Valid ≠ Necessary: Diagnosing Latent Inefficiency in Chain-of-Thought

Daeyeop Lee[1,2] **and Hwanjo Yu**[2*]
[1]KT Corporation
[2]Pohang University of Science and Technology
daeyeop.lee@kt.com
hwanjoyu@postech.ac.kr

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 gap, we propose **CAID** (Context-Aware Information Density), a training-free metric grounded in information theory that identifies low-utility steps. To validate the metric's practical utility, we apply it within **PACE**, a post-hoc compression strategy. Additional control experiments show that the gains of PACE are not explained by trivial pruning: compared with random step removal and PRM-based compression baselines, it preserves accuracy at substantially higher compression rates. 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, CoT helps bridge the gap between question and answer. However, these gains often come at the cost of inference efficiency. Recent studies show that LLMs tend to **"over-reason"** by generating verbose explanations, repetitive statements, or contextually irrelevant details that increase computational cost without adding deductive value (Turpin et al., 2023; Wang et al., 2023; Chiang and Lee, 2024). More recent work has therefore begun to treat *reasoning efficiency* itself as an important objective, for example through rationale reduction and concise intermediate reasoning formats (Jang et al., 2025; Xu et al., 2025).

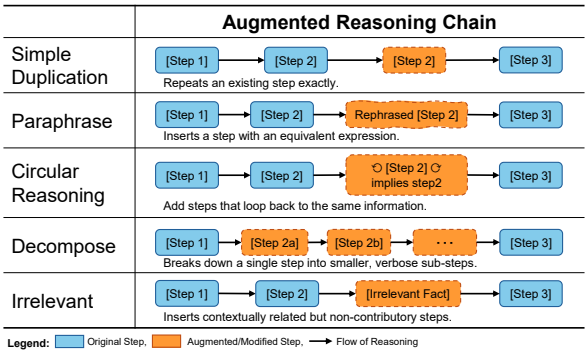

Figure 1: Taxonomy of reasoning inefficiencies in RIV-GSM8K. The diagram illustrates how five distinct types of redundant steps are synthetically injected into the reasoning chain to simulate valid but dispensable "froth."

To assess and improve reasoning quality, various Process Reward Models (PRMs) and reasoning-step evaluators have been proposed, including ReasonEval and Math-Shepherd (Xia et al., 2025; Wang et al., 2024). More broadly, recent benchmarks and surveys suggest that reasoning quality extends beyond step correctness to dimensions such as coherence, utility, and simplicity (Song et al., 2025; Lee and Hockenmaier, 2025). In parallel, faithfulness studies have shown that final-answer accuracy can sometimes remain high even when parts of a reasoning trace are truncated or perturbed,

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 38998–39014

highlighting the need for careful controls when evaluating compressed CoT traces (Lanham et al., 2023). Despite these developments, existing evaluators remain largely optimized for **correctness** and **logical validity**. Our analysis reveals a critical blind spot: they often fail to distinguish *inefficiency* from *reasoning*. In particular, they may assign high scores to "valid but redundant" steps—such as excessive decomposition or circular logic—simply because those steps remain factually true and linguistically coherent. As a result, current reasoning evaluation still focuses primarily on "Is this step true?", while largely overlooking the equally important question: **"Is this step truly necessary?"**

In this paper, we shift the focus of reasoning evaluation from correctness alone toward **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 injects five distinct types of inefficiency, ranging from simple duplication to subtle circular reasoning. Using this benchmark, we show 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 that evaluates reasoning steps using information-utility signals such as local novelty, global goal alignment, and information density. Unlike prior metrics, CAID is designed to identify informational "froth" within reasoning chains without relying on a reference trace. To **empirically validate the diagnostic precision** of this metric, we introduce **PACE** (Pruning And Compression for Efficiency), a post-hoc compression strategy. Rather than simply deleting steps, PACE identifies *latent inefficiency* in reasoning chains and predominantly **compresses** verbose steps (Merge) while pruning irrelevant ones, making the chain substantially more compact while preserving logical coherence. To strengthen this validation, we further compare PACE against random step-removal controls and PRM-based compression baselines, showing that its gains are not explained by trivial pruning and that validity-oriented evaluators yield only limited compression in the same setting.

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 inefficiency.
- We propose **CAID**, an interpretable, training-free metric that estimates 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 with minimal accuracy loss. Additional control experiments with random pruning and PRM-based compression baselines show that these gains arise from **selective compression**, not trivial deletion. Together, these results provide empirical evidence for substantial *latent inefficiency* in standard CoT reasoning, suggesting that high-quality reasoning data can be considerably more compact than commonly assumed.

2 Related Work

2.1 Over-reasoning and Inference Efficiency

Chain-of-Thought (CoT) prompting (Wei et al., 2022) has substantially improved multi-step reasoning in LLMs, but often at the cost of inference efficiency. Recent studies report that LLMs frequently produce overly long rationales, repetitive verification loops, or contextually unhelpful details—a phenomenon often described as **over-reasoning** (Turpin et al., 2023; Chen et al., 2024; Chiang and Lee, 2024). Such redundancy can increase computational cost and, in some cases, even harm performance by distracting the model from task-relevant information (Jiang et al., 2023). More recent work has therefore started to treat *reasoning efficiency* itself as an optimization target, for example by shortening verbose rationales or encouraging concise intermediate reasoning formats (Jang et al., 2025; Xu et al., 2025).

A related line of work aims to reduce inference cost through token pruning, such as H2O (Zhang et al., 2023) and Learned Token Pruning (Kim et al., 2022). These methods operate at the **token level**, primarily targeting runtime efficiency (e.g., attention or KV-cache reduction). By contrast, our focus is on **semantic inefficiency within reasoning steps**. Rather than pruning tokens during generation, PACE identifies and compresses low-utility reasoning content after generation. This makes our

approach complementary to runtime-oriented token pruning methods.

2.2 Reasoning Step Evaluation Methods

Beyond outcome-based evaluation, step-wise evaluation methods have been proposed to provide finer-grained supervision for reasoning. Process Reward Models (PRMs), such as PRM800K (Lightman et al., 2024) and Math-Shepherd (Wang et al., 2024), are primarily designed to distinguish correct from incorrect intermediate reasoning. More recently, **ReasonEval** (Xia et al., 2025) extended this direction by considering additional dimensions such as redundancy and clarity. In parallel, broader benchmarks and surveys have argued that reasoning quality should be assessed along multiple axes—including correctness, coherence, utility, and simplicity—rather than correctness alone (Song et al., 2025; Lee and Hockenmaier, 2025).

Despite this progress, existing evaluators remain limited for diagnosing inefficiency. Standard PRMs are optimized mainly for **correctness verification**: they penalize factual or logical errors, but often assign favorable scores to steps that are valid yet unnecessary. ReasonEval moves closer to our goal, but its judgments are learned from supervised annotations, where *necessity* and *conciseness* can be harder to define consistently than correctness. In contrast, our approach uses **RIV-GSM8K** as a controlled diagnostic benchmark, where inefficiencies are synthetically injected under explicit constraints. This setup enables a more direct test of whether an evaluator can distinguish *necessary* reasoning from *valid but inefficient* reasoning.

2.3 Redundancy Detection Metrics

Several prior works have addressed redundancy or information compression in text. ROSCOE (Golovneva et al., 2023) proposes a suite of metrics for evaluating reasoning quality, including semantic-similarity-based measures for detecting repetition. Similarly, Li et al. (2023) introduced Selective Context, which uses self-information (perplexity) to compress prompts by removing less informative content. Related faithfulness studies further showed that final-answer accuracy can sometimes remain high even when portions of a reasoning trace are truncated, highlighting an important caveat for compression-based evaluation of CoT (Lanham et al., 2023).

However, these approaches do not fully address inefficiency in multi-step reasoning. Similarity-based metrics such as ROSCOE are most effective for detecting overt repetition, but are less suited to cases where a step is locally fluent yet contributes little to global progress. Reference-based evaluation is also restrictive in our setting, because multiple reasoning chains may be valid and a reference trace is not necessarily efficiency-optimal. Metrics such as Selective Context capture static information content, but do not explicitly model step-wise relevance or logical progress. CAID addresses these gaps with a **reference-free** formulation that combines local redundancy, global goal alignment, information density, and **Semantic Delta** to estimate the utility of each reasoning step in context.

3 Methodology

We present a framework for analyzing and reducing reasoning inefficiency, shifting evaluation from a purely validity-centric perspective toward an **efficiency-aware** one. Our approach consists of three stages: (1) **Diagnosis:** we introduce **RIV-GSM8K** to test whether existing evaluators can detect inefficient but valid reasoning steps; (2) **Measurement:** we propose **CAID**, a reference-free metric for estimating the utility of individual reasoning steps; and (3) **Validation:** we apply **CAID** within **PACE** to examine whether the identified steps can be compressed without substantially degrading downstream performance.

3.1 RIV-GSM8K: Diagnosing Inefficiency

Evaluating reasoning efficiency is challenging because there is rarely a single ground-truth decomposition of which intermediate steps are *necessary*. Human judgments on necessity are often subjective, especially when multiple reasoning traces are valid. To obtain a more controlled testbed, we construct **RIV-GSM8K**, a diagnostic benchmark derived from GSM8K (Cobbe et al., 2021) under a **relative inefficiency** setting. Starting from baseline CoT traces, we inject perturbations that are *less efficient* than the original steps while remaining factually and logically valid. This yields a controlled setting for testing whether an evaluator can distinguish necessary reasoning from *valid but inefficient* reasoning.

Construction Process. The full construction procedure is described in Appendix A. We use a hybrid generation strategy to balance control and diversity. **Simple Duplication** is created by rule-based repeti-

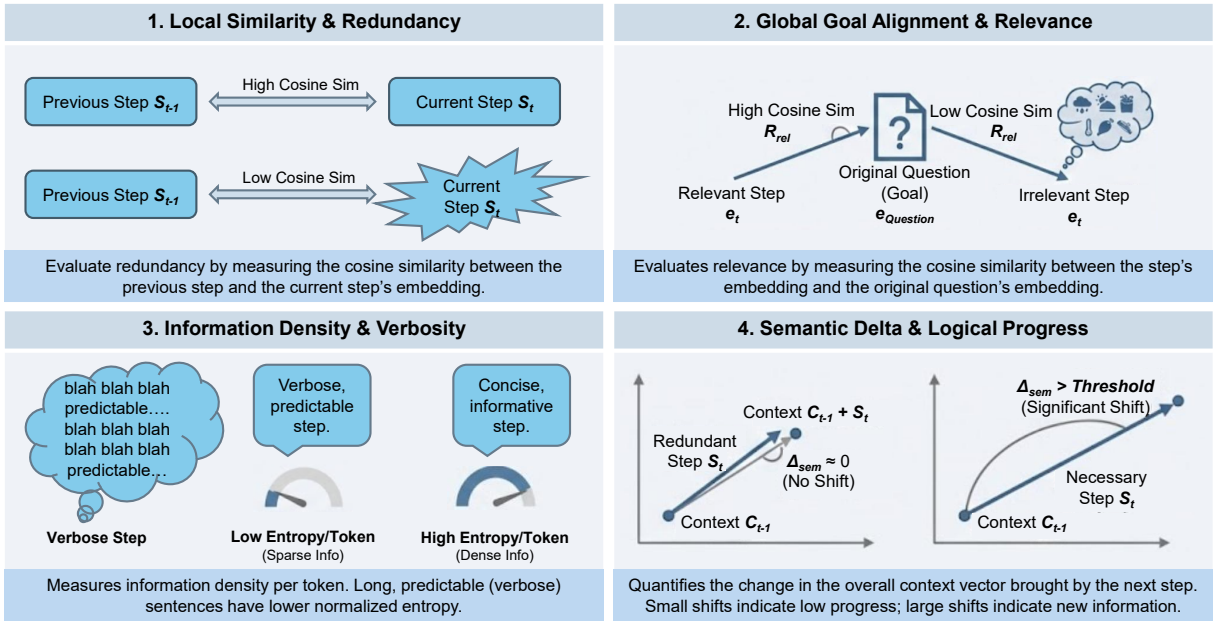

Figure 2: Conceptual overview of the four signals used by CAID. (1) Local Similarity detects surface-level redundancy via adjacent step comparison. (2) Global Goal Alignment filters irrelevant steps by measuring drift from the original question. (3) Information Density identifies verbose, low-entropy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

tion, while the more complex types—**Paraphrase**, **Decompose**, **Circular**, and **Irrelevant**—are generated with **GPT-4o** as the generator $\mathcal{G}$.

To ensure that injected steps remain *valid but inefficient*, we impose two constraints during generation:

- **No New Progress:** the generated step must not advance the reasoning state beyond the target step S_t.
- **Contextual Coherence:** especially for the *Irrelevant* type, the generated content must remain mathematically true and locally coherent with the surrounding context, while contributing nothing necessary to the solution.

The prompts for each perturbation type are provided in Appendix B. In total, RIV-GSM8K contains 7,473 samples and over 20,000 injected steps (Appendix A.2).

As illustrated in Figure 1, we define five types of reasoning inefficiency designed to approximate common over-reasoning patterns in LLMs (see Appendix C for examples). **Simple Duplication** and **Paraphrase** capture lexical and semantic redundancy. **Decompose** models excessive fragmentation of a single reasoning step into multiple low-utility micro-steps. **Circular Reasoning** and **Irrelevant** target stalled logical progress and deviation from the problem objective, respectively.

Human Verification. To verify the quality of the synthetic perturbations, we manually evaluated the four GPT-4o-generated types (*Paraphrase*, *Decompose*, *Circular*, and *Irrelevant*), excluding the rule-based *Simple Duplication*. We randomly sampled 30 instances per type (120 total) and checked whether each example satisfied *Logical Equivalence* and the *No New Progress* constraint.

Overall, the perturbations were highly reliable. *Irrelevant* and *Circular Reasoning* achieved near-perfect validity (29/30 and 30/30, respectively). *Decompose* showed a slightly higher failure rate (4/30), mainly for atomic steps (e.g., simple equations such as $3 \times x = 30$) that are difficult to decompose further without introducing hallucinated details or future-step leakage. Despite these edge cases, most generated steps matched the intended inefficiency type without altering the underlying solution logic.

3.2 CAID: Context-Aware Information Density

Existing evaluators are largely optimized for factual correctness, and therefore often miss reasoning steps that are valid but inefficient. To address this gap, we propose **CAID**, a **reference-free** and unsupervised metric for estimating the *utility* of a reasoning step in context. Rather than targeting a single error type, CAID combines four comple-

mentary signals that capture redundancy, relevance, information density, and contextual progress. Figure 2 summarizes these components.

1. Local Similarity (Redundancy). To detect local repetition, we measure the cosine similarity between the current step S_t and its immediate predecessor S_{t-1} using a lightweight encoder $\mathcal{E}$:

$$\mathcal{M}_{sim}(S_t) = \text{CosSim}(\mathcal{E}(S_t), \mathcal{E}(S_{t-1})).$$

A high similarity score indicates that S_t contributes little beyond the immediately preceding step, which is characteristic of duplication or inefficient paraphrasing.

2. Global Goal Alignment (Relevance). Useful reasoning steps should remain aligned with the problem objective. We therefore measure the similarity between the step and the original question Q:

$$\mathcal{M}_{rel}(S_t) = \text{CosSim}(\mathcal{E}(S_t), \mathcal{E}(Q)).$$

A low alignment score suggests that the step is locally fluent but weakly connected to the goal of the reasoning process.

3. Information Density (Verbosity). We measure how much information a step conveys relative to its length using length-normalized perplexity under a causal language model $\mathcal{M}$:

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

Low-density steps are typically verbose and predictable relative to their token count. This signal is particularly useful for identifying overly decomposed micro-steps that preserve validity while contributing little new content.

4. Semantic Delta (Logical Progress). A reasoning step can be valid yet still fail to move the reasoning state forward. To capture this, we define *Semantic Delta* as the change in the contextual representation after adding S_t:

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

where C_{t-1} denotes the accumulated context up to step $t-1$. A near-zero delta indicates that the new step makes little contextual progress, as in tautological restatements or circular verification.

Because later reasoning steps often yield smaller marginal gains as the chain approaches its conclusion, we use a decaying threshold for $\mathcal{M}_{delta}$:

$$\tau_\delta(t) = \tau_\delta^{base} \lambda^t,$$

which reduces sensitivity to small contextual updates in later positions.

Decision Logic. CAID maps these signals to an action set $\mathcal{A} = \{\text{PRUNE}, \text{MERGE}, \text{KEEP}\}$. Steps with high local redundancy or low goal alignment are treated as unnecessary and assigned PRUNE. Steps that remain relevant but have low information density or low semantic delta are assigned MERGE, indicating that their content may be useful but should be expressed more compactly. All remaining steps are assigned KEEP. This design allows CAID to distinguish *removable* steps from *compressible* ones, rather than treating all low-utility content as equally disposable.

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 use compression as a diagnostic probe: if the steps flagged by CAID can be removed or rewritten into denser forms without harming downstream reasoning, this provides evidence that they were inefficient in their original form.

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

- **PRUNE:** Removes steps flagged as highly redundant ($\mathcal{M}_{sim}$) or weakly relevant to the problem goal ($\mathcal{M}_{rel}$).
- **MERGE:** Compresses steps exhibiting *latent inefficiency* (i.e., valid but overly verbose or 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:** $\text{CosSim}(\mathcal{E}(S'_{last}), \mathcal{E}(S_t)) \geq \tau_{merge}$. This ensures that the new content remains logically compatible with the current context.

2. **Information Saturation:** $\mathcal{I}(S'_{last}) \leq \tau_{max}$. This prevents merging when the current accumulated step is already sufficiently information-dense, thereby avoiding readability loss.

If either constraint is violated, the merge is halted and a new step is initiated.

- **KEEP:** Retains steps that appear necessary for logical progress.

Addressing the "Trivial Accuracy" Concern. One might assume that maintaining accuracy is trivial if the final conclusion step is preserved. To reduce this possibility, we construct the evaluation prompt using the compressed chain C' while **excluding the final answer** (i.e., the "### Result" token). The model must therefore *regenerate* the final answer solely from the remaining reasoning trace. Since reasoning chains are causal, removing or altering a genuinely necessary intermediate step should break the dependencies required to derive the correct solution. In this sense, answer preservation after compression serves as evidence that the removed steps were not essential in their original form. In the experiments, we further strengthen this evaluation by comparing PACE against non-selective pruning controls, allowing us to distinguish selective compression from trivial answer recoverability.

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).

- **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 reasoning domains without task-specific tuning.

Baselines. For the diagnostic comparison on RIV-GSM8K, we evaluate **multiple scales** of representative PRM-based evaluators, including **ReasonEval**, **ThinkPRM**, and **Qwen2.5-Math-PRM**, alongside CAID. For compression validation via PACE, we use **Llama-3.1-8B-Instruct** as the backbone model and compare compressed chains against the standard **Zero-shot CoT** baseline to measure the trade-off between token reduction and answer accuracy on GSM8K, StrategyQA, and ARC-Challenge. To test whether the gains of PACE arise from *selective compression* rather than trivial answer recoverability, we additionally include **Remove-Last** and **Random Pruning** controls on **GSM8K**, following prior faithfulness concerns (Lanham et al., 2023). We further compare against **PRM-based compression baselines** on **GSM8K** by replacing CAID with **ThinkPRM-14B** and **Qwen2.5-Math-PRM-7B** within the same PACE pipeline, thereby isolating the effect of the evaluator while keeping the rewriting and answer-regeneration procedure fixed.

Implementation of CAID. We implement CAID using lightweight off-the-shelf models: **all-MiniLM-L6-v2** (22M) for semantic encoding and **GPT-2 Small** (124M) for density estimation. We use a fixed set of hyperparameters across all datasets without task-specific tuning to test robustness and cross-domain transferability. Detailed model configurations, threshold values, and sensitivity analyses 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 surpris-

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**	<u>0.1509</u>	**0.9571**
ReasonEval 34B	0.7779	0.9558	0.7251	<u>0.9533</u>	0.7604	<u>0.9571</u>	<u>0.3762</u>	0.9345	**0.0331**	0.9118
ThinkPRM 1.5B	<u>0.6264</u>	0.7342	<u>0.6821</u>	0.7486	<u>0.7187</u>	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	<u>0.9606</u>	0.9512	0.9521	0.9350	0.9433	0.8614	<u>0.9562</u>	0.9746	<u>0.9512</u>
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 <u>underlined</u> indicates the second-best performance.

ingly 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 Qwen2.5-Math-PRM 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 ($\approx$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% ($\approx$11,700 steps)** were flagged for **MERGE**.

As qualitatively analyzed in Appendix E (Table 8), these flagged steps are factually valid but functionally inefficient. For instance, steps that merely restate a calculated value (e.g., "Child = 4" $\rightarrow$ "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

We next evaluate whether the steps identified by CAID can be compressed while preserving downstream performance. To this end, we apply PACE to reasoning chains generated by Llama-3.1-8B. Table 2 summarizes the trade-off between answer accuracy and token reduction on standard reasoning benchmarks. To further test whether these gains reflect *selective compression* rather than trivial answer recoverability, we additionally compare PACE against non-selective pruning controls and PRM-based compression baselines on GSM8K.

Compression with Limited Accuracy Loss. Across GSM8K, StrategyQA, and ARC-Challenge, PACE reduces token usage by 31.0%–52.9% while maintaining similar answer accuracy. These results suggest that standard CoT traces contain a substantial amount of **compressible** content: many

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.

Method	Step Red. (%)	Tok Red. (%)	Acc (%)	Δ Acc
Remove Last Only	–	0.26	80.06	-1.97
Random Pruning (30%)	30.00	21.40	76.88	-5.16
Random Pruning (50%)	50.00	32.86	66.72	-15.31
Remove Last + Random (50%)	50.00	29.42	62.55	-19.48
PACE (Ours)	53.53	31.05	81.12	-0.91

Table 3: Comparison between PACE and non-selective pruning controls on GSM8K. At comparable compression rates, random pruning causes substantial accuracy degradation.

generated steps appear to be useful enough to preserve in condensed form, but not necessary in their original verbose form.

Selective Compression, Not Trivial Pruning. A natural concern is that final-answer accuracy may remain high after pruning simply because the model can reconstruct the answer from a partially preserved trace. To test this, we compare PACE against *Remove-Last* and *Random Pruning* controls. As shown in Table 3, selective compression is substantially more robust than non-selective deletion: PACE retains 81.12% accuracy with 31.05% token reduction, whereas random pruning at a comparable compression level leads to much larger degradation (e.g., 66.72% accuracy under 50% random pruning). Even removing only the final step lowers accuracy to 80.06%, suggesting that performance is not explained by simple answer copying from the last reasoning step alone. Overall, these results indicate that PACE preserves key dependencies in the reasoning chain while compressing low-utility content.

Comparison with PRM-based Compression Baselines. We further replace CAID with strong PRM evaluators within the same PACE pipeline, using ThinkPRM-14B and Qwen2.5-Math-PRM-7B as scoring modules. This comparison isolates the role of the evaluator while keeping the rewriting and answer-regeneration procedure fixed. As shown in Table 4, PRM-based variants can maintain competitive answer accuracy, but achieve only limited token reduction ($< 8\%$), whereas CAID yields 31.05% compression on GSM8K. This result is consistent with our main claim: evaluators optimized primarily for validity are less effective at identifying the redundancy/utility dimension targeted by CAID.

Evaluator in PACE	Acc (%)	Δ Acc	Tok Red. (%)
Baseline (Zero-shot)	82.03	–	–
ThinkPRM-14B	79.83	-2.20	7.20
Qwen2.5-Math-PRM-7B	83.02	+0.99	5.87
CAID (Ours)	81.12	-0.91	**31.05**

Table 4: PRM-based compression baselines on GSM8K within the same PACE pipeline. Strong PRM evaluators can maintain reasonably strong answer accuracy, but yield only marginal compression.

A Case of Accuracy Improvement. On ARC-Challenge, PACE improves accuracy by +0.94 points while reducing tokens by 43.6%. One plausible explanation is that, in information-heavy scientific reasoning, removing tangential or overly verbose steps can make the remaining context easier to use. We view this as suggestive evidence that compression can sometimes improve clarity, rather than merely reducing cost.

4.4 Ablation Study

We next analyze the contribution of each component of CAID on GSM8K. Figure 3 shows the cumulative change in preservation rate as each signal is added.

Importance of Semantic Delta. As shown in Figure 3, *Local Similarity* alone captures overt repetition but is insufficient for more subtle inefficiencies such as *Decomposition* and *Irrelevance*. Adding *Semantic Delta* substantially improves detection, indicating that contextual progress is important for distinguishing genuinely useful steps from those that merely restate or locally elaborate prior content.

Role of Information Density. *Information Density* is particularly helpful for *Circular Reasoning*. Tautological or overly predictable steps tend to have low normalized surprisal relative to their length, allowing CAID to identify verbose but low-utility reasoning that is not captured by similarity alone.

Why Compression Requires More than Deletion. We also evaluate different compression actions and safety constraints. Simply removing steps with low logical progress causes a substantial accuracy drop (-5.38%), suggesting that some low-progress steps still serve as connective structure in the chain.

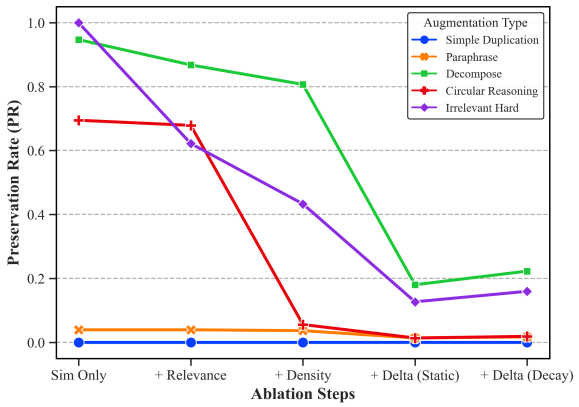

Figure 3: Cumulative effect of CAID components on preservation rate. Lower PR indicates better detection.

In contrast, PACE recovers much of this performance by merging such steps under safety constraints (Consistency and Saturation), supporting the distinction between PRUNE and MERGE. Detailed results are provided in Appendix F.

5 Discussion

Compressible Structure in CoT Traces. One notable result is that CAID marks roughly half of the human-written reference steps as **compressible** rather than strictly removable. As discussed in Section 4.2, most of these cases are assigned to MERGE rather than PRUNE, indicating that the issue is usually not factual incorrectness or irrelevance, but **how the reasoning is expressed**. In many cases, valid steps appear verbose, weakly progressive, or fragmented into micro-steps that could be stated more compactly. This suggests that human-written CoT traces are not always efficiency-optimal, even when they are logically sound.

Compression Requires a Different Signal from Validity. Our control experiments further clarify this distinction. Random pruning substantially degrades accuracy at similar compression levels, while replacing CAID with strong PRM evaluators inside the same PACE pipeline yields only limited compression. Taken together, these results suggest that **correctness-oriented evaluation** and **efficiency-oriented compression** are related but distinct objectives. Existing PRMs are useful for filtering incorrect reasoning, but appear less sensitive to verbose yet valid steps that can be compressed without materially harming downstream performance.

Implications and Limitations. Although PACE is introduced here as a post-hoc compression method, the results suggest two broader directions. First, compressed traces may be useful for constructing denser reasoning datasets for training or distillation. Second, offline compression could help reduce reasoning-context cost in retrieval-based systems. At the same time, our evaluation remains answer-level: it does not test whether compressed traces preserve the model's full continuation trajectory. A fuller account of redundancy in CoT should therefore examine not only final-answer robustness, but also how intermediate continuations change after compression.

6 Conclusion

We studied reasoning inefficiency in Chain-of-Thought (CoT) generation, focusing on cases where intermediate steps are valid yet unnecessarily verbose, repetitive, or weakly connected to the problem objective. Our results suggest that existing reasoning-step evaluators, while effective at verifying correctness, are less effective at identifying this type of low-utility reasoning content.

To address this gap, we introduced three components: **RIV-GSM8K**, a controlled benchmark for diagnosing inefficient but valid reasoning steps; **CAID**, a reference-free metric for estimating the utility of reasoning steps in context; and **PACE**, a post-hoc compression procedure for testing whether the identified steps can be removed or compressed without substantially harming downstream performance.

Across arithmetic, commonsense, and scientific reasoning benchmarks, PACE reduces token usage by 31–53% while maintaining similar answer accuracy. Additional control experiments show that these gains are not explained by trivial pruning, and that strong PRM-based evaluators achieve only limited compression within the same pipeline. Taken together, these findings suggest that standard CoT traces often contain a meaningful amount of compressible content. More broadly, our results motivate reasoning evaluation that considers not only correctness, but also efficiency and utility.

Limitations

Our study has several limitations.

Post-hoc Overhead. PACE follows a generate-then-refine procedure and therefore does not reduce the latency of the initial inference pass. As a result,

it is better suited to offline settings, such as dataset construction or context compression, than to real-time acceleration.

Dependence on Rewriter Quality. The MERGE action depends on the ability of the underlying LLM to rewrite verbose steps without losing important information. Although our safety constraints (τ_{merge} and τ_{sat}) help reduce semantic drift, smaller models may still oversimplify complex reasoning during compression.

Domain Scope. We evaluate arithmetic (GSM8K), commonsense (StrategyQA), and scientific reasoning (ARC-Challenge). The notion of efficiency may differ in more open-ended domains, such as creative writing, where verbosity can serve stylistic or communicative purposes.

Hyperparameter Transfer. Although CAID is reasonably stable across our benchmarks with a fixed set of thresholds, transferring it to domains with substantially different linguistic characteristics (e.g., code or legal text) may require additional calibration.

Ethics Statement

This work is motivated in part by the goal of improving the efficiency of LLM reasoning, which may help reduce unnecessary computational cost. At the same time, we note several ethical considerations.

Data and Privacy. All experiments use publicly available datasets. Our study does not involve private or personally identifiable information. The synthetic perturbations in RIV-GSM8K were generated with GPT-4o for research purposes and were designed to remain task-relevant and non-harmful.

Potential Biases. Efficiency-oriented metrics may penalize linguistic styles that are longer or more elaborative by convention. As a result, care is needed when applying CAID beyond the benchmark settings considered here, especially in contexts where verbosity may reflect dialectal, stylistic, or communicative variation rather than inefficiency.

Use of AI Assistants. GPT-4o was used in the data augmentation pipeline to generate synthetic reasoning perturbations for RIV-GSM8K. AI assistants were also used for preliminary code drafting and language editing. The authors reviewed all such outputs and take full responsibility for the final manuscript.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2022-00143911, AI Excellence Global Innovative Leader Education Program).

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*.

Joonwon Jang, Jaehee Kim, Wonbin Kweon, Seonghyeon Lee, and Hwanjo Yu. 2025. Verbosity-aware rationale reduction: Sentence-level rationale reduction for efficient and effective reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20769–20784, Vienna, Austria. Association for Computational Linguistics.

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*.

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.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamile Lukosiute, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *Preprint*, arXiv:2307.13702.

Jinu Lee and Julia Hockenmaier. 2025. Evaluating step-by-step reasoning traces: A survey. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1789–1814, Suzhou, China. Association for Computational Linguistics.

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*.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025. 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)*, pages 25299–25346, Vienna, Austria. Association for Computational Linguistics.

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.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *Preprint*, arXiv:2502.18600.

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 $\mathcal{D}_{RIV}$

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

A.2 Dataset Statistics

Table 5 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).

B Prompt Details and Configuration

We utilize `gpt-4o` for all augmentation tasks. To ensure generation diversity and structural consistency, 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 6. These examples are derived from a single original step in the GSM8K dataset: *"Adults = 10 * 8 = 80"*.

As shown in Table 6, 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) to compute cosine similarity for Local Similarity ($\mathcal{M}_{sim}$) and Global Goal Alignment

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

($\mathcal{M}_{rel}$). 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).

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 8 presents concrete examples from the GSM8K dataset.

F Detailed Ablation on Compression Strategy

In this section, we provide the extended ablation study on the compression strategies employed in PACE, validating the necessity of the MERGE action and safety constraints.

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 7: 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.

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 8: 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).

Deletion vs. Compression. As shown in Table 7, 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.

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.

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).