Valid $\neq$ Necessary: Diagnosing Latent Inefficiency in Chain-of-Thought

Anonymous ACL submission

Abstract

Chain-of-Thought (CoT) prompting has significantly advanced the reasoning capabilities of Large Language Models (LLMs), yet it often incurs substantial computational costs due to "over-reasoning"—the generation of redundant, verbose, or irrelevant steps. While existing reasoning step evaluators effectively detect logical fallacies and factual errors, our analysis reveals a critical blind spot: they fail to penalize "valid but inefficient" reasoning steps that inflate token usage without contributing to the solution. To systematically diagnose this limitation, we introduce **RIV-GSM8K**, a diagnostic benchmark injected with five distinct types of inefficiencies, including circular reasoning and excessive decomposition. Diagnostic experiments reveal that state-of-the-art evaluators struggle to distinguish these inefficiencies from necessary reasoning. To address this, we propose **CAID** (Context-Aware Information Density), a training-free metric grounded in information theory that effectively identifies low-utility steps. To validate the metric's practical utility, we apply it within **PACE**, a post-hoc compression strategy. Empirical results on GSM8K, StrategyQA, and ARC-Challenge demonstrate that PACE reduces token consumption by 31–53% while maintaining accuracy, confirming that CAID successfully distills informational "froth" from reasoning chains without compromising deductive validity.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in complex reasoning tasks, largely driven by the Chain-of-Thought (CoT) prompting strategy (Wei et al., 2022). By decomposing complex problems into intermediate steps, LLMs can bridge the gap between question and answer. However, this performance gain often comes at the cost of inference efficiency. Recent studies indicate that LLMs exhibit a tendency towards **"over-reasoning"**—generating verbose explanations, repetitive statements, or contextually irrelevant details that inflate computational costs without adding deductive value (Turpin et al., 2023; Wang et al., 2023; Chiang and Lee, 2024).

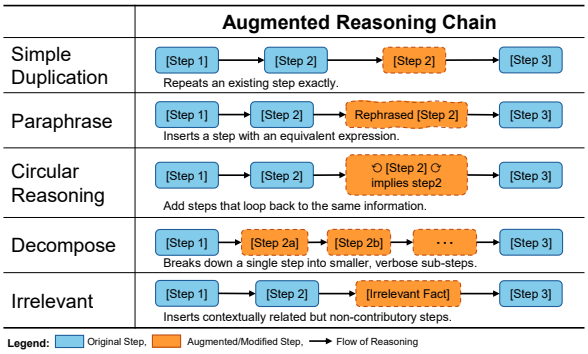

Figure 1: Taxonomy of reasoning inefficiencies in RIV-GSM8K. The diagram illustrates how five distinct types of redundant steps are synthetically injected into the reasoning chain to simulate valid but dispensable "froth."

To assess and improve reasoning quality, various Process Reward Models (PRMs) and reasoning step evaluators have been proposed, such as ReasonEval and Math-Shepherd (Xia et al., 2025; Wang et al., 2024). These methods have primarily focused on **correctness** and **logical validity**, aiming to penalize factual errors or hallucinations. While effective for validity, our analysis reveals a critical blind spot in these state-of-the-art evaluators: they struggle to distinguish *inefficiency* from *reasoning*. Specifically, they often assign high scores to "valid but redundant" steps—such as excessive decomposition or circular logic—merely because these steps remain factually true and linguistically coherent. Consequently, the fundamental question in current reasoning evaluation is limited to "Is this step true?", neglecting the equally important dimension: **"Is this step truly necessary?"**

In this paper, we aim to shift the evaluation paradigm from verifying correctness to optimizing **information density**. To systematically diagnose the limitations of current evaluators, we first

introduce **RIV-GSM8K**, a diagnostic benchmark derived from GSM8K (Cobbe et al., 2021). As illustrated in Figure 1, RIV-GSM8K is synthetically injected with five distinct types of inefficiencies ranging from simple duplication to subtle circular reasoning. Using this benchmark, we empirically demonstrate that existing validity-focused PRMs are largely insensitive to explicit redundancy.

To address this gap, we propose **CAID** (Context-Aware Information Density), a novel unsupervised metric grounded in information theory that evaluates reasoning steps based on local novelty, global goal alignment, and information density. Unlike previous metrics, CAID effectively identifies informational "froth" within reasoning chains. To **empirically validate the diagnostic precision** of this metric, we present **PACE** (Pruning And Compression for Efficiency), a post-hoc compression strategy. Notably, PACE goes beyond simple deletion; it identifies *latent inefficiency* in reasoning chains and predominantly **compresses** verbose steps (Merge) while pruning irrelevant ones, ensuring that the process remains logically sound but significantly more compact.

Our main contributions are summarized as follows:

- We uncover the "efficiency blind spot" of current reasoning evaluators through **RIV-GSM8K**, a stress-test benchmark designed to diagnose specific types of reasoning inefficiencies.
- We propose **CAID**, an interpretable, training-free metric that quantifies the informational utility of reasoning steps, distinguishing essential logic from redundant "froth."
- We validate our approach via **PACE**, which reduces token consumption by 31–53% across arithmetic, commonsense, and scientific reasoning tasks without compromising accuracy. This **serves as empirical evidence** for the existence of significant *latent inefficiency* in standard CoT reasoning, demonstrating that high-quality reasoning data can be far more compact than previously assumed.

2 Related Work

2.1 Over-reasoning and Inference Efficiency

While Chain-of-Thought (CoT) prompting (Wei et al., 2022) has revolutionized LLM reasoning, it has also introduced the challenge of inference inefficiency. Recent studies highlight that LLMs suffer from **"over-reasoning"**—a tendency to generate verbose explanations, repetitive loops, or contextually irrelevant details (Turpin et al., 2023; Chen et al., 2024; Chiang and Lee, 2024). Jiang et al. (2023) demonstrated that such redundant context not only inflates computational costs but can also degrade performance by distracting the model.

To mitigate computational costs, token pruning methods such as H2O (Zhang et al., 2023) and Learned Token Pruning (Kim et al., 2022) have been proposed. These approaches reduce sequence length by discarding tokens with low attention scores during inference. However, they operate at the **token level**, focusing primarily on latency reduction (KV cache optimization) rather than the informational quality of the content. In contrast, our strategy, PACE, addresses inefficiency at the **semantic step level**. Rather than competing with runtime token pruners, we focus on identifying and filtering out *latent inefficiency* embedded in the reasoning chain itself. This semantic compression serves a complementary role, potentially enhancing the quality of training data for future models.

2.2 Reasoning Step Evaluation Methods

Moving beyond outcome-based evaluation, stepwise evaluation methods have emerged to provide granular supervision. Process Reward Models (PRMs) like PRM800K (Lightman et al., 2024) and Math-Shepherd (Wang et al., 2024) train verifiers to distinguish between correct and incorrect reasoning paths. More recently, **ReasonEval** (Xia et al., 2025) has attempted to assess reasoning quality beyond mere validity, incorporating scores for redundancy and clarity.

However, existing methods face two critical limitations. First, standard PRMs primarily focus on **correctness verification**, penalizing factual errors but often rewarding valid yet inefficient steps (e.g., stating the obvious). Second, while ReasonEval attempts to detect redundancy, it relies on supervised training with human-annotated data. **Crucially, human annotators often prioritize validity over conciseness, leading to sparse and ambiguous labels regarding what constitutes "unnecessary."** In contrast, our approach utilizes **RIV-GSM8K**, where inefficiencies are synthetically injected based on a distinct taxonomy. This allows for a deterministic evaluation of detection capability.

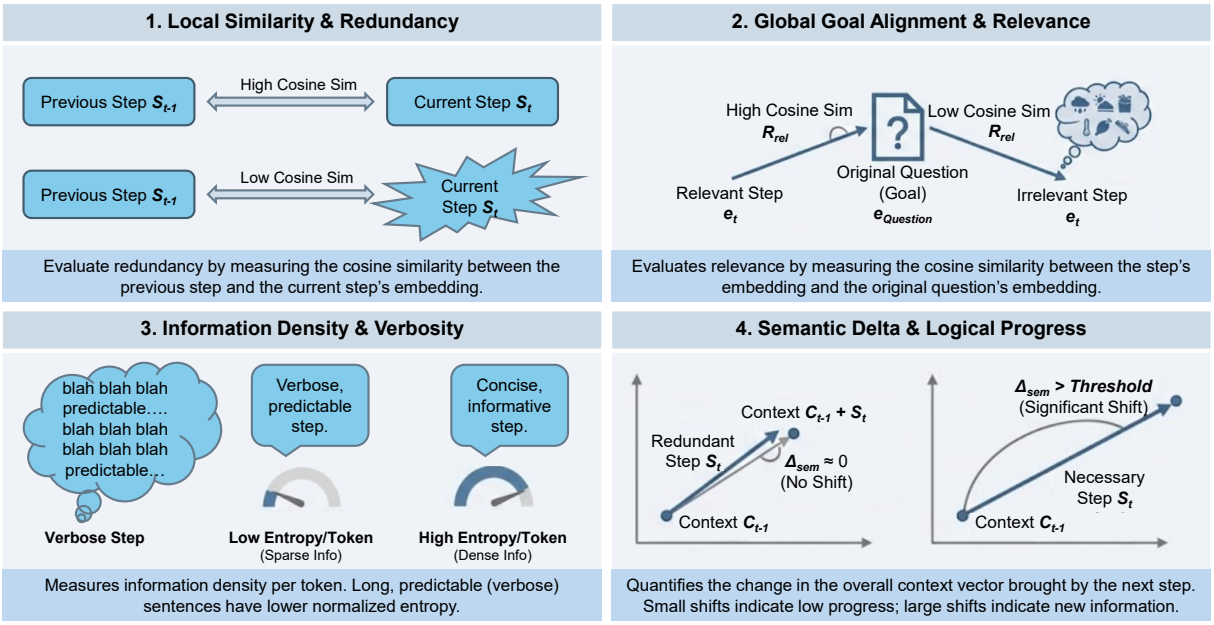

Figure 2: Conceptual overview of the four information-theoretic indicators in CAID. (1) Local Similarity detects surface-level redundancy via adjacent step comparison. (2) Global Goal Alignment filters irrelevant steps by measuring drift from the original question. (3) Information Density identifies verbose, low-entropy steps using length-normalized perplexity. (4) Semantic Delta quantifies the vector shift in the context, ensuring substantial logical progress.

2.3 Redundancy Detection Metrics

A few studies have addressed the evaluation of text redundancy. ROSCOE (Golovneva et al., 2023) proposes a suite of metrics for text generation, using semantic similarity (e.g., SimCSE) to detect repetitions. Similarly, LI et al. (2023) introduced Selective Context, which uses self-information (perplexity) to prune input prompts.

These approaches, however, are limited in the context of multi-step reasoning. ROSCOE relies heavily on **surface-level similarity** and often requires a **reference text** (golden truth) for accurate evaluation. **However, relying on references is problematic because, as our findings suggest, even human-written golden truths often contain significant inefficiency.** Furthermore, metrics like Selective Context measure static information content but do not account for the logical flow. CAID advances these by introducing **Semantic Delta** and **Goal Alignment**, which measure not just the static information of a step, but its dynamic contribution to the logical *progress* towards the solution without requiring a reference chain.

3 Methodology

We propose a comprehensive framework to diagnose, measure, and mitigate reasoning inefficiency, shifting the evaluation paradigm from validity-centric to **efficiency-aware**. Our approach operates in three stages: (1) **Diagnosis:** We introduce **RIV-GSM8K** to expose the "efficiency blind spots" of current evaluators. (2) **Measurement:** We propose **CAID**, an information-theoretic metric designed to detect valid but inefficient steps without human annotations. (3) **Validation:** We apply **CAID** within the **PACE** strategy to empirically validate its diagnostic precision through post-hoc chain compression.

3.1 RIV-GSM8K: Diagnosing Inefficiency

A critical challenge in evaluating reasoning efficiency is the lack of an absolute ground truth for "necessary" steps. Human annotations are often subjective and sparse. To address this, we construct **RIV-GSM8K**, a diagnostic benchmark derived from GSM8K (Cobbe et al., 2021) based on a **"relative inefficiency"** paradigm. By synthetically injecting controlled perturbations into baseline reasoning chains (Golden CoT), we create steps that are *strictly less efficient* than the original ones while remaining factually correct. This allows for a deterministic evaluation of whether an evaluator can distinguish "necessary" logic from "redundant" froth.

Construction Process. The overall construction procedure is detailed in Appendix A. We employ a hybrid generation approach to balance diversity

and control. **Simple Duplication** is generated via rule-based repetition to serve as a baseline. For complex types (**Paraphrase, Decompose, Circular, Irrelevant**), we utilize **GPT-4o** as the generator $\mathcal{G}$. Crucially, we enforce strict constraints during generation to ensure that the injected steps are *valid but inefficient*:

- **No New Progress:** The generated step must not advance the reasoning state beyond the target step S_t. It must remain logically stationary or redundant.
- **Contextual Coherence:** Particularly for the *Irrelevant* type, the generator is prompted to produce sentences that are mathematically true and linguistically blended with the current context, but strictly "non-contributory" to the solution logic.

The detailed prompts used for each perturbation type are provided in Appendix B. In total, we constructed 7,473 samples containing over 20,000 augmented steps (see Appendix A.2 for detailed statistics).

As illustrated in Figure 1, we define five distinct taxonomies of reasoning inefficiency designed to mimic the "over-reasoning" behaviors of LLMs (visual examples in Appendix C). We first introduce **Simple Duplication** and **Paraphrase** to serve as baselines for detecting lexical and semantic redundancy, respectively. To address informational dilution, we define **Decompose**, which models the excessive fragmentation of a single logical step into multiple *low-density* micro-steps. Finally, we simulate stalled logical progress and deviations from the problem objective through **Circular Reasoning** (self-referential verification) and **Irrelevant** steps (non-contributory facts), thereby testing the model's sensitivity to *global goal alignment*.

Quality Verification via Human Evaluation. To ensure the validity of the synthetically generated inefficiencies, we conducted a human evaluation on the four GPT-4o-generated types (Paraphrase, Decompose, Circular, Irrelevant), excluding the rule-based Simple Duplication. We randomly sampled 30 instances per type (120 total) and verified adherence to the generation constraints, specifically checking for *Logical Equivalence* and the *No New Progress* rule. The evaluation revealed a high overall success rate. The *Irrelevant* and *Circular Reasoning* types achieved near-perfect validity (29/30 and 30/30, respectively). The *Decompose* type showed a slightly higher failure rate (4 failures), primarily due to "atomic" steps (e.g., simple equations like $3 \times x = 30$) that are inherently indivisible, leading the model to either hallucinate details or mistakenly include future reasoning steps. Despite these edge cases, the vast majority of generated steps correctly introduced the intended inefficiency without altering the ground truth logic.

3.2 CAID: Context-Aware Information Density

Current evaluators primarily focus on factual validity, often overlooking the inefficiencies modeled in RIV-GSM8K. To bridge this gap, we propose **CAID**, a reference-free, unsupervised metric designed to quantify the *informational utility* of a reasoning step. Rather than relying on heuristics tailored to specific error types, CAID is grounded in **information theory**, evaluating steps based on their contribution to the reasoning process relative to their length and context. CAID integrates four core indicators, as conceptualized in Figure 2:

1. Local Similarity (Redundancy). To capture redundancy at the surface and semantic levels, we measure the cosine similarity between the current step S_t and its immediate predecessor S_{t-1} using a lightweight encoder $\mathcal{E}$ (e.g., MiniLM) as $\mathcal{M}_{sim}(S_t) = \text{CosSim}(\mathcal{E}(S_t), \mathcal{E}(S_{t-1}))$. A high similarity score indicates that the step provides negligible new information compared to the immediate history, signaling potential repetition or inefficient paraphrasing.

2. Global Goal Alignment (Relevance). Reasoning steps must remain relevant to the problem objective. We measure the alignment between the step and the original question Q by computing $\mathcal{M}_{rel}(S_t) = \text{CosSim}(\mathcal{E}(S_t), \mathcal{E}(Q))$. A significantly low alignment score suggests that the step has drifted from the core problem goal, identifying content that may be linguistically coherent but contextually irrelevant.

3. Information Density (Verbosity). Efficient reasoning should convey maximum information with minimum tokens. We define density as the length-normalized perplexity using a causal language model $\mathcal{M}$ (e.g., GPT-2):

$$\mathcal{M}_{density}(S_t) = \frac{\log(\text{PPL}_{\mathcal{M}}(S_t))}{\text{Length}(S_t)} \quad (1)$$

Steps with exceptionally low density imply a lack of information content relative to their verbosity. This metric effectively penalizes steps that excessively decompose simple logic into multiple, low-entropy micro-steps, detecting "informational froth."

4. Semantic Delta (Logical Progress). Crucially, a valid reasoning step must advance the logical state toward the solution. We define Semantic Delta as the shift in the context vector induced by adding S_t:

$$\mathcal{M}_{delta}(S_t) = 1 - \text{CosSim}(\mathcal{E}(C_{t-1}), \mathcal{E}(C_{t-1} \oplus S_t)) \quad (2)$$

A near-zero delta implies that S_t fails to update the semantic state of the context, characterizing tautologies or circular reasoning. As reasoning naturally converges toward the final answer, the marginal information gain of each subsequent step often diminishes. To address this, we employ an **adaptive decaying threshold** $\tau_\delta(t) = \tau_{base} \cdot \lambda^t$, which dynamically adjusts sensitivity based on the step position.

Decision Logic. CAID aggregates these indicators to classify steps into an action set $\mathcal{A} = \{\text{PRUNE}, \text{MERGE}, \text{KEEP}\}$ via a hierarchical decision process. Specifically, steps exhibiting high redundancy ($\mathcal{M}_{sim}$) or low relevance ($\mathcal{M}_{rel}$) are considered strictly unnecessary and flagged for **PRUNE**. In contrast, steps that are relevant but exhibit low information density ($\mathcal{M}_{density}$) or marginal logical progress ($\mathcal{M}_{delta}$) are targeted for **MERGE**, an action that preserves the underlying logic while condensing verbose or fragmented expressions. This multi-view approach ensures robust optimization, selectively removing distractions while refining the density of valid reasoning.

3.3 Application: Validating CAID via PACE

To empirically validate the diagnostic precision of CAID, we introduce **PACE** (Pruning And Compression for Efficiency) as a **post-hoc compression strategy**. Our primary objective here is not to accelerate real-time inference, but to serve as a diagnostic probe: demonstrating that the steps flagged by CAID are indeed dispensable "froth" that can be distilled without breaking the deductive chain.

PACE operates in a *Generate-then-Refine* pipeline. Based on the classification from CAID, we apply three actions:

- **PRUNE:** Removes steps flagged as high redundancy ($\mathcal{M}_{sim}$) or low relevance ($\mathcal{M}_{rel}$).
- **MERGE:** Compresses steps exhibiting *latent inefficiency* (valid but verbose/fragmented) using an LLM re-writer. To prevent semantic drift or information overload, we enforce two safety constraints before merging step S_t into the accumulated step S'_{last}:
  1. **Semantic Consistency:** $\text{CosSim}(\mathcal{E}(S'_{last}), \mathcal{E}(S_t)) \geq \tau_{merge}$. We ensure the new content is logically compatible with the current context.
  2. **Information Saturation:** $\mathcal{I}(S'_{last}) \leq \tau_{max}$. We prevent merging if the current step is already information-dense, avoiding readability loss.

  If constraints are violated, the merge is halted, and a new step is initiated.
- **KEEP:** Retains steps essential for logical progress.

Addressing the "Trivial Accuracy" Concern. One might assume that maintaining accuracy is trivial if the final conclusion step is preserved. However, in our evaluation, we construct the prompt using the compressed chain C' **excluding the final numerical answer** (i.e., the "### Result" token). The model is required to *regenerate* the final answer solely based on the logic provided in C'. Since reasoning chains are causal, removing or altering a necessary intermediate step would break the logical dependency required to derive the correct solution. Therefore, the fact that PACE maintains accuracy with significantly fewer tokens serves as robust evidence that the removed **or compressed** steps were indeed functionally redundant and that CAID correctly identified the core reasoning path.

4 Experiments

4.1 Experimental Setup

Datasets. We employ diverse benchmarks to conduct a two-stage evaluation, assessing both diagnostic sensitivity and practical compression utility.

- **RIV-GSM8K:** A controlled diagnostic set used to measure the sensitivity of metrics to explicit, synthetically injected inefficiencies (Section 3.1).

Model	simple_duplication		paraphrase		decompose		circular_reasoning		irrelevant	
	Aug PR (↓)	Gold PR	Aug PR (↓)	Gold PR	Aug PR (↓)	Gold PR	Aug PR (↓)	Gold PR	Aug PR (↓)	Gold PR
ReasonEval 7B	0.6555	**0.9897**	0.7338	**0.9867**	0.7917	**0.9853**	0.4809	**0.9746**	0.1509	**0.9571**
ReasonEval 34B	0.7779	0.9558	0.7251	0.9533	0.7604	0.9571	0.3762	0.9345	**0.0331**	0.9118
ThinkPRM 1.5B	0.6264	0.7342	0.6821	0.7486	0.7187	0.7663	0.7179	0.7810	0.6667	0.7353
ThinkPRM 7B	0.6849	0.8003	0.7834	0.8149	0.7326	0.8046	0.7619	0.8881	0.7999	0.8682
ThinkPRM 14B	0.8582	0.9142	0.8859	0.9140	0.8150	0.8706	0.9118	0.9274	0.8474	0.9188
Qwen2.5-Math-PRM-7B	0.9679	0.9606	0.9512	0.9521	0.9350	0.9433	0.8614	0.9562	0.9746	0.9512
Qwen2.5-Math-PRM-72B	0.8368	0.9517	0.8896	0.9457	0.9108	0.9502	0.8703	0.9554	0.8961	0.9504
CAID (Ours)	**0.0000**	0.5752	**0.0174**	0.5596	**0.2006**	0.5142	**0.0190**	0.4799	0.1598	0.5155

Table 1: Step Preservation Rate (SPR) comparison by augmentation type. Aug PR: Augmented Step Preservation Rate (lower is better), Gold PR: Gold Step Preservation Rate (higher indicates retention). **Bold** indicates the best performance, and underlined indicates the second-best performance.

- **Standard Benchmarks:** To validate PACE on real-world reasoning, we use **GSM8K** (Cobbe et al., 2021), **StrategyQA** (Geva et al., 2021), and **ARC-Challenge** (Clark et al., 2018). These datasets cover arithmetic, commonsense, and scientific reasoning, respectively, allowing us to test whether CAID generalizes across different domains without task-specific tuning.

Baselines. For the diagnostic comparison on RIV-GSM8K, we evaluate state-of-the-art PRMs including **ReasonEval-34B** (Xia et al., 2025), **ThinkPRM-7B** (Khalifa et al., 2025), and **Qwen-Math-PRM-72B** (Yang et al., 2024). For the compression validation via PACE, we use **Llama-3.1-8B-Instruct** as the backbone model and compare the compressed chains against the standard **Zero-shot CoT** baseline to measure the trade-off between token reduction and accuracy.

Implementation of CAID. We utilize lightweight off-the-shelf models for efficiency: **all-MiniLM-L6-v2** (22M) for semantic encoding and **GPT-2 Small** (124M) for density estimation. To ensure robustness, we use a fixed set of hyperparameters across all datasets without task-specific tuning. Detailed model configurations, threshold values, and sensitivity analysis are provided in Appendix D.

4.2 Results 1: Diagnostic Capability on RIV-GSM8K

We evaluate how well different evaluators handle the inefficiencies injected into RIV-GSM8K. Instead of binary accuracy, we report the **Step Preservation Rate (SPR)**, defined as the ratio of steps retained after evaluation.

- **Augmented PR (Aug PR):** Measures recall on inefficient steps. **Lower is better**, indicating the model successfully removed or flagged the inefficient step.

- **Gold PR:** Measures retention of original human-written steps. **Higher typically indicates safety**, assuming human steps are perfectly efficient. However, as discussed below, we challenge this assumption.

Blind Spot of Validity-Focused Evaluators. As shown in Table 1, existing methods exhibit surprisingly high Aug PR scores. A critical finding is the performance of **ReasonEval**. Despite being a specialized reasoning step evaluator equipped with an explicit *redundancy score*, it fails to effectively penalize redundant steps, retaining approximately 70% of *Simple Duplication*, *Paraphrase*, and *Decompose* types. Similarly, even the 72B-parameter Qwen-Math preserves 83.68% of *Simple Duplications*. This confirms that validity-focused models, regardless of their size or specific scoring sub-metrics, remain essentially blind to inefficiency as long as the statement is factually correct.

Effectiveness and Efficiency of CAID. In contrast, CAID achieves near-perfect detection on redundancy, with an Aug PR of **0.0000** for Duplication and **0.0174** for Paraphrasing. It also effectively identifies complex inefficiencies like Circular Reasoning (0.0190) and Decomposition (0.2006), where baselines struggle significantly. Regarding *Irrelevant* steps, while the large-scale supervised model **ReasonEval-34B** achieves the best performance (0.0331), CAID (0.1598) demonstrates competitive capability, performing comparably to **ReasonEval-7B** (0.1509). Crucially, CAID achieves this with a total of only **146M parameters** (124M GPT-2 + 22M MiniLM), whereas ReasonEval requires 7B to 34B parameters. This demonstrates that CAID delivers robust diagnostic

precision with orders of magnitude greater computational efficiency than large-scale supervised evaluators.

Redefining "Gold": Deletion vs. Compression. A distinct characteristic of CAID is its lower Gold PR (≈0.55) compared to baselines (> 0.90). While this might initially appear as over-penalization, a granular analysis of the action distribution reveals that CAID is not "wrong," but rather stricter regarding information density. Out of 11,899 Gold steps not fully preserved by CAID:

- Only **1.5% (184 steps)** were flagged for removal (PRUNE), primarily due to high redundancy (169 steps) or irrelevance (15 steps).
- The remaining **98.5% (≈11,700 steps)** were flagged for **MERGE**.

As qualitatively analyzed in Appendix E (Table 5), these flagged steps are factually valid but functionally inefficient. For instance, steps that merely restate a calculated value (e.g., "Child = 4" → "Ticket is $4") trigger the **Low Semantic Delta** criteria due to a lack of logical progress. Similarly, steps that verbally describe an operation before executing it (e.g., "Then multiply the number...") are flagged for **Low Information Density**. This empirical evidence suggests that standard datasets contain significant **latent inefficiency**, validating our approach of *compression* over blind retention.

4.3 Results 2: Efficiency via PACE

To empirically validate the diagnostic precision of CAID, we applied PACE as a compression probe to reasoning chains generated by Llama-3.1-8B. Table 2 summarizes the trade-off between token reduction and reasoning accuracy.

Dataset	Method	Performance		Efficiency	
		Acc (%)	Δ	Tokens	Red (%)
GSM8K	Baseline	82.03	-	214.6	-
	PACE	81.12	-0.91	**148.0**	**-31.0%**
StrategyQA	Baseline	70.31	-	327.7	-
	PACE	69.93	-0.37	**154.4**	**-52.9%**
ARC-C	Baseline	83.70	-	277.8	-
	PACE	**84.64**	**+0.94**	**156.7**	**-43.6%**

Table 2: Comparison of accuracy and token usage between baseline CoT and PACE. PACE significantly reduces tokens while maintaining or improving accuracy.

Validating Latent Inefficiency via Compression. PACE successfully reduces token consumption by 31.0% to 52.9% across all datasets with negligible impact on accuracy ($< 1\%$). This empirical finding serves as strong evidence for the prevalence of **latent inefficiency** in standard CoT reasoning. It confirms that a significant portion of generated tokens acts as informational "froth"—valid but functionally unnecessary—and that CAID correctly identifies this redundancy without disrupting the deductive chain.

Enhancing Clarity in Information-Heavy Domains. Interestingly, on ARC-Challenge, PACE actually improves accuracy by +0.94%p while reducing tokens by 43.6%. Unlike arithmetic tasks where steps are strictly sequential, scientific reasoning often suffers from hallucinated tangents or excessive context. By pruning these distractions (via Goal Alignment) and condensing verbose explanations (via Density), PACE effectively increases the **signal-to-noise ratio** of the reasoning context, helping the model focus on critical scientific principles.

4.4 Ablation Study

To investigate the contribution of each information-theoretic component, we conducted a cumulative ablation study on GSM8K. Figure 3 visualizes the performance gain as we integrate each metric.

Necessity of Semantic Delta. As shown in the figure, reliance on *Local Similarity* alone effectively detects lexical repetitions but fails completely on *Decomposition* and *Irrelevance* (near 0% detection). Adding *Semantic Delta* triggers a sharp performance boost, confirming that measuring the "vector shift" or logical velocity is essential for identifying stalled reasoning.

Role of Information Density. *Information Density* proves critical for detecting *Circular Reasoning*. Tautological statements often exhibit low perplexity (high probability), which CAID successfully flags as low-density steps. This confirms our hypothesis that verbosity is not just about raw length, but about information content relative to length (entropy rate).

Impact of Compression Strategy. We further evaluated the effectiveness of different compression actions and safety constraints. Our results show that simply deleting steps with low logical progress causes a sharp accuracy drop (-5.38%), implying that these steps serve as essential "connective tissue." In contrast, PACE's merging strategy, reinforced by safety constraints (Consistency and

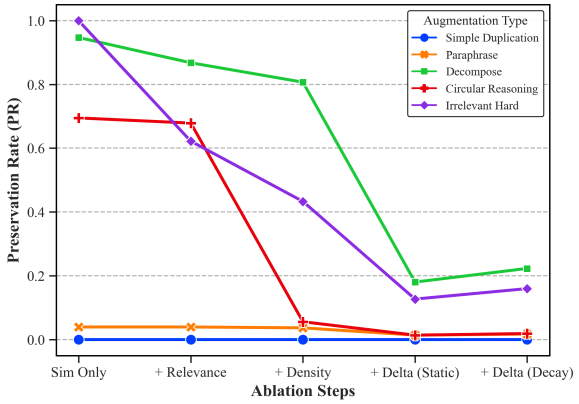

Figure 3: Impact of cumulative CAID components on Preservation Rate (PR). Lower PR indicates better detection. The integration of **Information Density** and **Semantic Delta** proves critical for detecting complex inefficiencies (e.g., *Circular Reasoning*, *Decomposition*) that bypass surface-level similarity.

Saturation), successfully recovers accuracy while maintaining high efficiency. Detailed ablation results and analysis are provided in Appendix F.

5 Discussion

The Phenomenon of Latent Inefficiency. A significant revelation in our experiments is that CAID identifies approximately 50% of human-written (Golden) steps as **compressible** rather than strictly necessary. As detailed in Section 4.2, the vast majority (98.5%) of these flagged steps were categorized for MERGE due to **low information density** or **minimal logical progress**, with only 1.5% identified as redundant or irrelevant enough for removal. This points to a prevalent phenomenon of *Latent Inefficiency* in current CoT datasets: valid reasoning is often diluted by excessive verbosity or fragmented micro-steps. It challenges the prevailing assumption that human-written reasoning is the "gold standard" for efficiency, highlighting the need for metrics like CAID to prioritize **information density** alongside correctness.

Beyond Post-hoc: Implications for Data-Centric AI. While PACE is presented here as a diagnostic compression strategy, its capability to distill informational "froth" offers substantial value in two key data-centric applications:

- **Efficiency-Aware Instruction Tuning:** PACE can serve as a high-quality filtration pipeline for Data-Centric AI. By constructing **efficiency-aware training datasets** from dense reasoning chains, we can potentially fine-tune smaller models (Students) to reason efficiently from scratch. This effectively transfers the information density inherent in CAID to the model weights, eliminating the need for runtime post-processing.

- **RAG Context Compression:** In Retrieval-Augmented Generation (RAG) systems, retrieved reasoning paths often consume significant memory. PACE can pre-compress these chains offline, allowing systems to retrieve and process more relevant context within the same token budget, thereby enhancing multi-turn reasoning efficiency via 'Context Reuse'.

6 Conclusion

In this work, we investigated the critical inefficiency of "over-reasoning" in Large Language Models, diagnosing that current models often generate valid but redundant, verbose, or irrelevant steps that inflate computational costs without contributing to the solution. While existing reasoning step evaluators excel at verifying correctness, we demonstrated their inability to detect these subtle forms of inefficiency due to a lack of information-theoretic criteria.

To address this, we proposed a comprehensive framework comprising diagnosis, measurement, and validation. First, we introduced **RIV-GSM8K**, a diagnostic benchmark that exposed the "efficiency blind spots" of state-of-the-art evaluators. Second, we developed **CAID**, a reference-free metric that quantifies the informational utility of reasoning steps by integrating local novelty, information density, and global goal alignment. Finally, we validated the diagnostic precision of this metric through **PACE**, a post-hoc compression strategy.

Our experiments confirmed that PACE achieves substantial token reductions of 31–53% across arithmetic, commonsense, and scientific domains while maintaining reasoning accuracy. This empirically validates the existence of significant "latent inefficiency" in standard CoT reasoning, proving that a large portion of tokens represents informational "froth" that can be safely compressed without breaking the deductive chain. By shifting the paradigm of reasoning evaluation from "Correctness-only" to **"Correctness-and-Efficiency,"** this work establishes a foundation for building more sustainable reasoning systems and constructing efficiency-aware training datasets.

Limitations

While our framework effectively diagnoses and mitigates reasoning inefficiency, we acknowledge several limitations.

Computational Overhead of Post-hoc Process. PACE operates as a generate-then-refine strategy, meaning it does not reduce the latency of the initial inference pass. Consequently, it is best suited for offline applications—such as compressing retrieved context for RAG or synthesizing efficiency-aware training data—rather than real-time latency reduction.

Dependency on Writer Capability. The `MERGE` action relies on the semantic capability of the underlying LLM to rewrite verbose steps without information loss. While our safety constraints (τ_{merge}, τ_{sat}) mitigate semantic drift, there remains a residual risk that smaller models may over-simplify complex logic during compression.

Domain Generalizability. Our experiments focused on arithmetic (GSM8K), commonsense (StrategyQA), and scientific reasoning (ARC-C). The definition of "efficiency" may differ in creative or open-ended domains (e.g., storytelling) where verbosity serves a stylistic purpose. Future work is required to adapt information-theoretic criteria for such subjective tasks.

Hyperparameter Sensitivity. Although CAID demonstrated robustness across our benchmarks with a fixed set of thresholds, applying the metric to domains with vastly different linguistic densities (e.g., code generation or legal texts) may require task-specific hyperparameter tuning.

Ethics Statement

This work aligns with the goals of **Green AI** by promoting sustainable computation through the reduction of token consumption in LLMs.

Data Usage and Privacy. Our research utilizes publicly available datasets and does not involve private or personally identifiable information. The synthetic perturbations in RIV-GSM8K were generated using standard LLMs (GPT-4o) and do not contain harmful content.

Potential Biases. We acknowledge that automated efficiency metrics could inadvertently penalize specific linguistic styles or dialects that are naturally verbose. Users should exercise caution to ensure that valid dialectal variations are not unfairly classified as redundant "froth."

Use of AI Assistants. We utilized GPT-4o as a component of our data augmentation pipeline to generate synthetic reasoning inefficiencies for the RIV-GSM8K benchmark, as detailed in Section 3.1 and Appendix A. Additionally, AI assistants were used for preliminary code implementation and linguistic polishing of the manuscript. The authors have proofread all AI-generated content and remain responsible for the final output.

References

Lingjiao Chen, Matei Zaharia, and James Zou. 2024. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*. Featured Certification.

Cheng-Han Chiang and Hung-yi Lee. 2024. Over-reasoning and redundant calculation of large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–169, St. Julian's, Malta. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457.

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. *Preprint*, arXiv:2110.14168.

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

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023. LLMLingua: Compressing prompts for accelerated inference of large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *arXiv preprint arXiv:2504.16828*.

Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. 2022. Learned token pruning for transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 784–794, New York, NY, USA. Association for Computing Machinery.

YUCHENG LI, BO DONG, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

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*.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. Evaluating mathematical reasoning beyond accuracy. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26):27723–27730.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

A RIV-GSM8K Construction Details

In this section, we provide the detailed algorithm and statistical breakdown of the RIV-GSM8K dataset construction.

A.1 Construction Algorithm

Algorithm 1 outlines the step-by-step procedure for injecting inefficiencies into the GSM8K dataset.

Algorithm 1 RIV-GSM8K Benchmark Construction

Require: Dataset $\mathcal{D}_{GSM8K} = \{(Q, C)\}$, Generator $\mathcal{G}$
Require: Types $\mathcal{T} = \{\text{Dup, Para, Dec, Circ, Irr}\}$
Ensure: RIV-GSM8K Dataset $\mathcal{D}_{RIV}$
$\mathcal{D}_{RIV} \leftarrow \emptyset$; $\mathcal{I}_{\mathcal{T}} \leftarrow \text{Cycle}(\mathcal{T})$
for all $(Q, C = \{S_1, \ldots, S_N\}) \in \mathcal{D}_{GSM8K}$ **do**
  $T \leftarrow \text{next}(\mathcal{I}_{\mathcal{T}})$
  Pick random index $t \in [1, N]$ {Calc. steps only if T=Dec}
  if $T ==$ Simple Dup **then**
    $S_{new} \leftarrow S_t$
  else
    $S_{new} \leftarrow \mathcal{G}(Q, C, S_t \mid T)$
  end if
  Update: If T=Dec, replace S_t with S_{new}; else insert S_{new}.
  Add (Q, C_{new}, T) to $\mathcal{D}_{RIV}$
end for
return $\mathcal{D}_{RIV}$

A.2 Dataset Statistics

Table 3 summarizes the distribution of the five inefficiency types within the constructed benchmark. We explicitly distinguish between 'Augmented Steps' (the injected noise) and 'Normal Steps' (the original baseline reasoning).

Perturbation Type	Samples	# Aug. Steps	# Norm. Steps
Simple Duplication	1,495	1,495	6,829
Paraphrase	1,495	1,495	6,849
Decompose	1,495	7,834	5,364
Circular Reasoning	1,494	3,955	6,857
Irrelevant	1,494	5,402	6,811
Total	**7,473**	**20,181**	**32,780**

Table 3: Statistics of the RIV-GSM8K Benchmark. We explicitly distinguish between Augmented Steps (injected noise) and Normal Steps (baseline reasoning).

B Prompt Details and Configuration

We utilize `gpt-4o` for all augmentation tasks. To ensure generation diversity and structural consis-

Task 1: Paraphrase

System Prompt:

```
You are an expert paraphrasing assistant. Your task is to rephrase the given text while preserving
its original meaning. Ensure that the paraphrased text is clear, coherent, and maintains the same
intent as the original. Avoid using overly complex language or altering the tone significantly.
Crucially, wrap all calculations in «calculation=result» format.
```

User Input Template:

"Please paraphrase the following text: [INPUT_TEXT]"

Task 2: Decompose

System Prompt:

```
You are an expert at breaking down complex tasks into smaller, manageable, verbose subtasks.
Your goal is to analyze the given task description and decompose it into a list of sequential
steps that can be followed to accomplish the task effectively. Ensure that decomposed steps
maintain same intent as the original. Avoid using overly complex language or altering the tone
significantly. Crucially, wrap all calculations in «calculation=result» format.
```

User Input Template:

"Please decompose the following task into smaller subtasks: [INPUT_TASK]"

Figure 4: Configuration and prompt details for basic augmentation strategies (*Paraphrase* and *Decompose*). The top block shows shared hyperparameters.

tency, we employ a **temperature of 1.0** and a **maximum token limit of 5,000**. Additionally, we enforce a **strict JSON Schema** for all outputs to facilitate robust parsing.

Below, we detail the specific system prompts and input templates used for our pipeline. Figure 4 outlines the basic augmentation types, while Figure 5 details the context-aware strategies.

C Qualitative Examples of Reasoning Augmentations

To better understand the nature of the perturbations introduced by our pipeline, we provide concrete examples of generated reasoning steps in Table 4. These examples are derived from a single original step in the GSM8K dataset: *"Adults = 10 * 8 = 80"*.

As shown in Table 4, our augmentation strategies cover a spectrum of "over-reasoning" behaviors observed in Large Language Models:

- **Surface-level Redundancy:** *Simple Duplication* and *Paraphrase* retain the exact logic of the original step but introduce lexical variations or repetitions, testing the model's robustness to verbose phrasing.
- **Granularity Expansion:** *Decompose* breaks down a single atomic operation into a verbose chain of micro-steps (e.g., identifying variables, stating the operation, calculating, and restating the result), significantly inflating the token count without adding deductive value.
- **Logical Loops:** *Circular Reasoning* mimics a model's tendency to "double-check" itself unnecessarily. It uses inverse operations (division/multiplication) to verify an already established fact, creating a closed logical loop that adds computational cost.
- **Contextual Noise:** *Irrelevant* introduces distractors that differ from hallucinations; they are mathematically true and contextually plausible (e.g., discussing ticket prices or family groups) but contribute nothing to the solution path.

These qualitative samples illustrate the diverse challenges our dataset poses to reasoning evaluators.

D Implementation Details

D.1 Model Configuration

CAID is designed to be computationally efficient and widely applicable. We employ the following off-the-shelf models:

- **Semantic Encoder ($\mathcal{E}$):** We use `all-MiniLM-L6-v2` (22M parameters)

Task 3: Circular Reasoning

System Prompt:

```
You are an expert at inserting circular reasoning into mathematical solutions. Your task is to generate a sequence of steps that redundantly verifies a previously established fact or calculated number using inverse operations or self-referential logic.
You will be provided with:

1. The Question
2. Previous Reasoning Steps
3. The Current Target Step

Generate a reasoning section that:

• Takes a number or fact already established in the 'Previous Reasoning Steps'.
• Performs a set of operations that eventually lead back to the original number (e.g., "Since X is 5, multiplying by 2 gives 10, and dividing by 2 returns 5, confirming X is indeed 5.").
• Is mathematically true but strictly unnecessary for solving the problem.
• Does not alter the final answer or the logical path required for the solution.

Crucially, wrap all calculations in «calculation=result» format.
```

User Input Template:

"Based on the context below, generate circular reasoning sentences that could be inserted after the Current Step:"

Task 4: Irrelevant (Hard)

System Prompt:

```
You are an expert at generating context-aware distractions. Your task is to generate mathematically correct but irrelevant sentences that sound like they belong to the solution flow but do not advance the solution logic or provide any new information needed for the answer.
You will be provided with:

1. The Question
2. Previous Reasoning Steps
3. The Current Target Step
4. Next Reasoning Steps

Generate a reasoning section that:

• naturally fits between the previous reasoning, the current target step, and the next reasoning steps,
• maintains the same tone, context, and mathematical domain,
• uses a smooth transitional phrase to connect the surrounding steps,
• is mathematically true but does not contribute to solving the problem,
• does not alter any variables, numbers, or assumptions in the reasoning,
• and does not suggest new solution paths or constraints.

Crucially, wrap all calculations in «calculation=result» format if any numbers appear.
```

User Input Template:

"Based on the context below, generate irrelevant sentences that could be inserted after the Current Step:"

Figure 5: Prompt configurations for context-aware augmentation types (*Circular Reasoning* and *Irrelevant*). Note that these tasks require full context inputs (question, previous/next reasoning steps).

to compute cosine similarity for Local Similarity ($\mathcal{M}_{sim}$) and Global Goal Alignment ($\mathcal{M}_{rel}$). This model was selected for its high speed and strong performance on semantic textual similarity tasks.

- **Density Estimator ($\mathcal{M}$):** We use `GPT-2 Small` (124M parameters) to calculate the perplexity for Information Density ($\mathcal{M}_{density}$).

The total parameter count for CAID is approximately 146M, which is significantly smaller than the baseline PRMs (e.g., ReasonEval-34B).

Augmentation Type	Generated Reasoning Step
Original Input	**Adults = 10 * 8 = $«10*8=80»80**
1. Simple Duplication	Adults = 10 * 8 = $«10*8=80»80
2. Paraphrase	Adults: 10 multiplied by 8 equals $«10*8=80»80
3. Decompose	Identify the number of adults involved, which is given as 10. → Identify the multiplication factor for the adults, which is 8 in this case. → Multiply the number of adults (10) by the factor (8): «calculation=80». → State the result of the multiplication, which is $80.
4. Circular Reasoning	Having calculated $80 for the adults, we can verify this by noting that $80 divided by the 10 adults gives us $«80/10=8»8 per adult. → Re-multiplying this $8 per adult by 10 adults returns $«8*10=80»80, ensuring our number is correct. → Alternatively, if we consider twice the accounted $80, we get $«80*2=160»160, and half of this sum is $«160/2=80»80, which confirms our original total remains consistent at $80.
5. Irrelevant	Since the adults paid a total of $80, this also means they chose to spend their money on tickets rather than concessions at the game, an interesting financial decision given the usual pricing. → It's notable that the sum of $124 offers no remainder when considering integer ticket prices, suggesting an elegance in the problem structure. → The relationship between the number of adults and children indicates a typical family group could have attended this game together.

Table 4: Examples of augmented reasoning steps generated from a single original step. The generated text aims to mimic specific reasoning flaws or stylistic variations.

D.2 Hyperparameters

We utilize a fixed set of thresholds across all experiments (GSM8K, StrategyQA, ARC-Challenge) to demonstrate the generalizability of our metric. The specific values are:

- **Removal Thresholds (PRUNE):**
  - High Redundancy: $\tau_{sim} = 0.85$
  - Low Relevance: $\tau_{rel} = 0.25$
- **Compression Candidates (MERGE):**
  - Low Information Density: $\tau_{density} = 0.1$
  - Low Semantic Delta (Base): $\tau_{delta} = 0.03$
- **Adaptive Decay:**
  - Decay Factor: $\lambda = 0.95$ (Applied as $\tau_\delta(t) = \tau_{delta} \cdot \lambda^t$)

Sensitivity Analysis. We observed that the performance of CAID is relatively stable around these threshold values. For instance, varying τ_{sim} between 0.80 and 0.90 or τ_{rel} between 0.20 and 0.30 resulted in minimal fluctuations in the Step Preservation Rate (SPR) on the RIV-GSM8K validation set. This suggests that the chosen hyperparameters are robust and not overfitted to a specific dataset distribution.

E Qualitative Analysis of Latent Inefficiency

To better understand the nature of "Latent Inefficiency" in human-written Gold data, we provide a detailed qualitative analysis of steps flagged for MERGE by CAID. Table 5 presents concrete examples from the GSM8K dataset.

Previous Step (S_{t-1})	**Target Step** (S_t) **[Gold]**	**Reason**
Child = 44/11 = $«44/11=4»4	*Each child's ticket is $«4=4»4.*	**Low Delta**
→ ***Diagnosis:*** *The target step merely repeats the value '4' established in the previous step, contributing no new deductive information (No Progress).*		
*...when self-checkout is broken... 160 * 1.2 = 192 complaints/day*	*Then multiply the number of complaints per day by the number of days...: 192 * 3 = 576...*	**Low Density**
→ ***Diagnosis:*** *The step explicitly describes the operation before performing it, inflating token usage (Verbose).*		

Table 5: Qualitative examples of Gold steps flagged for MERGE. By utilizing the full width for diagnosis, we clarify why valid steps are identified as inefficient (e.g., lack of progress or verbosity).

F Detailed Ablation on Compression Strategy

In this section, we provide the extended ablation study on the compression strategies employed in

ID	Method Description	Performance		Efficiency		Ratio
		Acc (%)	Δ	Tok Red (%)	Step Red (%)	(Tok)
0	Baseline (Original CoT)	82.03	0.00	0.00	0.00	1.00
1	+ Similarity (Remove)	84.46	+2.43	7.36	15.47	1.08
2	+ Relevance (Remove)	84.38	+2.35	8.91	17.06	1.10
3	+ Density (Remove)	**84.53**	**+2.50**	20.12	21.69	1.25
4	+ Delta (Remove)	76.65	-5.38	11.18	**72.52**	1.13
5	+ Merge (No Safety)	76.95	-5.08	**45.09**	70.41	**1.82**
6	**PACE (Full Method)**	81.12	-0.91	31.05	53.53	1.45

Table 6: Ablation study of PACE components. We analyze the impact of each module on accuracy and compression efficiency. **Modes 1–4** use removal-only logic, while **Modes 5–6** introduce the merging mechanism. **PACE (Mode 6)** achieves the best balance between accuracy recovery and token reduction.

PACE, validating the necessity of the MERGE action and safety constraints.

Deletion vs. Compression. As shown in Table 6, while removing redundant steps (Modes 1–3) yields slight accuracy gains, simply deleting steps with low logical progress (Mode 4, Delta) causes a sharp accuracy drop (-5.38%). This implies that even repetitive or slow-progressing steps serve as essential "connective tissue" in the reasoning chain, carrying implicit dependencies. They cannot be blindly removed (Prune) but must be **merged** to preserve logical continuity while reducing verbosity.

Necessity of Safety Constraints. Mode 5 (Merge without constraints) achieves high token reduction (45%) but suffers significant accuracy degradation (-5.08%) due to semantic drift and information overload. By enforcing our safety constraints (Consistency and Saturation), **PACE (Mode 6)** successfully recovers the accuracy (Acc 81.12%) while still delivering substantial efficiency (31% token reduction), demonstrating that our density-aware merging strategy achieves the optimal trade-off between compression and validity.