMs, rather than from the structure-preserving numerical variants used by SLIDER for estimating the PID components. Thus, the reported reasoning length directly reflects the token counts of the original model responses.

D.4 TRAJECTORY-RRI GUIDED DATA SELECTION

Table 3: Test accuracy on GSM8K across different average Trajectory-RRI levels.

Avg. Trajectory-RRI	Accuracy (%)
3.50	88.55
3.34	89.31
3.19	89.46
3.02	89.31
2.84	88.78
2.42	89.16
2.04	88.48
1.88	87.41
1.71	89.16

Trajectory-RRI-Guided Dataset Construction. We construct fine-tuning datasets systematically with varying levels of average Trajectory-RRI using 1,000 problems from GSM8K. For each problem, we obtain two candidate reasoning trajectories: one generated by QwQ-32B and the other by DeepSeek-R1-Distill-Qwen-32B. We compute the Trajectory-RRI of each candidate trajectory using SLIDER, resulting in a pair of trajectories with potentially different levels of repetitive reasoning for each problem. To control the average Trajectory-RRI of the resulting training data while keeping the set of problems fixed, we construct each fine-tuning dataset by selecting exactly one of the two candidate trajectories for every problem. Specifically, for each problem, we identify the candidate trajectory with the higher Trajectory-RRI and select it with probability p. This procedure yields a training set containing 1,000 problem with different average Trajectory-RRI for each value of p. For example, when $p = 0$, the lower-Trajectory-RRI trajectory is selected for every problem, whereas $p = 1$ selects the higher-Trajectory-RRI trajectory for every problem. Intermediate values of p produce mixtures of the two, allowing us to systematically vary the average Trajectory-RRI of the training data without changing the underlying set of GSM8K problems. These datasets are subsequently used to fine-tune the same base model under an identical training configuration, allowing us to examine the relationship

between the repetitiveness of the training trajectories defined by our measure, Trajectory-RRI, and the actual reasoning behavior of the resulting fine-tuned models.

LLM-as-a-judge efficiency score. We use Qwen2.5-72B-Instruct (Team, 2024) as an LLM judge to evaluate the reasoning efficiency of the fine-tuned models on the test set. Specifically, the judge assesses each generated reasoning trajectory per question for conciseness, loops, and redundant reasoning. For each fine-tuned model, we report the overall efficiency as the average efficiency score across all examples in the test set. We use the following prompt for this evaluation:

[System Prompt]
You are an expert AI judge evaluating the efficiency of reasoning traces. Focus strictly on conciseness, loops, and redundancy. Output ONLY raw JSON.

[User Prompt]
Evaluate the efficiency and repetitiveness of this reasoning trace. Ignore whether the final answer is right or wrong; focus entirely on the pacing and overhead of the logic.

Question: `{Original Question}`

Trace: `{Reasoning Trace}`

Score Guidelines (0–10 Efficiency Score):

- **10:** Maximally concise. Direct, optimal path with zero bloat or repetition.
- **7–9:** Highly efficient, but contains minor phrasing redundancy or slight over-explanation.
- **4–6:** Noticeably inefficient. Contains repetitive steps, circular logic, or heavy fluff.
- **1–3:** Extremely bloated, highly repetitive, or stuck in reasoning loops.
- **0:** Total failure of efficiency (e.g., endless repeating strings or infinite loops).

Output Format (JSON only, no markdown):

```
{
  "efficiency_score": 0-10,
  "reason": "One-sentence analysis of redundancy and pacing."
}
```

Fine-tuning setup. We fine-tune Qwen2.5-7B-Instruct using Low-Rank Adaptation (LoRA) (Hu et al., 2021). We use a LoRA rank of $r = 32$, a scaling factor of $\alpha = 32$, and a LoRA dropout of 0.05. LoRA adapters are applied to all attention and MLP projection layers. We train each model for 10 epochs using the AdamW optimizer with a learning rate of 1×10^{-5}, weight decay of 1×10^{-4}, and momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.95$. We use a cosine learning-rate scheduler with a warmup ratio of 0.05. The per-device batch size is set to 1, with gradient accumulation of 8 steps, resulting in an effective global batch size of 32, where the number of GPUs used for training is 4. We set the maximum sequence length to 8192 tokens and use bfloat16 precision.

For distributed training, we use Fully Sharded Data Parallel (FSDP) with full parameter sharding and automatic module wrapping. We additionally enable gradient checkpointing to reduce memory consumption. Model checkpoints are saved after each training epoch, and no intermediate evaluation is performed during training. All models are trained using the same optimization and fine-tuning configuration to ensure a controlled comparison across datasets with average Trajectory-RRI values.

D.5 Runtime and Resource Usage

All experiments are conducted using NVIDIA RTX 6000 GPUs with CUDA 12.8. For GSM8K, SLIDER takes an average of approximately 181 seconds per sample for QwQ-32B and 82 seconds per sample for DeepSeek-R1-Distill-Qwen-32B. For PRMBench, the average runtime of SLIDER is approximately 38 seconds per sample. For AIME2025, the average runtimes are approximately 609 and 320 seconds per sample for QwQ-32B and DeepSeek-R1-Distill-Qwen-32B, respectively.