When Compression Helps and When It Hurts: Condition-Aware Analysis of Chain-of-Thought Distillation

Siyang Lyu[1,2]*, **Xinghao Chen**[1,3]*, **Zhijing Sun**[1,2],
Tong Liu[4], **Dawei Zhu**[5], **Xiaoyu Shen**[1]†
[1]Zhejiang Key Laboratory of Industrial Intelligence and Digital Twin, EIT, Ningbo
[2]Viterbi School of Engineering, University of Southern California
[3] The Hong Kong Polytechnic University [4] LMU Munich
[5] Saarland University, Saarland Informatics Campus
siyangly@usc.edu xyshen@eitech.edu.cn

Abstract

Chain-of-Thought (CoT) distillation transfers multi-step reasoning from large reasoning models to smaller students, but verbose teacher traces inflate both training and inference cost. Existing CoT compression methods fall into two families, selective pruning and generative rewriting, yet prior studies have left key factors entangled: granularity is confounded with importance criteria in pruning, restructuring level is rarely isolated in rewriting, and compression budgets are not systematically evaluated across domains or regimes. We recast CoT compression along three dimensions: importance criterion, restructuring level, and compression budget. Sweeping these across two model families, Math and General domains, and Long-/Short-CoT regimes, we find that (i) importance criterion utility is strictly governed by granularity: step-level criteria converge on a shared reasoning backbone, while token-level pruning requires symbol-aware signals to preserve the logical core; (ii) restructuring level inverts across domains: Math degrades monotonically with structural disruption, while aggressive rewriting acts as a denoiser on General tasks; (iii) training-time compression does not necessarily translate to inference-time savings: Long-CoT students retain verbose habits despite concise supervision, making the training ratio an optimistic lower bound on deployment cost. These findings yield condition-aware guidelines for matching compression to deployment context.[1]

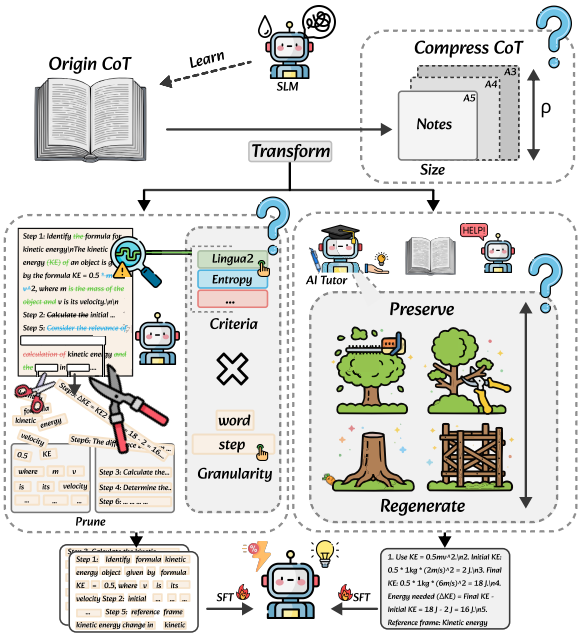

Figure 1: The CoT Compression Pipeline. To reduce computational costs, verbose reasoning is compressed by pruning low-value words or regenerating core logic, producing concise training data for efficient SFT.

1 Introduction

Large reasoning models (LRMs, Chen et al. 2026a) have recently achieved strong performance on complex reasoning tasks (DeepSeek-AI et al., 2025; OpenAI, 2024b). Because directly deploying these models is costly, a common strategy is to distill their reasoning traces into smaller student models (Huang et al., 2024; Ye et al., 2025). However, these traces are often verbose, containing redundant verification, speculative detours, and repeated reformulations (Chen et al., 2025c; Wang et al., 2025c; Zhen et al., 2025). Such verbosity increases training-time token consumption and may also teach students to generate unnecessarily long reasoning traces at inference time. CoT compression therefore aims to shorten reasoning supervision while preserving the information needed for effective distillation.

Existing CoT compression methods generally fall into two families. *Selective pruning* deletes parts of the original trace using saliency, uncertainty, surprisal, or related importance scores (Pan et al., 2024; Xia et al., 2025; Li et al., 2026;

*Equal Contributions.
†Corresponding authors.
[1]The code will be released at https://github.com/EIT-NLP/CoT-Distillation-Compression.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 35204–35231
October 24-29, 2026 ©2026 Association for Computational Linguistics

Zhao et al., 2025b; Xu et al., 2025b). *Generative rewriting* produces a new shorter rationale through paraphrasing, step fusion, or structured regeneration (Yu et al., 2025; Kang et al., 2025; Wu et al., 2025; Jiang et al., 2026). While both families successfully reduce cot length, current research still lacks a unified understanding of *what underlying factors actually drive effective compression*. In particular, three critical issues remain unresolved.

First, for *selective pruning* methods, the role of *granularity* is insufficiently understood. Existing approaches typically operate either at the step level (Li et al., 2026; Zhao et al., 2025b) or at the token level (Xia et al., 2025), but granularity is often entangled with the importance criterion itself. As a result, it remains unclear whether performance differences arise from the scoring signal or from the level at which compression is applied.

Second, for *generative rewriting* methods, the role of *restructuring level* is rarely studied in isolation. Existing rewriting approaches implicitly introduce different degrees of structural transformation (Yu et al., 2025; Kang et al., 2025; Wu et al., 2025), ranging from light paraphrasing to aggressive regeneration of reasoning traces. However, prior work neither isolates restructuring level as an independent variable nor controls it against compression ratio, making it difficult to determine whether performance differences stem from information preservation or structural disruption.

Third, across both pruning and rewriting methods, the role of *compression budget* lacks systematic evaluation. Existing studies typically report token reduction and downstream accuracy under a single task domain or compression setting, without examining how compression behavior changes across domains, reasoning regimes, or compression intensities (Jiang et al., 2026; Wu et al., 2025; Wang et al., 2025b).

Collectively, these gaps reduce CoT compression to trial-and-error tuning rather than principled design. In this work, we address these limitations by reframing CoT compression as a controlled design problem along three dimensions (Figure 1): *importance criterion*, which determines what reasoning content is retained; *restructuring level*, which characterizes how strongly the original reasoning structure is modified; and *compression budget*, which distinguishes the target compression ratio, the realized training ratio, and the actual inference-time token ratio during deployment. We systematically study these dimensions across two model families, two task domains, and both Short-CoT and Long-CoT supervision regimes.

Our analysis reveals the following principal findings. **(i)** Importance criterion utility is strictly governed by granularity: at the step level, distinct criteria converge on a shared reasoning backbone, rendering criterion choice largely irrelevant; at the token level, this robustness collapses, and only symbol-aware signals preserve the logical core under aggressive pruning. **(ii)** The effect of restructuring level inverts across domains: on Math, accuracy degrades monotonically as structural disruption increases, leaving structure-preserving methods as the only reliable choice; on General tasks, aggressive rewriting acts as a denoiser, matching or exceeding structure-preserving baselines. **(iii)** Training-time compression does not necessarily translate into inference-time efficiency. Under Short-CoT supervision, students learn genuinely concise reasoning patterns and achieve real deployment savings. Under Long-CoT supervision, however, students retain verbose reasoning behaviors even when trained on compressed traces, causing the realized inference cost to remain substantially higher than the nominal training compression ratio.

These findings lead to a practical guideline for choosing CoT compression strategies. For symbolic or math-heavy tasks, compression should preserve the original reasoning structure, and token-level pruning should use symbol-aware criteria that protect constraints, operators, and intermediate values. For general-domain reasoning, more aggressive semantic fusion or framework-level rewriting can be beneficial because it can remove noise without destroying a unique derivation path. When deployment efficiency is the goal, practitioners should measure the student's actual inference length rather than relying only on compressed training-token counts. Overall, our study shows that effective CoT compression is not a universal token-reduction recipe, but a set of condition-aware choices that must be matched to the structure of the task and the intended deployment setting.

2 Related Works

Overthinking in LRMs Recent LRMs, like OpenAI's o1 (OpenAI, 2024b) and DeepSeek-R1 (DeepSeek-AI et al., 2025), have demonstrated strong complex-task performance. Distilling their reasoning traces enables smaller models to acquire reasoning abilities (Huang et al., 2024). However,

distillation also transfers the teacher's overthinking behavior (Chen et al., 2025c)—redundant verification, speculative detours, and repetitive restates that inflate both training-time token budgets and inference-time latency without significant scaling benefits in student performance (Wang et al., 2025a; Chen et al., 2025a; Wang et al., 2025c). Motivated by this, recent work on efficient CoT explores ways to retain the utility of explicit reasoning while reducing its cost (Sui et al., 2025). A complementary line of work tackles overthinking from a model-architecture perspective, examining the effectiveness of pruning strategies on reasoning-augmented LRMs (Ding et al., 2026) and developing dynamic layer pruning that adaptively allocates computation per token (Zhao et al., 2025a). In parallel, a growing body of work studies **CoT compression** by pruning intermediate steps or compressing teacher traces via summarization and rewriting, thereby obtaining cheaper supervision for student training (Zhu and Li, 2025; Sui et al., 2025).

Trace-level CoT Compression: Selective Pruning and Generative Rewriting One thread, *selective preservation and rationale pruning*, retains critical trace segments: saliency-based estimators isolate informative spans to remove fluff (Pan et al., 2024; Xia et al., 2025); information-theoretic signals, uncertainty or surprisal, identify essential reasoning blocks (Li et al., 2026; Cui et al., 2025; Xu et al., 2025b); and dependency-aware frameworks preserve logically indispensable components by modeling inter-step relations through graphs or heuristic rules (Zhao et al., 2025b; Jia et al., 2025). A complementary thread, *generative refinement and structural re-organization*, paraphrases and condenses teacher traces across a transformation spectrum—from skeleton-preserving polishing (Yu et al., 2025), to step-merging into abstract statements (Jiang et al., 2026; Kang et al., 2025), and intent-guided rewrites that replace the teacher template (Wu et al., 2025). Beyond discrete trace editing, recent work has also explored shifting reasoning into continuous latent spaces (Chen et al., 2025b), with particular attention to designing effective supervision signals for latent trajectories (Chen et al., 2026b). Orthogonal to trace editing, *length control* constrains or predicts reasoning length, from lightweight inference-time constraints (Xu et al., 2025a; Ding et al., 2024) to budget-aware methods that distill budget-aligned traces as supervision to internalize efficiency (Han et al., 2025). Our work systematically characterizes this compression spectrum to map the trade-offs between structural fidelity and distillation performance.

3 Experimental Setup

3.1 Problem Formulation

We represent CoT compression as:

$$\tilde{r}_i = \mathcal{C}(r_i; \rho, \phi), \tag{1}$$

which maps a teacher trace r to a compressed version $\tilde{r}$ for fine-tuning. Here, ρ denotes the compression level and ϕ captures methodological choices.

Design Dimensions We organize methods and experiments along three dimensions:

(i) **Importance criterion** $\phi_{\text{importance}}$: Determines <u>what</u> content is removed via a score-threshold-prune pipeline: each unit (token or step) is assigned an importance score, and those below a threshold derived from the target compression ratio are pruned. We instantiate three importance scores: LLMLingua2 (learned saliency), Entropy (uncertainty), and NLL (surprise).

(ii) **Restructuring level** ϕ_{restruct}: Characterizes <u>how</u> the trace is restructured, along a spectrum from surface trimming to template reconstruction. We use Gemini 2.5 Flash (Google DeepMind Gemini Team, 2025) as the rewriter to span four levels: structural preservation (**L1: SP**), step editing (**L2: SE**), semantic fusion (**L3: SF**), and framework reconstruction (**L4: FR**) [2].

(iii) **Compression Level** ρ**:** This factor governs the aggressiveness of supervision degradation. We manipulate a target ratio ρ_{target} to observe the realized ratio ρ_{realized}–the actual trace length–which ultimately determines the end-to-end inference token ratio $\rho_{\text{inference}}$ as the deployment cost. This factor applies universally: as a threshold parameter for pruning or a length constraint for rewriting.

Objective and Evaluation. A student model M_θ is fine-tuned to minimize the standard negative log-likelihood on the compressed data $(x, \tilde{r}, y)$, derived from a specific teacher regime $\mathcal{R} \in \{\text{long}, \text{short}\}$.

We evaluate the resulting impact surface defined by task performance (Perf) and inference token cost (Cost) across domains d:

$$\big(\text{Perf}, \text{Cost}\big) = f(\rho, \phi, \mathcal{R}; d), \tag{2}$$

[2]Prompts and comparative cases are in Appendix A and Appendix K. We additionally verify using GPT-4o-mini (OpenAI, 2024a) in Appendix D.4

where $\mathcal{R}$ denotes the training-trace regime (long/short) and d the evaluation domain.

3.2 Models, Data, and Metrics

Student Models. We employ LLaMA-3.1-8B-Instruct (Grattafiori et al., 2024) as the primary student model, with the Qwen2.5 series (Yang et al., 2024) used to verify cross-model and scale generalization; full results are provided in Appendix E.

Training Corpora. To analyze the impact of reasoning length, we construct two training sets: a Short-CoT regime sampled from Natural Reasoning (Yuan et al., 2025) (5k instances), and a Long-CoT regime from LIMO (Ye et al., 2025).

Evaluation Benchmarks. The evaluation spans five benchmarks in two domains: (1) Math: GSM8K, MATH500 for *short-CoT* regime and MATH500, AMC23, AIME24 for *long-CoT* regime;(2) General: GPQA Diamond and MMLU-PRO for both regimes.

Evaluation Metrics. We report Accuracy for reasoning quality and the Inference Token Ratio ($\rho_{\text{inference}}$)—the ratio of generated tokens relative to the uncompressed baseline—for cost efficiency.

Implementation Details We fine-tune all models via LoRA (Hu et al., 2022). Evaluation uses a standardized greedy-like decoding with temperature 0.7. Full hyperparameters, prompt templates for rewriting, and importance score definitions are provided in Appendix C.1.

4 Effects of Importance Criterion

Whether CoT compression helps hinges on a simple premise: teacher traces must contain content that can be removed without changing the final performance, and the pruning mechanism must be able to target such content without discarding globally critical information. Rather than estimating intrinsic information redundancy directly, we adopt an operational definition of compression tolerance: the capacity of a model to maintain downstream accuracy under increasing pruning rates.

4.1 Empirical compression tolerance

Long-CoT training exhibits higher empirical tolerance. As shown in Figure 2, Long-CoT training yields a larger budget of removable tokens: student accuracy stays stable across a wider range of compressing ratios compared to Short-CoT. This is not because long traces contain a specific, easily-identifiable class of "redundant" steps, such as reflective statements(Table 1) as suggested by prior work (Jia et al., 2025). In fact, in our controlled experiments (Appendix F), pruning methods that deliberately target reflective statements perform no better than those that ignore them. Instead, redundancy in Long-CoT is diffuse: the critical reasoning skeleton is repeated and rephrased throughout the trace. Therefore, as long as that global skeleton survives, moderate compression is robust, irrespective of which specific elaborations are removed.

Setting	Reflective steps (%)
Short-CoT	0.57%
Long-CoT	30.2%

Table 1: Content-level redundancy proxies for short- vs. long-CoT traces. The percentage of reflective or self-correction steps per trace (e.g., "wait", "let me check").

Signal	Symbol / math	Stopwords	Other content
Entropy	0.30	0.29	0.21
NLL	0.30	0.28	0.21
LLMLingua	0.15	0.54	0.11

Table 2: The **realized** 0.3 target deleted ratio for NATURALREASONING under token-level pruning. Compared with entropy/NLL, LLMLingua deletes fewer constraint-bearing tokens, shifting deletions toward stopwords.

Domain-Dependent Sensitivity. General benchmarks sustain accuracy under much stronger pruning than Math benchmarks. This is because General tasks typically accept diverse valid rationales and do not hinge on a specific intermediate state. Consequently, performance slowly degrades even when the training signal is partially corrupted.

Math tasks, by contrast, require internally consistent intermediate quantities and precise constraint satisfaction. Small errors introduced by compression are amplified down the reasoning chain, leading to earlier and sharper accuracy drops under aggressive pruning [3].

4.2 Importance Score Quality and Token Preservation

LLMLingua2 yields more stable pruning under high-risk compression. At matched realized compression ratios, LLMLingua2-based pruning degrades more smoothly and suffers fewer sharp accuracy drops than entropy or NLL baselines. The gap is largest on Math benchmarks and under the strongest compression.

[3] Representative failure cases are in Appendix L

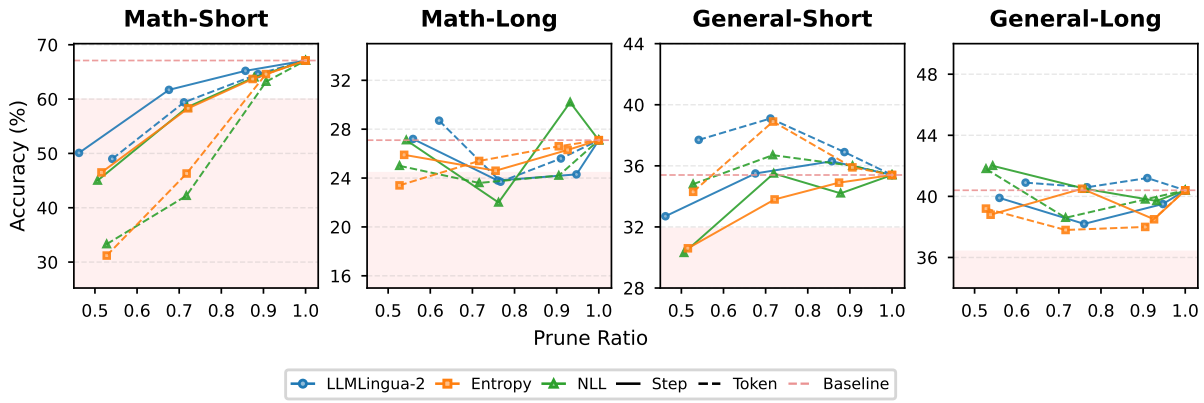

Figure 2: Accuracy of compressed students versus realized prune ratio for extraction-based pruning.

Pair	0.1	0.3	0.5
ENT–L2	0.75	0.53	0.36
NLL–L2	0.76	0.54	0.38
ENT–NLL	0.92	0.85	0.78

Table 3: Pairwise Jaccard overlap between preserved-step sets for different step-level signals under matched target ratios. (ENT: Entropy; L2: LLMLingua2)

To understand why, we analyze which tokens each method deletes. We classify removed tokens into three categories: (i) math-like patterns (operators, numbers, logic markers), (ii) stopwords, and (iii) other content. Table 2 shows that entropy and NLL delete a much larger fraction of math-like tokens than LLMLingua2, which concentrates its deletions on stopwords and other content. A full breakdown is in Appendix G.

Deleting math-like tokens is directly harmful. Failure cases [4] show that their removal correlates with catastrophic reasoning failures. Crucially, a controlled ablation [5] demonstrates that if we simply prevent entropy/NLL from deleting math tokens, their performance gap to LLMLingua2 largely disappears. This confirms that LLMLingua2's advantage comes specifically from preserving symbolic anchors.

Thus, in high-risk extraction, the quality of the importance score dictates stability. Unlike probability-based heuristics that risk fracturing logical steps, LLMLingua2 functions as a symbol preserving filter, saving the symbolic anchors essential for valid derivation.

4.3 Granularity Modulates Importance Criterion Utility

The utility of an importance criterion depends largely on the granularity of the deletion unit.

Step-level pruning: Convergence on the backbone. Despite their different scoring metrics, step-level strategies based on Entropy, NLL, and LLMLingua2 yield similar accuracy. We analyze the pairwise Jaccard overlap of their pruning decisions, as reported in Table 3. The key insight is that the criteria agree strongly on which steps to preserve, where the overlap is 70–80%, but disagree substantially on which steps to delete [6]. This means that all criteria converge on the same critical reasoning backbone. The remaining steps are peripheral and interchangeable; removing one versus another has little effect. Hence, step-level pruning is robust to the choice of importance criterion.

Token-level pruning: importance score quality matters. In stark contrast, token-level pruning is highly sensitive to importance score quality. Under strong compression, LLMLingua2 maintains a smooth accuracy curve, while Entropy and NLL baselines collapse, especially on Math. This divergence comes down to whether the criterion preserves local structural integrity. As shown in Section 4.2, LLMLingua2 preferentially retains symbolic anchors and syntactic structures. Probability-based heuristics often indiscriminately delete key operators and numbers alongside stopwords. This risks fragmenting the internal logic of a reasoning step rather than trimming verbose phrasing, a pattern we confirm via coherence analysis [7].

[4] Representative failure cases are in Appendix L
[5] Results are in Appendix G.3
[6] Details are in Appendix H
[7] Results and analysis details are in Appendix I

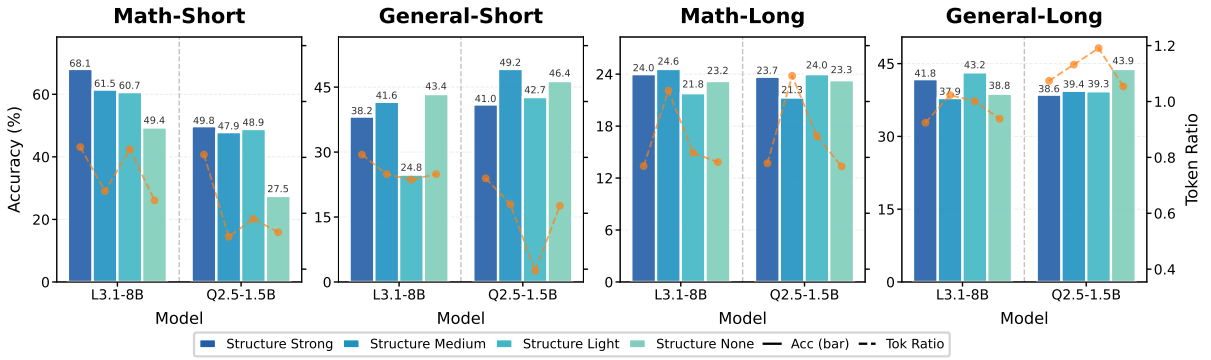

Figure 3: Accuracy and compression ratio across restructuring levels. Math exhibits a strict monotonic dependency on restructuring level (monotonic decay from SP to FR), whereas General domains show a restructuring benefit, where aggressive rewriting (SF/FR) matches or exceeds structure-preserving baselines.

Conclusion

Importance criterion utility is governed by granularity. At step level, all criteria converge on a shared reasoning backbone, making criterion choice irrelevant. At token level, this robustness collapses: symbol-aware criteria, like LLMLingua2, thin redundancy without shattering the logical core, while probability-based heuristics risk removing constraint-bearing tokens.

5 Effect of Restructuring Level

CoT compression changes not only length but also the organization of the reasoning chain. We use the **restructuring level** to describe how much a compressed CoT preserves the teacher's step segmentation, local ordering, and high-level solution plan. We define a qualitative spectrum — from minimal to maximal perturbation — to characterize how each rewriting method restructures the reasoning path, while keeping compression ratio comparable. Each level below represents an existing class of rewriting strategies, with representative work cited in line with prompts and comparative cases in Appendix A and Appendix K:

- **L1: Structural Preservation (SP):** The rewriter is constrained to only remove redundant words to target surface verbosity, strictly preserving the verbatim logical sequence (Yu et al., 2025).
- **L2: Structural Editing (SE):** The rewriter performs step-level operations like keeping, deleting or rephrasing to create a more efficient but structurally similar reasoning path (Jiang et al., 2026).
- **L3: Semantic Fusion (SF):** The rewriter is encouraged to retain only "key information" to fuse multi-step sequences into novel statements (Kang et al., 2025).
- **L4: Framework Replacement (FR):** The rewriter discards the original structure to regenerate reasoning via a predefined formal template (Wu et al., 2025).

5.1 Robustness Across Restructuring Level

As illustrated in Figure 3, a consistent pattern along this spectrum: *across domains, lower restructuring levels are more robust as compression strengthens, whereas higher restructuring levels deteriorate earlier and show more domain-dependent variability.*

Robustness Degrades with Restructuring Level From SP to SE to SF as the restructuring level increases: at light to moderate compression, SP/SE/SF typically reduce rationale length while staying near the original baseline, particularly on General tasks. When compression becomes strong, however, SE and SF drop faster, while SP remains comparatively stable. A plausible explanation is that SP/SE primarily shorten surface form while retaining the step skeleton, whereas SF already begins to fuse across steps; further compression then more easily weakens information that cannot be reconstructed from the remaining text. At the highest restructuring level, framework replacement (FR) pushes this unstable effect further by rewriting the entire solution template, and we frequently observe higher variance across settings.

5.2 Domain Sensitivity and Interaction

Math Reasoning Demands Minimal Restructuring. In LLM-based summarization, we observe a domain divergence. On Math, SF and SE degrade significantly faster than SP under strong compression, making SP the safest choice, as illustrated by representative failure cases in Appendix L. In Math

domains, Accuracy declines monotonically as the restructuring level increases from SP to FR. This fragility stems from the strict logical dependencies in mathematical reasoning: errors introduced by disrupting step boundaries through Semantic Fusion (L3) or by editing and rephrasing steps Structural Editing (L2) are amplified downstream. Conversely, General tasks often tolerate a wider range of valid rationales. Methods at higher restructuring levels remain competitive because these tasks are less sensitive to how intermediate steps are packaged, allowing for more aggressive abstraction.

FR: The High-Variance Extremity. FR amplifies this domain split. On Math, it is among the weakest strategies; on General, it frequently reaches parity with, or even surpasses, structure-preserving methods under tight budgets. As shown in Figure 3, the Framework Replacement approach collapses on logic-heavy Math problems but remains competitive in General domains. We attribute this to the rigidity of the solution space. Imposing a predefined framework (L4) on Math problems disrupts the precise logical flow of the original rationale, leading to derivation errors. In contrast, on General tasks, template-level rewriting can effectively reorganize and denoise the trace, yielding a more focused rationale.

Conclusion

The effect of restructuring level inverts across domains: on Math, robustness degrades monotonically with restructuring level, leaving SP the only reliable choice. In General, this monotonicity breaks down: aggressive restructuring (SF/FR) can match or exceed structure-preserving baselines.

6 Effect of Compression Budget

Building on the compression setup in Section 3, we characterize compression dynamics via three metrics: the target ratio ρ_{target} as the control knob, the empirical realized ratio ρ_{realized} reflecting actual compression on CoT traces, and the inference token ratio $\rho_{\text{inference}}$ measuring deployment cost. Beyond fixed-ρ_{target} constraints, we also evaluate a free ratio mode, where the rewriter optimizes ρ_{realized} naturally within a fixed structural template without a prescribed target.

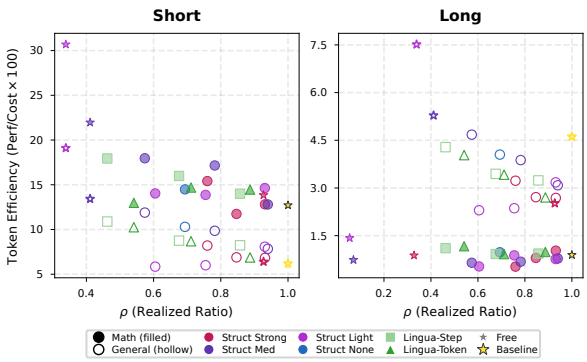

Figure 4: Token efficiency (Perf/Cost) vs. Realized Ratio (ρ_{realized}), illustrating how structural fidelity defines length bands while the Free Ratio (stars) naturally identifies high-efficiency equilibrium points.

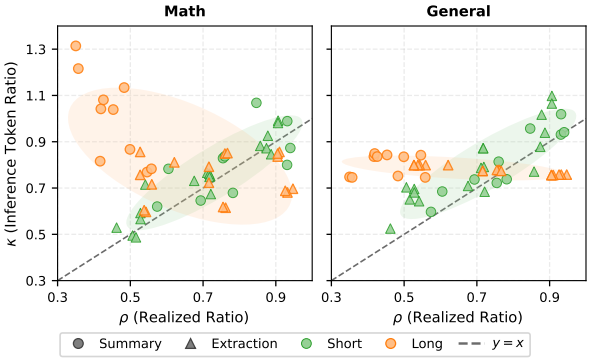

Figure 5: Efficiency translation analysis. Short-CoT aligns with the $y = x$ diagonal while Long-CoT exhibits an efficiency gap ($\rho_{\text{inference}} > \rho_{\text{realized}}$).

6.1 Stability and Performance Decay

Our empirical analysis reveals that model accuracy remains remarkably resilient, staying within 90% of the original baseline across a significant compression interval. However, as shown in Figure 6, this stability is eventually followed by a sharp nonlinear degradation. Once compression exceeds a critical threshold, performance collapses rapidly, suggesting a fundamental limit to how much information can be discarded before the underlying reasoning logic is compromised.

The compression resilience is modulated by two distinct factors. Rather than a uniform decay, we observe that mathematical reasoning tasks are inherently more fragile than general-domain benchmarks, yet this sensitivity is significantly counteracted by long-CoT teachers, which systematically extend the viable compression range.

6.2 Target vs. Realized Compression

Given the observed resilience thresholds, we evaluate two distinct compression styles and employ a

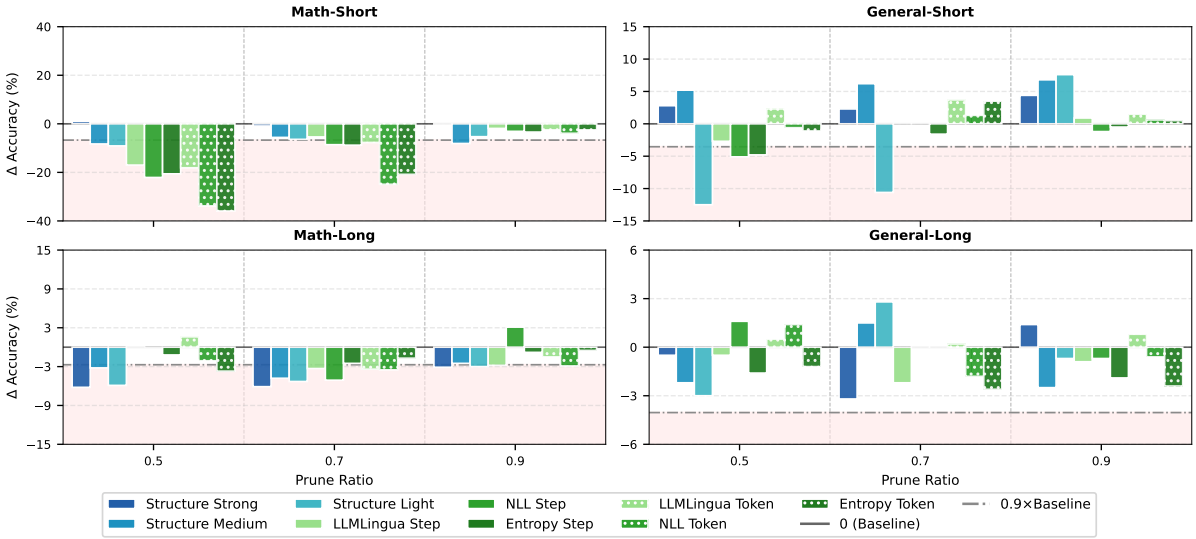

Figure 6: Comparison of summarization- and extraction-based CoT compression across domains and training regimes. Each subplot reports accuracy under different prune ratios on Math/General × Short/Long settings. Bars show accuracy at prune ratios $0.5/0.7/0.9$, and the orange line shows the corresponding inference token ratio.

length-band analysis to characterize the resulting variability in token reduction.

Summarization as intent-driven rewriting. Unlike extractive methods, summarization modulates length indirectly via its rewriting objective. We observe that realized lengths do not scale linearly with target ratios; instead, the teacher LLM, constrained by linguistic fluency and structural coherence, converges to method-specific characteristic lengths. Thus, the target ratio signals compression intensity rather than a precise numerical constraint.

Structural constraints define the length band. As shown in Figure 4, the choice of structural fidelity dictates the achievable compression range (the band), while the target ratio only facilitates marginal adjustments within that band.

In High-fidelity methods, such as SP, strict structural constraints force the model to preserve the original reasoning skeleton, yielding a narrow, conservative band. Looser prompts methods such as SF/SE, permissive constraints allowing cross-step fusion enable the model to fold multiple steps into single spans, shifting the band toward smaller ratios and widening its variance. This suggests that stronger structural constraints yield stable but conservative results, while weaker constraints enable more aggressive but volatile compression.

Free Ratio as a Natural Equilibrium. The free-ratio represents a method's intrinsic set-point absent of rigid constraints. Figure 4 confirms that this "natural" landing point often yields superior token efficiency, appearing as "star" points on the optimal frontier. In structure-preserving methods, this mechanism serves as a self-regularizer, ensuring the model settles at stable, moderate ratios. Although less protective for aggressive variants with naturally lower bands, the free ratio avoids the instability of forced compression. Thus, we propose using the free ratio as a primary configuration for ensuring consistent performance.

6.3 The Gap Between Training and Inference

We observe a positive correlation between the end-to-end inference token ratio and realized compression ratio. As shown in Figure 5, this correlation is tight in the short-CoT regime but weakens under long-CoT regime. Generally, ρ_{target} serves as a viable proxy, but the efficiency translation varies significantly by regime. Short-CoT exhibits a near-linear correlation ($\rho_{\text{inference}} \approx \rho_{\text{realized}}$) along the diagonal, indicating successful internalization of concise reasoning styles. In contrast, the Long-CoT regime reveals a significant efficiency gap where $\rho_{\text{inference}} > \rho_{\text{realized}}$. This divergence stems from a translation loss where verbose habits persist despite concise training supervision. Thus, for complex tasks, ρ serves as an optimistic lower bound rather than a precise predictor of deployment cost. For practical use, Appendix B consolidates the findings from Sections 4–6 into a condition-aware decision guide.

Conclusion

As compression budget decreases, accuracy degrades as a "plateau-then-drop", with resilience shaped by domain and source-CoT regime. More critically, efficiency translation is regime-conditional: Short-CoT compresses faithfully, while Long-CoT does not, as verbose habits persist despite concise supervision, making ρ an optimistic lower bound for deployment cost.

7 Conclusion

We have reframed CoT compression into three design dimensions: compression budget, importance criterion, and restructuring level. Each is governed by a distinct factor. Whether training-side compression translates to shorter inference depends on the source-CoT regime, with Long-CoT students retaining verbose habits despite concise supervision. Criterion utility is governed by granularity: step-level methods converge on a shared backbone, while token-level pruning demands symbol-aware criteria such as LLMLingua2 to preserve the logical core. Restructuring level acts in opposite directions across domains, denoising General tasks but degrading Math monotonically. Together, these results show that compression efficacy is conditioned on regime, granularity, and domain; no single strategy universally dominates. Effective deployment requires aligning each design choice with its specific task context and inference constraints.

Limitations

Our experiments adopt LoRA-based fine-tuning as the training objective, as our goal is to understand how different compression strategies alter the supervision signal itself rather than to compensate for supervision degradation through stronger distillation objectives. Full-parameter fine-tuning and more aggressive objectives remain important directions for future work. Furthermore, our Math/General partition is a pragmatic dichotomy that isolates two prototypical classes of structural sensitivity rather than an exhaustive taxonomy of reasoning task types. More specific task types, such as code generation, deep-search reasoning, and multi-hop problems, may exhibit hybrid sensitivity combining symbolic dependencies with more flexible solution spaces, suggesting that the General category could be further decomposed along structural properties. We leave such a finer-grained taxonomy as a concrete direction for future work.

Ethics Statement

The datasets utilized in this study are publicly available benchmarks. We acknowledge, however, that public web-sourced data may inherently contain socially harmful or toxic language, despite our inability to guarantee its complete absence. Our experiments rely on the Google Gemini API; at no point did we access or attempt to access the proprietary training data or underlying system weights of the models. We confirm that the datasets used do not contain personally identifiable information (PII), ensuring user privacy. All datasets and models were used in strict accordance with their permissive licenses and intended usage guidelines. We believe this work adheres to the ACL Code of Ethics. Finally, we employed GPT-5[8]to correct grammatical errors in this paper.

Acknowledgments

We thank the Center for Advanced Research Computing (CARC) at the University of Southern California for providing the computational resources used in this work.

References

Art of Problem Solving. n.d. AIME problems and solutions. Accessed: 2025-12-23.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In Proceedings of the ACL Interactive Poster and Demonstration Sessions, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2026a. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. Science China Information Sciences, 69(6):161101.

Xinghao Chen, Chak Tou Leong, Wenjin Guo, Jian Wang, Wenjie Li, and Xiaoyu Shen. 2026b. What makes effective supervision in latent chain-of-thought: An information-theoretic analysis. In Forty-third International Conference on Machine Learning.

Xinghao Chen, Zhijing Sun, Guo Wenjin, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and Xiaoyu Shen. 2025a. Unveiling the key factors for distilling chain-of-thought reasoning. In Findings of the Association

[8] https://chatgpt.com/

for Computational Linguistics: ACL 2025, pages 15094–15119, Vienna, Austria. Association for Computational Linguistics.

Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. 2025b. Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. Preprint, arXiv:2505.16782.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025c. Do NOT think that much for 2+3=? on the overthinking of long reasoning models. In Proceedings of the 42nd International Conference on Machine Learning, volume 267 of Proceedings of Machine Learning Research, pages 9487–9499. PMLR.

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. CoRR, abs/2110.14168.

Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, Suhang Wang, Yue Xing, Jiliang Tang, and Qi He. 2025. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. In Findings of the Association for Computational Linguistics: ACL 2025, pages 18581–18597, Vienna, Austria. Association for Computational Linguistics.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In International Conference on Learning Representations.

DeepSeek-AI. 2026. DeepSeek-V4: Towards highly efficient million-token context intelligence. Preprint, arXiv:2606.19348.

DeepSeek-AI, Daya Guo, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. Preprint, arXiv:2501.12948.

Longwei Ding, Anhao Zhao, Fanghua Ye, Ziyang Chen, and Xiaoyu Shen. 2026. From llms to lrms: Rethinking pruning for reasoning-centric models. Preprint, arXiv:2601.18091.

Mengru Ding, Hanmeng Liu, Zhizhang Fu, Jian Song, Wenbo Xie, and Yue Zhang. 2024. Break the chain: Large language models can be shortcut reasoners. CoRR, abs/2406.06580.

Google DeepMind Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. Preprint, arXiv:2507.06261.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. 2024. The Llama 3 herd of models. Preprint, arXiv:2407.21783.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware LLM reasoning. In Findings of the Association for Computational Linguistics: ACL 2025, pages 24842–24855, Vienna, Austria. Association for Computational Linguistics.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022. OpenReview.net.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. 2024. O1 replication journey – part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? Preprint, arXiv:2411.16489.

Jinghan Jia, Hadi Reisizadeh, Chongyu Fan, Nathalie Baracaldo, Mingyi Hong, and Sijia Liu. 2025. EPiC: Towards lossless speedup for reasoning training through edge-preserving cot condensation. Preprint, arXiv:2506.04205.

Yuxuan Jiang, Dawei Li, and Francis Ferraro. 2026. DRP: Distilled reasoning pruning with mathematical skill-aware step decomposition for efficient large reasoning models. In Findings of the Association for Computational Linguistics: ACL 2026, pages 4020–4039, San Diego, California, United States. Association for Computational Linguistics.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3oT: Generating shorter chain-of-thought without compromising effectiveness. Proceedings of the AAAI Conference on Artificial Intelligence, 39(23):24312–24320.

Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. 2026. Making slow thinking faster: Compressing LLM chain-of-thought via step entropy. In The Fourteenth International Conference on Learning Representations.

Mathematical Association of America. 2025. American Mathematics Competitions (AMC). Accessed: 2026-01-06.

OpenAI. 2024a. GPT-4o system card. Preprint, arXiv:2410.21276.

OpenAI. 2024b. Learning to reason with LLMs. Accessed: 2025-12-30.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In Findings of the Association for Computational Linguistics: ACL 2024, pages 963–981, Bangkok, Thailand. Association for Computational Linguistics.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '20). IEEE/ACM.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In First Conference on Language Modeling.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. Transactions on Machine Learning Research.

Rui Wang, Hongru Wang, Boyang Xue, Jianhui Pang, Shudong Liu, Yi Chen, Jiahao Qiu, Derek Fai Wong, Heng Ji, and Kam-Fai Wong. 2025a. Harnessing the reasoning economy: A survey of efficient reasoning for large language models. Preprint, arXiv:2503.24377.

Yibo Wang, Li Shen, Huanjin Yao, Tiansheng Huang, Rui Liu, Naiqiang Tan, Jiaxing Huang, Kai Zhang, and Dacheng Tao. 2025b. R1-Compress: Long chain-of-thought compression via chunk compression and search. Preprint, arXiv:2505.16838.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In Advances in Neural Information Processing Systems, volume 37, pages 95266–95290.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025c. Thoughts are all over the place: On the underthinking of long reasoning models. In Advances in Neural Information Processing Systems, volume 38.

Yifan Wu, Jingze Shi, Bingheng Wu, Jiayi Zhang, Xiaotian Lin, Nan Tang, and Yuyu Luo. 2025. Concise reasoning, big gains: Pruning long reasoning trace with difficulty-aware prompting. Preprint, arXiv:2505.19716.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. TokenSkip: Controllable chain-of-thought compression in LLMs. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 3351–3363, Suzhou, China. Association for Computational Linguistics.

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

Xiaoang Xu, Shuo Wang, Xu Han, Zhenghao Liu, Huijia Wu, Peipei Li, Zhiyuan Liu, Maosong Sun, and Zhaofeng He. 2025b. A-Thought: Efficient reasoning via bidirectional compression for low-resource settings. In Advances in Neural Information Processing Systems, volume 38.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. CoRR, abs/2412.15115.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. LIMO: Less is more for reasoning. In Conference on Language Modeling.

Bin Yu, Hang Yuan, Haotian Li, Xueyin Xu, Yuliang Wei, Bailing Wang, Weizhen Qi, and Kai Chen. 2025. Long-short chain-of-thought mixture supervised fine-tuning eliciting efficient reasoning in large language models. Preprint, arXiv:2505.03469.

Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilia Kulikov, Kyunghyun Cho, Yuandong Tian, Jason E. Weston, and Xian Li. 2025. NaturalReasoning: Reasoning in the wild with 2.8M challenging questions. In Advances in Neural Information Processing Systems, volume 38, pages 74615–74646.

Anhao Zhao, Fanghua Ye, Yingqi Fan, Junlong Tong, Jing Xiong, Zhiwei Fei, Hui Su, and Xiaoyu Shen. 2025a. SkipGPT: Each token is one of a kind. In Forty-second International Conference on Machine Learning.

Shangziqi Zhao, Jiahao Yuan, Jinyang Wu, Zhenglin Wang, Guisong Yang, and Usman Naseem. 2025b. Can pruning improve reasoning? revisiting long-CoT compression with capability in mind for better reasoning. Preprint, arXiv:2505.14582.

Taihang Zhen, Jialiang Hong, Kai Chen, Guang Yang, Junlan Feng, Wenpeng Zhu, Jing Huo, Yang Gao, Depeng Wang, Haitao Wan, Xi Yang, Fanyu Meng, Yuyao Zhang, Ji Qi, and Xiangyu Zhou. 2025. Reconsidering overthinking: Penalizing internal and external redundancy in CoT reasoning. Preprint, arXiv:2508.02178.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Jason Zhu and Hongyu Li. 2025. Towards concise and adaptive thinking in large reasoning models: A survey. Preprint, arXiv:2507.09662.

Appendix

A Prompting Details

A.1 Prompting Templates for Summarization

All our summarization prompts share a common base structure. We generate prompts by populating slots within this template:

The base structure of summary prompt

TASK: Prune the following THOUGHT PROCESS. This is a high-difficulty task requiring careful consideration and strict adherence to all constraints.

STRICT LIMIT:
`{STRICT_LIMIT_SECTION}`

COMPRESS STRATEGY:
`{COMPRESS_STRATEGY_SECTION}`

VERIFICATION: After your summary, report the exact word count.

QUESTION:
`{question}`

THOUGHT PROCESS:
`{answer}`

Format your response EXACTLY like this:
[The compressed thought process]
Words: [Number of words in your summary]

Figure 7: The base structure of the prompt used for summarization-based methods. Different sections are populated based on the desired level of structural transformation.

A.2 Component 1: Degree of Structural Transformation

The `{COMPRESS_STRATEGY_SECTION}` in our base prompt is populated with instructions corresponding to the four levels of structural transformation defined in the main paper.

Level 1: Structural Preservation (SP). The prompt instructs the model to only remove redundant tokens without altering the logical structure.

The SP Compress Strategy Section

- For this task, "pruning" means you should only remove redundant tokens.
- The goal is to preserve the original structure and sequence of every logical step.
- This strategy explicitly forbids rewriting sentences, merging steps, or deleting any part of the reasoning path.
- Conclude with the "ANSWER" section only for questions with a definitive, boxable answer. Otherwise, omit it entirely.

Level 2: Structural Editing (SE). The prompt grants the model step-level editing capabilities, including keeping, deleting, and rewriting/merging steps.

The SE Compress Strategy Section

You are an expert in logical reasoning compression. Your task is to refine and compress the provided THOUGHT PROCESS by treating it as a sequence of implicit logical steps and applying the following revision actions:
1. **Evaluate each logical component/step** within the THOUGHT PROCESS.
2. Based on your evaluation, decide how to act on it:
* **KEEP:** If a step is necessary and concise, keep it.
* **DELETE:** If a step is redundant, unnecessary, or logically flawed, remove it.
* **SINGLE-STEP COMPRESS:** If a step is necessary but verbose, rewrite it more concisely.
* **MULTI-STEP COMPRESS:** If several adjacent steps can be logically combined into a single, more efficient step, merge them.
3. Synthesize your revisions into a single, coherent, and fluent reasoning path that preserves the core logical progression to the final answer. Ensure the final answer is clearly stated at the end.
4. Conclude with the "ANSWER" section only for questions with a definitive, boxable answer. Otherwise, omit it entirely.

Level 3: Semantic Fusion (SF). The prompt encourages abstraction by instructing the model to retain only the essential "key info."

The SF Compress Strategy Section

- Retain the key info needed to solve the question and do not add additional info that is not included in the origin.
- Conclude with the "ANSWER" section only for questions with a definitive, boxable answer. Otherwise, omit it entirely.

Level 4: Framework Replacement (FR). The prompt for FR marks a conceptual shift from "compression" to "reconstruction." This is explicitly signaled to the model by altering keywords in the base template itself. Specifically, the main `TASK` description is changed to **"Re-evaluate and Reconstruct..."** and the strategy section is retitled `RECONSTRUCTION STRATEGY`. This directs the model to abandon the original CoT's structure and generate a new reasoning trace from first principles.

The FR Reconstruction Strategy

1. **PRIMARY GOAL:** Your main goal is to **REPLACE** the original 'THOUGHT PROCESS'. You are **NOT** compressing or editing the original text. You are generating a superior, structured replacement from scratch.

2. **ASSESS COMPLEXITY:** First, analyze the provided 'QUESTION' to assess its complexity (simple, moderately complex, or highly complex).

3. **CHOOSE FRAMEWORK:** Based on your assessment, choose **ONE** of the following three frameworks to generate a new thought process.

4. **FRAMEWORKS:**

* **For simple problems:** **Analysis:** [Understand the core elements and goals of the problem]
Approach: [Propose direct solution methods]
Summary: [Concisely summarize the solution approach and key points]

* **For moderately complex problems:** **Analysis:** [Understand the problem and identify key points and challenges]
Initial Approach: [Propose preliminary solutions]
Reflection: [Evaluate the pros and cons of the initial approach]
Improvement: [Refine the solution based on reflection]
Summary: [Summarize the key points of the final solution]

* **For highly complex problems:**
Analysis: [Analyze various aspects of the problem and break down its structure]
Problem Decomposition: [Break complex problem into manageable sub-problems]
Sub-problem Processing: (Repeat for each sub-problem)
- Sub-problem X: [Initial approach, Reflection, Improved solution]
Integration: [Integrate sub-problem solutions]
Overall Reflection: [Evaluate the integrated solution]
Final Optimization: [Make final optimizations]
Summary: [Summarize key points]

5. **REFERENCE USAGE:** Use the original 'THOUGHT PROCESS' ('answer') only as a reference to understand the key values, steps, and final conclusion. **DO NOT** follow its structure, wording, or verbosity. Your new output must be self-contained and strictly adhere to the selected structured framework.

A.3 Component 2: Compression Ratio Control

The `{STRICT_LIMIT_SECTION}` component controls the target length of the compressed output. We use two modes.

Fixed-Ratio Mode. Used to enforce a specific compression ratio γ. The word limit is pre-calculated based on the original CoT's length.

The FR Compress Strategy Section

STRICT LIMIT: Your summary MUST be around `{word limit}` words. This is a non-negotiable hard limit.

Self-Determined Mode. Used to let the model determine the optimal length based on the compression strategy itself, which is useful for establishing a natural compression baseline for each strategy.

The FR Compress Strategy Section

STRICT LIMIT:
Phase 1: Internal Estimation (Do this in your mind)
1. First, carefully read the entire THOUGHT PROCESS.
2. Then, mentally apply the rules from the COMPRESS STRATEGY below.
3. Estimate the final word count that would result from this pruning process. Let's call this your "Self-Determined Budget".

Phase 2: Execution
1. Now, perform the actual pruning according to the COMPRESS STRATEGY.
2. Your final output's word count MUST be close to the "Self-Determined Budget" you estimated in Phase 1.

B Condition-Aware Decision Guide

Table 4 translates our empirical findings into practical choices within selective pruning and generative rewriting. It does not assume that either method family is universally superior.

C Experimental Details

C.1 Implementation Details

Training Corpora. ($\mathcal{R}$) To study how compression strategies perform across different reasoning lengths, we create training sets from two representative datasets. For the *short-CoT* regime, we use 5,000 instances from the Natural Reasoning dataset (Yuan et al., 2025), which features compact reasoning traces. For the *long-CoT* regime, we use the LIMO dataset (Ye et al., 2025), which contains extended, multi-step rationales. This setup allows us to analyze performance in both data-abundant (short) and data-scarce (long) but highly detailed reasoning scenarios.

Fine-tuning Configuration. We fine-tune all student models using Parameter-Efficient Fine-Tuning (PEFT), specifically LoRA (Hu et al., 2022), managed by the `LLaMA-Factory` (Zheng et al., 2024) framework. Key hyperparameters, which were optimized for either short or long context regimes, are summarized in Table 5. For both LLaMA-3.1 and Qwen2.5 models, LoRA was applied

Condition	Recommended choice	Evidence and caveat
Selective pruning		
Reasoning traces can be reliably segmented into steps	Use step-level pruning; the choice of importance criterion is secondary.	Different criteria converge on a shared reasoning backbone and yield similar downstream accuracy (Section 4.3).
Fine-grained token pruning is required	Use LLMLingua-2 or explicitly protect operators, numbers, constraints, and intermediate values.	Unprotected Entropy/NLL pruning may remove symbolic anchors, especially under aggressive compression on Math tasks (Sections 4.2–4.3).
Generative rewriting		
Symbolic or math-heavy tasks	Use structural preservation (SP) with a conservative compression budget.	Math accuracy degrades monotonically as restructuring increases from SP to FR (Section 5.2).
Knowledge-intensive, general-domain tasks	Consider semantic fusion (SF), or FR with explicit validation, when stronger compression is desired; use SP when stability is the priority.	Aggressive rewriting can match or exceed SP on General tasks, but FR exhibits higher variance (Sections 5.1–5.2).
Compression budget		
Unclear rewriting budget	Choose the restructuring level first, start from the free-ratio setting, then measure and validate the realized compression ratio.	Structural constraints determine the achievable length band; the target ratio is not a precise length controller (Section 6.2).
Aggressive compression is required	Sweep and validate nearby ratios rather than extrapolating from mild compression.	Accuracy follows a plateau-then-drop pattern; Math tasks generally reach the failure region earlier (Section 6.1).
Deployment		
The student is trained with Short-CoT supervision	Use the realized training ratio as an approximate inference-cost proxy, but verify it on the target model.	Training and inference ratios are closely aligned in the Short-CoT regime (Section 6.3).
The student is trained with Long-CoT supervision	Measure inference length directly against the deployment budget.	Students may retain verbose reasoning habits; the training ratio is an optimistic lower bound on inference cost (Section 6.3).

Table 4: A condition-aware decision guide derived from Sections 4–6. The recommendations describe empirically supported defaults within the models, domains, and compression regimes evaluated in this work.

Table 5: Core fine-tuning hyperparameters for short and long-CoT regimes. The long-CoT regime uses a lower learning rate, more epochs due to a smaller dataset, and a larger effective batch size to ensure stable and robust learning.

Parameter	Short-CoT Regime	Long-CoT Regime
LoRA Configuration		
Rank (r)	16	16
Alpha (α)	32	32
Dropout	0.05	0.05
Training Dynamics		
Learning Rate	5e-5	1e-5
LR Scheduler	Cosine	Cosine
Warmup Ratio	0.1	0.05
Num. Train Epochs	3	4
Optimizer	AdamW	AdamW (via DeepSpeed ZeRO-2)
Effective Batch Size	64	64
(per-dev bs × grad accum × #gpus)	*(4 × 16 × 1)*	*(1 × 32 × 2)*

to a comprehensive set of layers, including all attention-related projections (`q_proj, k_proj, v_proj, o_proj`) and feed-forward network layers (`gate_proj, up_proj, down_proj`), to maximize adaptive capacity.

Training was performed using `bfloat16` mixed-precision. To handle the extensive sequence lengths of the long-CoT regime and improve training efficiency, we employed FlashAttention-2 (Dao, 2024) and the DeepSpeed ZeRO-2 optimizer (Rajbhandari et al., 2020). The maximum sequence length (`cutoff_len`) was set to 2048 for short-CoT data and 16,384 for long-CoT data.

C.2 Evaluation

Evaluation Benchmarks (d) To assess the acquired reasoning capabilities across diverse domains, we evaluate on five challenging bench-

marks:

- **Mathematical Reasoning**: GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), AMC23 (Mathematical Association of America, 2025)and the high-school competition-level AIME24 (Art of Problem Solving, n.d.).
- **General Reasoning**: GPQA Diamond (Rein et al., 2024) (graduate-level science) and MMLU-PRO (Wang et al., 2024) (academic exams).

Inference Settings We run all evaluations with the vLLM backend. The backend is configured as `tensor_parallel_size=1`, `dtype=bfloat16`, `max_model_len=32768`, and `gpu_memory_utilization=0.9`, with optional LoRA adapters passed through `lora_adapter`. Sampling uses `temperature=0.7`, `top_p=1.0`, `seed=42`, and `n=1`. The maximum generation length is task-specific: `8192` tokens for GSM8K, MATH500, GPQA-Diamond, MMLU-Pro, and AMC23, and `15000` tokens for AIME24. All runs are executed on a single NVIDIA A100 40GB GPU with CUDA 12.9.1 and 8 CPU cores (64GB RAM).

Evaluation Metrics. We jointly measure reasoning quality and inference efficiency. All evaluations use a standardized decoding configuration (temperature=0.7, top-p=1.0) to ensure fair comparison across all models and strategies.

- **Accuracy:** We report the percentage of problems where the model's final answer matches the ground truth for performance (Perf)
- **Generated Length:** We report the average number of tokens in the model's generated response (CoT + answer) as the inference Token Ratio (κ) for cost.

Full implementation, hyperparameter, and evaluation details are provided in the Appendix C.1.

D Method Details

D.1 Extraction-based Pruning

Extraction methods prune a CoT, C, by following a three-stage pipeline: 1) Scoring: computing an importance score $I(u)$ for each unit $u \in C$; 2) Thresholding: deriving a pruning threshold I_γ from the scores based on a target ratio γ; and 3) Pruning: removing units with $I(u) < I_\gamma$. The core of this paradigm is the choice of the scoring function, which we term the redundancy signal.

Importance criterion. Each unit u (a word or a step) receives a score $s(u)$, where larger values indicate higher importance (more worth preserving).

- **LLMLingua-2 (preserve probability):** $I(u)$ is the probability that a unit should be preserved predicted by a bidirectional scorer M_B:

$$I_{\text{lingua}}(t_i) = P(y_i = \text{preserve} \mid \mathbf{t}; \theta_{M_B}) \quad (3)$$

- **Uncertainty (Entropy):** $I(u)$ downweights uncertain tokens, using the negative average token entropy under a language model:

$$H_j = -\sum_{v \in \mathcal{V}} P(v \mid u_{<j}, \mathcal{C}) \log P(v \mid u_{<j}\mathcal{C}) \quad (4)$$

$$I_{\text{entropy}}(u) = \frac{1}{|u|} \sum_{j=1}^{|u|} H_j \quad (5)$$

where $\mathcal{C}$ denotes the conditioning context.

- **Predictability (NLL / Surprisal):** $I(u)$ favors predictable units via negative average surprisal:

$$I_{\text{NLL}}(u) = \frac{1}{|u|} \sum_{j=1}^{|u|} \Big(-\log P(u_j \mid u_{<j}, \mathcal{C})\Big) \quad (6)$$

Unit aggregation (word/step). For a unit u consisting of tokens $\{t_j\}_{j=1}^{|u|}$, we use the mean token importance:

$$I(u) \;=\; \frac{1}{|u|} \sum_{j=1}^{|u|} I(t_j), \quad (7)$$

which is applied to word units and step units respectively.

Pruning Logic. Let $\gamma \in (0, 1]$ denote the target compression ratio (kept fraction), and define the drop ratio $\rho = 1 - \gamma$. We assign each unit an importance score $I(\cdot)$, where larger values indicate higher importance. We compute a threshold I_γ using an empirical percentile with a small offset:

$$I_\gamma = \text{Percentile}\big(\{I(u_i)\}_{i=1}^{N},\ 100\rho + 1\big). \quad (8)$$

Units are retained if their importance is strictly larger than the threshold:

$$\tilde{C} = (u_i \mid I(u_i) > I_\gamma)_{i=1}^{N}. \quad (9)$$

We form $\tilde{C}$ by keeping all retained units in their original order (i.e., an order-preserving subsequence of C). For token-level compression we

first aggregate token scores into word scores (mean over tokens within each word) and apply the same percentile rule over words; For step-level compression we apply the percentile rule directly over step scores.

D.2 Summarization-based Rewriting

Summarization methods leverage a powerful LLM Gemini2.5-Flash (Google DeepMind Gemini Team, 2025) as a "rewriter" to generate a shortened trajectory $\tilde{r}$, as shown in Eq. 1. To navigate this design space, we introduce the Degree of Restructing Level, a novel dimension that categorizes rewriting strategies based on how deeply they alter the original CoT's logic. It spans four ordered levels:

- **L1: Structural Preservation (SP):** The rewriter is constrained to only remove redundant tokens, preserving the original sequence of logical steps verbatim and targeting only surface-level verbosity.
- **L2: Structural Editing (SE):** The rewriter performs step-level operations like merging or rephrasing to create a more efficient but structurally similar reasoning path.
- **L3: Semantic Fusion (SF):** The rewriter is encouraged to retain only "key information," allowing it to abstract a multi-step sequence into a single, novel statement, achieving a higher level of abstraction.
- **L4: Framework Replacement (FR):** The rewriter discards the original structure entirely and regenerates the reasoning process following a predefined formal template, enforcing a complete structural overhaul.

The specific prompts used to elicit these behaviors are detailed in Appendix A.

D.3 Compression Budget

Compression ratios. For each example i, let L_i^{teacher} denote the token length of the original (uncompressed) teacher CoT, and L_i^{comp} the token length of the compressed CoT produced by a given method. We control and measure compression through three related ratios, computed at the instance level and then averaged over the dataset.

Target ratio. We denote by $\rho_{\text{target}} \in (0, 1]$ the desired compression level specified during training or prompting. Intuitively, ρ_{target} corresponds to the intended fraction of teacher tokens to keep:

$$\frac{\sum_{i=1}^{N} L_i^{\text{comp}}}{\sum_{i=1}^{N} L_i^{\text{teacher}}} \approx \rho_{\text{target}}. \tag{10}$$

For each strategy, we train variants at three target ratios $\gamma \in \{0.9, 0.7, 0.5\}$.

Realized ratio. The realized ratio $\rho_{\text{inference}}$ is the actual compression achieved by the method:

$$\rho_{\text{realized}} = \frac{1}{N} \sum_{i=1}^{N} \frac{L_i^{\text{comp}}}{L_i^{\text{teacher}}}. \tag{11}$$

This is the quantity we use on the horizontal axis when relating compression strength to accuracy.

Inference token ratio. At evaluation time, we also track the end-to-end token cost of answering benchmark questions. For each test example i, let T_i^{meth} be the number of tokens in the CoT-plus-final-answer produced by a compressed method, and T_i^{teacher} the same quantity under the original (uncompressed) teacher-CoT setting. The inference token ratio κ is defined as

$$\rho_{\text{inference}} = \frac{\frac{1}{N}\sum_{i=1}^{N} T_i^{\text{meth}}}{\frac{1}{N}\sum_{i=1}^{N} T_i^{\text{teacher}}}. \tag{12}$$

This is the quantity of direct interest for deployment-time token budgets.

D.4 Cross-Rewriter Analysis

Beyond Gemini 2.5-Flash, we explored GPT-4o-mini as an alternative rewriter to characterize how rewriter choice influences compression behavior under the Short-CoT regime. As shown in Table 6, the two rewriters differ in absolute compression calibration, yet the structural ordering is consistent: Strong fidelity yields the most conservative compression, Light the most aggressive, and band width grows as structural constraints loosen. This suggests that length band structure is primarily governed by the degree of structural transformation rather than the rewriter itself.

E Definitions, Pareto Construction, and Full Results

This appendix details the formal metrics used to characterize the safe regime, the construction of the Pareto efficiency frontiers, and the complete numerical tabulations for all experiments.

Structure	Target γ	Gemini ρ	GPT-4o-mini ρ
Strong	0.9	0.932	0.812
Strong	0.7	0.847	0.714
Strong	0.5	0.760	0.602
Medium	0.9	0.940	0.882
Medium	0.7	0.782	0.725
Medium	0.5	0.574	0.530
Light	0.9	0.931	0.831
Light	0.7	0.755	0.682
Light	0.5	0.605	0.511

Table 6: Realized compression ratios (ρ) for Gemini 2.5-Flash and GPT-4o-mini across structural fidelity levels and target ratios (γ) under the Short-CoT regime.

E.1 Effectiveness Criterion and Safe Regime

Let A_{teacher} denote the accuracy of the uncompressed teacher model, and let $A(r)$ denote the student accuracy achieved at a realized compression ratio r. We define a compression level as *effective* if it retains at least 90% of the teacher's performance:

$$A(r) \geq 0.9 \cdot A_{\text{teacher}}. \tag{13}$$

The **safe regime** is defined as the maximal contiguous interval of realized ratios starting from $r = 1.0$ (no compression) for which this condition holds. All token counts reported (T) include both the input prompt and the generated reasoning tokens, using the same tokenizer and decoding strategy across all comparisons.

E.2 Pareto Frontier Construction

To characterize the trade-off between deployment cost and performance, we analyze the Pareto frontier in the $(\text{Cost}, \text{Performance})$ plane. Let $\mathcal{C}$ be the set of all experimental configurations (points). A point $i \in \mathcal{C}$ with inference token count T_i and accuracy A_i is considered **Pareto-optimal** if there exists no other point $j \in \mathcal{C}$ such that:

$$T_j \leq T_i \quad \text{and} \quad A_j \geq A_i,$$

with at least one inequality being strict. Practically, this frontier is constructed by sorting all valid candidates by ascending token count and retaining only those points that establish a new strictly increasing maximum in accuracy. This frontier represents the upper bound of efficiency—the maximum accuracy achievable for a given inference budget.

E.3 Full Numerical Results

We provide the complete numerical data underpinning our analysis in Table 7, Table 8 and Table 9.

- **Table 7** and **Table 8** details *extraction-based* pruning results, comparing different scoring signals (LINGUA, Entropy, NLL) across both step-level and token-level granularities for CoT Compress on LLaMA-3.1-8B-Instruct and Qwen-2.5-1.5B-Instruct.
- **Table 9** details *generative* compression results for CoT Compress on LLaMA-3.1-8B-Instruct and Qwen-2.5-1.5B-Instruct, covering the spectrum of structural fidelity (SP, SE, SF, FR) across different prompt constraints.

For each configuration, we report the top-1 accuracy (**Acc**), the average length of the compressed CoT in tokens (**Tok**), and the end-to-end inference token ratio (κ, denoted as **InferRatio**). These metrics are reported separately for **MATH** and **GENERAL** benchmarks, covering both the NATURAL REASONING (Long-CoT) and LIMO (Short-CoT) training regimes. The data points in these tables constitute the candidate set used to derive the Pareto frontiers and "knee points" discussed in the main text.

E.4 Cross-Scale Verification on Qwen2.5-7B-Instruct

To probe whether key findings hold at a larger parameter scale, we conduct a targeted verification on Qwen2.5-7B-Instruct under the Short-CoT regime, focusing on the most diagnostic conditions for each finding rather than exhaustively replicating the full sweep.

F Reflection-Aligned Step Pruning

To verify whether the compressibility of Long-CoT stems from specifically targeting reflection steps, we designed a controlled experiment to decouple *reflection deletion rate* from the underlying *scoring signal*.

F.1 Reflection-step density.

For each CoT trace t with T steps, we define the reflection-step density as:

$$\text{refl_density}(t) = \frac{\#\text{reflection steps in } t}{T} \tag{14}$$

where reflection steps are identified by a rule-based tagger over discourse markers (e.g., "wait", "hmm") and self-referential commentary (Jia et al., 2025).

Methods	Ratio	MATH						GENERAL					
		Step			Token			Step			Token		
		Acc	Tok	InferRatioκ	Acc	Tok	InferRatioκ	Acc	Tok	InferRatioκ	Acc	Tok	InferRatioκ
NATURAL REASONING													
`Original`	-	67.1	528	67.1	67.1	528	67.1	35.4	572	32.6	35.4	572	32.6
`L2`	0.9	64.6	447	84.6	65.2	466	88.2	36.9	537	93.9	36.3	441	77.1
	0.7	59.4	404	76.6	61.7	386	73.1	39.1	450	78.7	35.5	405	70.9
	0.5	49	377	71.5	50.1	279	52.9	37.7	368	64.3	32.7	300	52.5
`ENT`	0.9	64.6	518	98.1	63.7	460	87.2	35.9	609	106.5	34.9	502	87.8
	0.7	46.3	395	74.8	58.3	356	67.4	38.9	499	87.2	33.8	391	68.4
	0.5	31.2	299	566.6	46.5	257	48.7	34.3	389	68.1	30.6	372	65.1
`NLL`	0.9	63.2	523	99.1	64.0	489	92.6	36.0	628	109.8	34.2	582	101.7
	0.7	42.2	402	76.2	58.5	398	75.4	36.7	499	87.2	35.5	452	79.1
	0.5	33.3	313	59.2	45	261	49.5	34.8	398	69.6	30.3	402	70.3
LIMO													
`Original`	-	27.1	3033	-	27.1	3033	-	40.4	1454	-	40.4	1454	-
`L2`	0.9	25.6	2596	85.6	24.3	2115	69.7	41.2	1161	79.8	39.5	1069	73.5
	0.7	23.7	2582	85.1	23.8	2575	84.9	40.6	1278	87.9	38.2	1244	85.5
	0.5	28.7	2461	81.1	27.2	2173	71.6	40.9	1289	88.6	39.9	1246	85.7
`ENT`	0.9	26.6	2477	81.7	26.3	2254	74.3	38	1100	75.6	38.5	1147	78.9
	0.7	25.4	2867	94.5	24.6	1935	63.8	37.8	1126	77.4	40.5	1078	74.1
	0.5	23.4	2116	69.7	25.9	1836	60.5	39.2	1162	79.9	38.8	1196	82.3
`NLL`	0.9	24.2	2318	76.4	30.2	2065	68.1	39.8	1102	75.8	39.7	1320	90.7
	0.7	23.6	2390	78.8	22.0	1867	61.5	38.6	1131	77.8	40.5	1127	77.5
	0.5	25.0	2193	72.3	27.1	1811	59.7	41.8	1078	74.1	42.0	1201	82.6

Table 7: Experimental results of `Cot Compress` on LLaMA-3.1-8B-Instruct. We report accuracy(acc), average CoT token count (Tok), and actual compression ratio (InferRatioκ) for comparison. Accuracy (Acc) and the actual compression ratio (InferRatioκ) are both reported as percentages. (L2: LLMLingua2, ENT: Entropy)

F.2 Experimental Setup and Definitions

For a teacher trace $\mathcal{S} = \{s_i\}_{i=1}^{N}$, each step receives a strictly defined importance score q_i (via LINGUA or NLL). We identify reflection steps using a fixed, case-insensitive keyword-based regular expression with word-boundary matching (e.g., `wait`, `hmm`, `let me think`, `reconsider`, `wrong`, `validate`) (Jia et al., 2025). Formally, we define $r_i = \mathbb{1}[\mathrm{RegexMatch}(s_i)]$, where $\mathrm{RegexMatch}(\cdot)$ returns 1 iff any keyword is matched, and 0 otherwise.

Given a step-level compression target $c \in \{0.5, 0.7\}$, we define the step deletion ratio as $r = 1 - c$ and set the deletion budget as

$$K = \mathrm{round}(r \cdot N) = \mathrm{round}((1-c) \cdot N).$$

The natural tendency of a method to remove reflections is measured by the reflection deletion ratio:

$$\alpha_{\text{del}} = \frac{1}{K} \sum_{i \in \mathcal{D}} r_i, \tag{15}$$

where $\mathcal{D}$ is the set of deleted step indices (with $|\mathcal{D}| = K$). Empirically, LINGUA tends to yield a higher α_{del} (aggressive reflection removal) compared to NLL.

F.3 Constructing Reflection-Aligned Variants

We construct aligned variants to enforce a specific target ratio α_{target} while preserving the ranking logic of the base signal. We partition indices into Reflection ($\mathcal{R}$) and Other ($\mathcal{O}$) buckets.

We then allocate the deletion budget across the two buckets:

$$K_R = \mathrm{round}(\alpha_{\text{target}} \cdot K), \qquad K_O = K - K_R, \tag{16}$$

subject to feasibility (if one bucket is insufficient, the remainder is reassigned to the other bucket). Within each bucket, deletions preferentially target low-score steps under the same base scoring signal, and the final selection enforces the total budget K.

We define two anchors based on the natural behavior of the base methods at $c \in \{0.5, 0.7\}$: α_{low} (from NLL) and α_{high} (from LINGUA). This yields cross-aligned datasets:

- **NLL-HighRef**: Forces NLL to match LINGUA's aggressive reflection deletion (α_{high}).

Methods	Ratio	MATH						GENERAL					
		Step			Token			Step			Token		
		Acc	Tok	InferRatioκ	Acc	Tok	InferRatioκ	Acc	Tok	InferRatioκ	Acc	Tok	InferRatioκ
NATURAL REASONING													
Original	-	53.1	507	-	53.1	507	-	33.6	537	-	33.6	537	-
L2	0.9	50.3	442	87.3	51.2	425	83.8	32.7	505	94	30.9	447	83.1
	0.7	40.5	502	99.1	48.2	355	70.1	33.2	618	115	31.1	378	70.3
	0.5	25.3	821	162.1	43.5	279	55.1	35	593	110.3	31.4	328	61
ENT	0.9	47.2	447	88.3	48	421	83	29	500	93.1	31.6	409	76.2
	0.7	27.3	381	75.2	42.5	364	71.8	29.7	480	89.4	31.7	358	66.6
	0.5	17.7	332	65.4	36.1	300	59.1	29.7	405	75.3	35.4	266	49.4
NLL	0.9	42.5	446	88.1	47.3	433	85.5	31.7	548	102	30.8	397	73.8
	0.7	25.8	365	72	42.4	381	75.2	31.4	527	98	33.9	348	64.7
	0.5	17.3	222	43.8	37	304	59.9	30.8	403	75	36.1	280	52.1
LIMO													
Original	-	27	1368	-	27	1368	-	38.9	537	-	38.9	537	-
L2	0.9	26.9	1053	77	25.6	869	63.5	40.4	499	92.8	40.1	547	101.8
	0.7	25.9	1026	75	24	998	72.9	39.1	535	99.6	39.4	553	102.9
	0.5	23.3	1137	83.1	25.5	1151	84.1	35.7	545	101.4	37	548	102
ENT	0.9	23.7	1262	92.2	24.2	899	65.7	40.7	541	100.7	38.4	572	106.5
	0.7	25.8	1079	78.9	26.2	1023	74.8	40.3	564	104.9	40.5	585	108.8
	0.5	26	981	71.7	24	813	59.4	41.7	562	104.7	40.9	571	106.3
NLL	0.9	25.7	1137	83.1	26.2	953	69.6	38.7	553	102.9	39.7	513	95.5
	0.7	24	886	64.8	24	850	62.1	38.8	556	103.4	40.1	542	100.8
	0.5	26.1	900	65.8	24.3	1008	73.7	40.1	543	101	39.1	544	101.2

Table 8: Experimental results of `Cot Compress` on Qwen2.5-1.5B-Instruct. We report accuracy(acc), average CoT token count (Tok), and actual compression ratio (InferRatioκ) for comparison. Accuracy (Acc) and the actual compression ratio (InferRatioκ) are both reported as percentages. (L2: LLMLingua2, ENT: Entropy)

- **Lingua-LowRef**: Forces LINGUA to match NLL's conservative reflection deletion (α_{low}).

F.4 Results: Redundancy is Diffuse

We trained students on these variants and compared them to unconstrained baselines. As shown in Table 12, **the impact of reflection alignment was negligible.** At matched step budgets (K), manipulating α_{del} produced no significant deviation in downstream accuracy. Specifically:

- Forcing NLL to delete more reflections did not improve its compression efficiency.
- Restricting LINGUA to keep more reflections did not degrade its performance.

These results confirm that the "safe regime" in Long-CoT is not driven by the specific removal of reflection steps. Instead, redundancy appears to be diffuse: critical information is distributed across both reflective and non-reflective steps, and the model is robust to pruning as long as the global reasoning skeleton is maintained.

G Detailed Analysis of Token Preservation

To understand the qualitative behavior of different pruning signals, we categorize the tokens removed by each method. We implemented a Python-based classifier to map every deleted token into one of three mutually exclusive categories based on the heuristics defined below.

G.1 Token Categorization Methodology

We first tokenize text via whitespace and strip trailing punctuation. Each token is then classified sequentially according to the following priority rules, strictly aligning with our evaluation script:

1. **Class 1: Constraint-Bearing & Symbolic (High Priority).** This class captures the structural anchors of the reasoning chain. It includes:
   - **Logic Markers:** Key deductive connectives (e.g., *"let", "assume", "implies", "therefore", "since", "if", "iff"*).
   - **Math Operators & Symbols:** All standard arithmetic operators, comparators,

Methods	Ratio	LLaMA-3.1-8B-Instruct						Qwen2.5-7B-Instruct					
		MATH			GENERAL			MATH			GENERAL		
		Acc	Tok	InferRatioκ	Acc	Tok	InferRatioκ	Acc	Tok	InferRatioκ	Acc	Tok	InferRatioκ
`Original`	-	67.1	528	61.7	35.4	572	32.6	53.1	507	–	33.6	537	–
NATURAL REASONING													
`SP`	0.9	66.9	522	98.9	39.8	582.5	101.9	50.6	464	91.6	397	533.5	99.3
	0.7	66.2	564	106.8	37.7	547	95.7	48.9	432	85.2	40.4	208	38.7
	0.5	68.1	442	83.7	38.2	465	81.4	49.8	411	81	41	389.5	72.5
	0.35	66.8	482	91.4	34.7	541	94.7	50.8	457	90.2	30.3	496.5	92.5
`SE`	0.9	59	461	87.3	42.2	537.5	94.1	48.1	319	63	45.4	406.5	75.7
	0.7	61.5	358	67.9	41.6	421.5	73.8	47.9	262	51.6	49.2	339.5	63.2
	0.5	58.8	327	62	40.6	341	59.7	48.8	230	45.3	487	202.5	37.7
	0.35	58.1	265	50.1	32.4	241	42.2	46.9	174	34.3	45	134.5	25
`SF`	0.9	61.8	422	0.8	43	532	93.1	49.9	370	73.1	47.4	530	98.7
	0.7	60.7	438	82.9	24.8	412.5	72.2	48.9	294	58	42.7	211.5	39.4
	0.5	58	413	78.3	22.9	391.5	68.5	49.3	237	46.8	43.3	143.5	26.7
	0.35	59.3	193	36.6	34.2	179	31.3	44.4	157	30.9	49	86	16
`FR`	0.5	49.4	341	64.6	43.4	421	73.7	27.5	270	53.2	46.4	336.5	62.7
LIMO													
`SP`	0.9	24	2333	76.9	41.8	1343	92.4	23.7	1066	77.9	38.6	577	107.4
	0.7	21	2629	86.7	37.2	1248	85.8	27.4	1061	77.5	38.7	552	102.8
	0.5	20.9	3439	113.4	39.9	1290	88.7	21.9	1118	81.7	38.8	599	111.5
	0.35	27	3069	101.2	40.8	1308	89.9	23.1	1333	97.4	39.8	583	108.6
`SE`	0.9	24.6	3153	103.9	37.9	1487	102.3	21.3	1494	109.2	39.1	595	110.8
	0.7	22.3	3287	108.1	41.9	1317	90.5	24.9	1210	88.5	39.4	608	113.2
	0.5	23.9	3690	121.6	38.2	1543	106.1	23	1303	95.2	39.6	582	108.4
	0.35	22.6	3068	101.1	35.9	896	61.6	19.3	1118	81.7	43.3	480	89.4
`SF`	0.9	24.1	3126	104.2	39.7	1528	105.1	25.3	1197	87.5	41.7	610	113.6
	0.7	21.8	2474	81.6	43.2	1456	100.1	24	1199	87.6	39.3	640	119.1
	0.5	21.2	3987	131.4	37.4	1618	111.3	24	1322	96.6	41.3	569	105.9
	0.35	26.3	1844	60.8	36.2	741	51	18.1	96.8	70.8	40.7	434	80.8
`FR`	0.5	23.2	2375	78.3	38.8	1364	93.8	23.3	1051	76.8	43.9	566	105.4

Table 9: Experimental results of `Cot Compress` on LLaMA-3.1-8B-Instruct and Qwen2.5-1.5B-Instruct. We report accuracy (Acc), average CoT token count (Tok), and actual compression ratio (InferRatioκ) for comparison. Accuracy (Acc) and the actual compression ratio (InferRatioκ) are both reported as percentages.

Granularity	Signal	Ratio	MATH		GENERAL	
			Acc	ρ	Acc	ρ
—	Baseline	—	75.3	1.00	40.1	1.00
Token	Entropy	0.7	54.6	0.95	36.8	0.99
	Entropy	0.5	46.7	0.97	37.5	0.86
	NLL	0.7	15.8	1.37	33.5	1.09
	LLMLingua-2	0.7	74.1	1.21	39.6	1.14
Step	Entropy	0.7	74.8	1.03	40.8	1.06
	Entropy	0.5	73.3	0.87	40.8	0.91

Table 10: Targeted extraction-based verification on Qwen2.5-7B-Instruct (Short-CoT). Token-level Entropy and NLL collapse on Math while step-level methods remain stable, consistent with findings on LLaMA-3.1-8B and Qwen2.5-1.5B.

Structure	Ratio	MATH		GENERAL	
		Acc	ρ	Acc	ρ
Baseline	—	75.3	1.00	40.1	1.00
SP	0.5	76.8	0.68	39.5	0.81
SE	0.5	76.2	0.69	40.2	0.66
SF	0.5	71.7	0.51	40.1	0.62

Table 11: Targeted summarization-based verification on Qwen2.5-7B-Instruct (Short-CoT). The fidelity wall reproduces: Math accuracy degrades from SP to SF while General performance remains stable, consistent with findings across all model families.

and Unicode mathematical symbols (e.g., $\in, \subseteq, \forall, \sum$).

- **LaTeX Commands:** Substrings indicative of formal notation (e.g., `\frac`, `\boxed`, `\sqrt`).
- **Numerical Patterns:** Any token containing digits, including scientific notation and

Table 12: Results of reflection-aligned extraction. **Lingua-LowRef** constrains Lingua to match NLL's lower reflection deletion rate, while **NLL-HighRef** forces NLL to match Lingua's higher deletion rate. Accuracy remains stable despite significant shifts in reflection retention.

Ratio (c)	Method	Math		General	
		Acc (%)	Tok	Acc (%)	Tok
0.5	LINGUA-BASE	27.2	2173	39.9	1246
	LINGUA-LOWREF	27.0	2315	38.1	1176
	NLL-BASE	27.1	2319	42.0	1201
	NLL-HIGHREF	27.7	2555	40.5	1210
0.7	LINGUA-BASE	23.8	2575	38.2	1244
	LINGUA-LOWREF	25.4	2082	35.9	1121
	NLL-BASE	22.0	2391	40.5	1127
	NLL-HIGHREF	23.3	2474	38.8	1208

	0.1			0.3			0.5		
Signal	Sym	Stop	Other	Sym	Stop	Other	Sym	Stop	Other
Entropy	0.10	0.11	0.08	0.30	0.29	0.21	0.49	0.44	0.36
NLL	0.11	0.12	0.07	0.30	0.28	0.21	0.51	0.40	0.34
LLMLingua	0.03	0.29	0.02	0.15	0.54	0.11	0.32	0.68	0.25

Table 13: Extended breakdown of deleted token types on NATURALREASONING for token-level pruning across target compression ratios.

fractions.

2. **Class 2: Stopwords.** Tokens found in standard English stopword lists (sourced from NLTK (Bird and Loper, 2004)/scikit-learn). **Crucially**, to prevent misclassification, any stopword that also appears in the *Logic Markers* list (e.g., "if", "not") is removed from this set and forced into Class 1. Thus, Class 2 strictly represents low-semantic syntactic glue.
3. **Class 3: Content.** All remaining tokens that do not fall into Class 1 or Class 2. These are typically domain-specific nouns, verbs, and adjectives describing the problem context.

G.2 Results: Deletion Distribution

Table 13 presents the **Deletion Rate** for each category, defined as the fraction of tokens of that type removed from the original trace.

The data reveals a clear behavioral signature: LLMLINGUA-2 acts as a "logic-aware" filter. Even at aggressive compression levels, it consistently yields significantly lower deletion rates for **Sym** (Class 1) tokens compared to Entropy and NLL baselines, while shifting the deletion budget heavily towards **Stop** (Class 2) tokens. This confirms that semantic probability estimators are better aligned with the logical structure of chain-of-thought reasoning than simple perplexity-based metrics.

G.3 Validation of the Symbolic Anchor Hypothesis

To further verify that loss of symbolic structure in NLL-based degradation, we conducted a controlled experiment with a restricted pruning scope. In this setting, we excluded tokens categorized as math-like patterns from the pruning candidate set entirely. Consequently, the NLL-based ranking and deletion were applied solely to the remaining vocabulary (i.e., stopwords and other content tokens), while all math-related tokens were preserved by default.

As shown in Table 14, this selective retention strategy significantly improved the stability of NLL-based pruning. By simply preventing the removal of these symbolic anchors, the optimized NLL baseline recovered a large portion of the performance gap relative to LLMLingua2. This result confirms that the standard NLL metric fails in high-risk settings specifically because it assigns low probabilities to essential symbolic operators, not because the metric itself is incapable of ranking redundant prose.

H Step-Level Decision Overlap Analysis

To empirically verify the claim that different pruning signals converge on a shared reasoning backbone while disagreeing on peripheral redundancy, we analyzed the pairwise consensus of their pruning decisions.

	Math		General	
Ratio	NLL	+Protect	NLL	+Protect
0.9	0.632	**0.675**	0.360	**0.364**
0.7	0.422	**0.632**	0.367	**0.376**
0.5	0.333	**0.608**	0.348	**0.404**

Table 14: Ablation study on token protection during NLL pruning. Results reported on Llama 3.1 8B Instruct (Short CoT). "+Protect" denotes NLL pruning where math-related tokens are excluded from the deletion candidates.

Pair	0.1	0.3	0.5
ENT–L2	0.09	0.16	0.31
NLL–L2	0.10	0.17	0.33
ENT–NLL	0.70	0.70	0.76

Table 15: Pairwise Jaccard overlap between **deleted** step sets for different step-level scoring signals under matched target ratios. (ENT: Entropy; L2: LLMLingua2).

H.1 Calculation Methodology

We implemented a strict index-based comparison pipeline. For any pair of pruning methods $\mathcal{M}_A$ and $\mathcal{M}_B$ (e.g., Entropy vs. LLMLingua-2):

1. **Alignment:** We align samples by question ID across the generated '.jsonl' outputs.
2. **Index Mapping:** For each sample, we map the retained text segments back to the original teacher step indices $\{1, \ldots, N\}$.
3. **Set Construction:** We derive the set of *deleted* indices for each method, denoted as $\mathcal{D}_A$ and $\mathcal{D}_B$.
4. **Metric:** We compute the Jaccard Similarity Coefficient for the deletion sets:

$$J(\mathcal{D}_A, \mathcal{D}_B) = \frac{|\mathcal{D}_A \cap \mathcal{D}_B|}{|\mathcal{D}_A \cup \mathcal{D}_B|} \tag{17}$$

This process is repeated for each target ratio and averaged across the dataset.

H.2 Divergence in Deletion Choices

Table 15 presents the overlap of **deleted steps** across methods. We observe a clear dichotomy:

- **High Internal Consistency:** Entropy (ENT) and NLL show high overlap (> 0.70), indicating that probability-based metrics largely identify the same "easy" tokens to remove.
- **Low Cross-Paradigm Overlap:** When comparing probability-based methods against the symbol-aware LLMLingua-2 (L2), the overlap drops significantly (0.10–0.33).

Despite this substantial disagreement on *what to delete*, the main text (Section 4) shows that their downstream accuracy is nearly identical. This confirms that the non-overlapping deleted steps represent **interchangeable redundancy**—peripheral information whose removal (whether by probability or semantic estimation) does not disrupt the core reasoning chain.

I LLM-Based Coherence Judge

While quantitative metrics measure task accuracy, they do not fully capture the linguistic quality of the compressed traces. Our analysis suggests that token-level pruning—unlike step-level selection—can severely disrupt local continuity and readability. While other method (Wang et al., 2025b) measures a token-level LM loss as a proxy of inter-chunk coherence, we are interested in how coherent a compressed CoT is as a teaching script for smaller models, so we implement an LLM-based coherence judge to perform pairwise comparisons.

Task Definition. We evaluate the *reasoning coherence* of two Chain-of-Thought (CoT) rationales (Method A vs. Method B) generated for the same question. The judge is explicitly instructed to ignore the correctness of the final answer and focus solely on linguistic and logical flow. The output is a decision label in $\{\texttt{A}, \texttt{B}, \texttt{TIE}\}$.

Judge Implementation. We utilize a powerful LLM, `gemini-2.5-flash` (Google DeepMind Gemini Team, 2025), as the judge. The prompt instructs the model to evaluate: (i) global structural organization, (ii) logical continuity between steps, (iii) proper usage of intermediate conclusions, and (iv) the absence of disjointed or "broken" text.

Table 16: **Coherence Evaluation Results.** Pairwise comparison of semantic compression (LLMLINGUA-2) versus naive ratio pruning over 100 samples per setting. "Math-Prot" denotes the variant with protection for mathematical symbols.

Granularity	Configuration	Target ρ	A Wins	B Wins	Win Rate (A)
Step	LLMLingua-2	0.7	95	3	0.960
Step	LLMLingua-2	0.9	92	5	0.935
Token	LLMLingua-2	0.7	100	0	1.000
Token	LLMLingua-2	0.9	100	0	1.000
Token	LLMLingua-2 (Math-Prot)	0.7	100	0	1.000

The pairwise comparison prompt used for the coherence judge.

You are evaluating the quality of two chain-of-thought (CoT) rationales for the **same question**. Your goal is to decide which CoT is more coherent and easier to follow **as a reasoning process**, ignoring whether the final answer is correct.
Do **not** consider whether the final numerical/choice answer is correct. You should focus only on how clear and coherent the reasoning itself is.
[Question] question
[CoT A]

$$cot_a$$

[CoT B]

$$cot_b$$

Please compare CoT A and CoT B only in terms of **reasoning coherence**, according to the following criteria:
1. Global organization: the rationale states the goal/-known facts clearly and follows a sensible overall plan. 2. Step-to-step logic: each step follows logically from previous steps, without unexplained jumps. 3. Use of intermediate conclusions: important intermediate results are clearly derived and then used later. 4. Noise and redundancy: the rationale avoids unnecessary digressions, off-topic commentary, or excessive repetition that makes it harder to follow.
If one CoT is much shorter but clearly skips important steps, treat it as **less** coherent than a slightly longer but well-explained CoT. Ignore minor formatting issues such as non-consecutive step numbers. Focus on whether the reasoning itself is logically coherent and easy to follow.
If one CoT is more coherent, choose that one. If their coherence is very similar, choose TIE.
First, briefly explain your comparison in 2–3 sentences. Then, on a separate line, output your decision in the format:
DECISION: A or DECISION: B or DECISION: TIE

Aggregation. For each experimental configuration, we evaluate 100 pairs. The win rate for Method A is calculated as:

$$\text{WinRate(A)} = \frac{\#(\texttt{A}) + 0.5 \cdot \#(\texttt{TIE})}{\#(\texttt{A}) + \#(\texttt{B}) + \#(\texttt{TIE})}.$$

Results: Extraction Table 16 summarizes the pairwise coherence results comparing semantic compression (Method A: LLMLINGUA-2) against a standard length-based baseline (Method B: RATIO). The results overwhelmingly favor semantic compression, particularly at the token level (100% win rate), confirming that naive token deletion catastrophic destroys coherence, whereas semantic estimators preserve readability.

Results: Summarization Table 17 extends the evaluation to summarization-based methods, comparing each structural fidelity level against the extraction baseline across compression ratios.

Structure	$\gamma = 0.5$	$\gamma = 0.7$	$\gamma = 0.9$	Free
None	0.41	—	—	—
Light	0.52	0.53	0.73	0.21
Medium	0.41	0.58	0.79	0.30
Strong	0.12	0.24	0.34	0.38

Table 17: Coherence win rates of summarization methods against an extraction baseline (win rate > 0.5 favors summarization).

Structure Light and Medium show increasing coherence win rates as compression tightens, reaching 0.73 and 0.79 at $\gamma = 0.9$. Structure Strong consistently scores below 0.5 across all settings, reflecting surface-level fragmentation from strict token-removal constraints—a pattern consistent with its strong Math accuracy in the main experiments. Structure None yields the lowest win rate overall, aligning with its higher output variance and degraded Math performance observed in Section 5.

J Quantitative verification of generative rewriting

J.1 Quantitative verification of restructuring order

Sample construction. To verify that the four restructuring levels are ordered by depth in an objective manner, we construct a frozen, balanced sample of 800 rewritten traces. The sample includes both supervision regimes: LIMO for Long-CoT and NaturalReasoning for Short-CoT, with 100 traces per restructuring level per dataset. Sampling is performed with a single fixed seed, and compressed traces are aligned to their original uncompressed traces before evaluation.

Metric. We compute the metric copy_rate, defined as the fraction of the rewritten trace copied verbatim from the original in spans of at least five tokens. Before matching, both traces are normalized by lowercasing, removing leading fillers, and collapsing whitespace and punctuation spacing. Verbatim spans are extracted using a SequenceMatcher procedure. The metric is formally defined as

$$\text{copy_rate} = \frac{\text{matched verbatim tokens}}{\text{total tokens in rewritten trace}}.$$

Higher values indicate that the rewritten trace remains closer to the original wording, whereas lower values indicate deeper restructuring.

Results. The measured copy_rate decreases monotonically from L1 to L4 in both datasets. For LIMO, the median copy_rate values are approximately 0.664, 0.187, 0.096, and 0.035 for L1–L4, respectively; for NaturalReasoning, the corresponding values are approximately 0.635, 0.234, 0.120, and 0.041. We also compute the Spearman rank correlation between restructuring level and copy_rate; the correlation is negative and highly significant in both datasets: for LIMO, $\rho = -0.6757$ with $p = 1.14 \times 10^{-54}$; for NaturalReasoning, $\rho = -0.6454$ with $p = 1.70 \times 10^{-48}$. These results show that the intended ordering of restructuring depth is recovered objectively and is not merely assumed from prompt design or model-specific behavior.

Restructuring Level	LIMO Median [Q1, Q3]	NaturalReasoning Median [Q1, Q3]
L1 SP	0.664 [0.364, 0.910]	0.635 [0.419, 0.866]
L2 SE	0.187 [0.116, 0.190]	0.234 [0.205, 0.325]
L3 SF	0.096 [0.042, 0.133]	0.120 [0.115, 0.140]
L4 FR	0.035 [0.022, 0.049]	0.041 [0.018, 0.067]
Spearman ρ	-0.68, $p < 1\text{e}{-54}$	-0.65, $p < 1\text{e}{-48}$

Table 18: Copy-rate summary by dataset and restructuring level.

Restructuring level	LIMO	NaturalReasoning
L1 Structural Preservation	0.94	0.99
L2 Structural Editing	0.91	0.91
L3 Semantic Fusion	0.87	0.98
L4 Framework Replacement	0.86	0.93
Overall	0.895	0.952

Table 19: Answer-preservation rates by dataset and restructuring level.

J.2 Quantitative verification of answer preservation

Evaluation setup. We further verify that increasing restructuring level does not systematically alter the final answer. To this end, answer preservation was evaluated using a blind judge with DeepSeek-V4-Flash (DeepSeek-AI, 2026) at temperature 0. The judge is asked whether the rewritten trace reaches the same final answer as the original trace, and the restructuring level is not provided to the judge. The judge output is a structured response with values `true`, `false`, or `NA`, and only comparable items are included in the denominator.

Metric. The answer preservation rate measures the fraction of comparable rewritten traces that produce the same final answer as the original trace.

Results. Across all levels, answer preservation remains high in both datasets. For LIMO, the preservation rates are 94%, 91%, 87%, and 86% for L1–L4, respectively; for NaturalReasoning, the rates are 99%, 91%, 98%, and 93%. These results indicate that stronger rewriting does not systematically break the final answer, even when the rewritten trace is substantially more compressed or structurally transformed.

K CoT Compression Case Study

question	**Calculate the energy needed for a 1 kg mass to accelerate from 2 m/s to 6 m/s, considering the reference frame for kinetic energy and speed. Provide a detailed explanation of your calculation, including how you account for the change in kinetic energy and the relevance of the reference frame.**
Extraction Method	**Generated Output**
Original	## Step 1: Identify the formula for kinetic energy The kinetic energy (KE) of an object is given by the formula $KE = 0.5 * m * v^2$, where m is the mass of the object and v is its velocity. ## Step 2: Calculate the initial kinetic energy To find the initial kinetic energy, we use the given initial velocity ($v_1 = 2$ m/s) and the mass ($m = 1$ kg) in the kinetic energy formula: $KE_1 = 0.5 * 1 * (2)^2 = 0.5 * 1 * 4 = 2$ Joules. ## Step 3: Calculate the final kinetic energy Using the given final velocity ($v_2 = 6$ m/s) and the same mass ($m = 1$ kg) in the kinetic energy formula: $KE_2 = 0.5 * 1 * (6)^2 = 0.5 * 1 * 36 = 18$ Joules. ## Step 4: Determine the change in kinetic energy The change in kinetic energy (ΔKE) is the difference between the final and initial kinetic energies: $\Delta KE = KE_2 - KE_1 = 18 - 2 = 16$ Joules. ## Step 5: Consider the relevance of the reference frame The calculation of kinetic energy and the change in kinetic energy depends on the reference frame, as kinetic energy is a frame-dependent quantity. However, since we are calculating the change in kinetic energy for an object accelerating from one speed to another within the same reference frame (implied by the problem statement), the reference frame does not affect the calculation of the energy needed for the acceleration in this context. The energy needed is the same regardless of the observer's frame of reference because the initial and final velocities are given relative to the same frame. The final answer is: $\boxed{16}$
Word level Entropy at 50% compression ratio	## Step Identify formula kinetic The kinetic (KE) of is $KE0.5 * m$ and velocity. ## Calculate initial kinetic To find initial energy, we use given initial velocity (v1 = m/s) and the mass (m kg) kinetic formula: $KE11 * (2)^2 = 0.5$ Joules. ## Calculate Using given the same (m kg) formula: Determine change The (ΔKE) is the difference final and energies: - Consider the relevance the The calculation of kinetic and the change in kinetic depends on the reference frame, as kinetic is a frame-dependent quantity. However, since we are calculating the change kinetic for an object accelerating from one speed within the reference (implied by the problem statement), the reference frame does not affect the calculation of the energy needed for the acceleration in this context. The energy needed is the same regardless of the observer's frame of reference because the initial and final velocities are given relative to the same frame. The final answer is: $\boxed{16}$
Word level LLMLingua2 at 50% compression ratio	Step Identify formula kinetic energy formula $KE = 0.5 * m * v^2$ m mass v velocity Calculate initial kinetic energy velocity ($v_1 = 2$ m/s) mass ($m = 1$ kg) $KE_1 = 0.5*1*(2)^2 = 0.5*1*4 = 2$ Joules Calculate final kinetic energy velocity ($v_2 = 6$ m/s) mass ($m = 1$ kg) $KE_2 = 0.5*1*(6)^2 = 0.5*1*36 = 18$ Joules Determine change in kinetic energy difference final initial kinetic energies $\Delta KE = KE_2 - KE_1 = 18 - 2 = 16$ Joules Consider relevance reference frame calculation kinetic energy depends reference frame accelerating another within the same reference frame (implied by the problem statement), the reference frame does not affect the calculation of the energy needed for the acceleration in this context. The energy needed is the same regardless of the observer's frame of reference because the initial and final velocities are given relative to the same frame. The final answer is: $\boxed{16}$
Word-level NLL at 50% compression ratio	## Step Identify formula kinetic The kinetic (KE) of is given $KE0.5 * m$ of and its velocity. ## Calculate initial kinetic To find initial energy, we use given initial velocity ($v_1 =$ m/s) and the mass (m kg) kinetic formula: $KE_1 1 * (2)^2 = 0.5$ Joules. ## Calculate Using given same kg) formula: Determine change The (ΔKE) is the difference and initial energies: - Joules. ## Consider the relevance The calculation of kinetic and the change kinetic depends on reference frame, as kinetic is a frame-dependent quantity. However, since we are calculating the change for an object accelerating from one speed within the same reference (implied by the problem statement), the reference does affect the calculation of the energy needed for the acceleration in this context. The energy needed is the same regardless of the observer's frame of reference because the initial and final velocities are given relative to the same frame. The final answer is: $\boxed{16}$

Table 20: Case study: reflection-aligned extraction at 50% compression. For the same kinetic-energy prompt, we report the original uncompressed response and extracted outputs produced using token-level redundancy signals (Entropy, Lingua, and NLL).

question	**Calculate the energy needed for a 1 kg mass to accelerate from 2 m/s to 6 m/s, considering the reference frame for kinetic energy and speed. Provide a detailed explanation of your calculation, including how you account for the change in kinetic energy and the relevance of the reference frame.**
Generative Rewriting	**Generated Output**
L4:**Framework Replacement** at 50% compression ratio	**Analysis:** Determine energy for 1 kg accelerating 2 m/s to 6 m/s (ΔKE). Critically, address reference frame impact on KE and required energy. **Initial Approach:** Compute initial KE ($0.5 * 1 * 2^2$) and final KE ($0.5 * 1 * 6^2$). Energy needed is ΔKE. KE depends on frame. **Reflection:** Though KE is frame-dependent, *work done* (ΔKE) is invariant across inertial frames if velocities are consistently measured from one point. **Improvement:** $KE_1 = 2$ J; $KE_2 = 18$ J. So, $\Delta KE = 16$ J. This energy change remains consistent for all inertial observers for this specific acceleration. **Summary:** Energy needed = $\Delta KE = 16$ J (18 J final - 2 J initial). This change is invariant across inertial frames, as velocities were consistently defined. ANSWER: $\boxed{16}$
L3:**Semantic Fusion** at 50% compression ratio	The energy needed is the change in kinetic energy. First, calculate the initial kinetic energy (KE_1) using $KE = 0.5 * m * v^2$: $KE_1 = 0.5 * 1\,\text{kg} * (2\,\text{m/s})^2 = 2$ Joules. Next, calculate the final kinetic energy (KE_2): $KE_2 = 0.5 * 1\,\text{kg} * (6\,\text{m/s})^2 = 18$ Joules. The energy required is $\Delta KE = KE_2 - KE_1 = 18\,\text{J} - 2\,\text{J} = 16$ Joules. Although kinetic energy is frame-dependent, the change in kinetic energy for acceleration from 2 m/s to 6 m/s is calculated within the same reference frame, making the energy needed (16 Joules) a consistent and relevant value for this specific scenario.
L2:**Structural Editing** at 50% compression ratio	The energy required for acceleration is equivalent to the change in the object's kinetic energy (ΔKE). Kinetic energy (KE) is calculated using the formula $KE = 0.5 * m * v^2$. For the 1 kg mass: 1. Initial Kinetic Energy: At 2 m/s, $KE_1 = 0.5 * 1\,\text{kg} * (2\,\text{m/s})^2 = 2$ J. 2. Final Kinetic Energy: At 6 m/s, $KE_2 = 0.5 * 1\,\text{kg} * (6\,\text{m/s})^2 = 18$ J. The energy needed is the difference: $\Delta KE = KE_2 - KE_1 = 18\,\text{J} - 2\,\text{J} = 16$ J. While kinetic energy itself is frame-dependent, the *change* in kinetic energy and the associated energy input are frame-independent when the initial and final velocities are measured within the same reference frame, as specified in this problem. Therefore, the calculated 16 J accurately reflects the energy needed regardless of the chosen inertial reference frame. ANSWER: $\boxed{16}$
L1:**Structural Preservation** at 50% compression ratio	## Step 1: Identify the formula for kinetic energy Kinetic energy (KE) formula: $KE = 0.5 * m * v^2$; m is mass, v is velocity. ## Step 2: Calculate the initial kinetic energy Initial KE: Use $v_1 = 2$ m/s, $m = 1$ kg in KE formula: $KE_1 = 0.5 * 1 * (2)^2 = 0.5 * 1 * 4 = 2$ Joules. ## Step 3: Calculate the final kinetic energy Final KE: Use $v_2 = 6$ m/s, $m = 1$ kg in KE formula: $KE_2 = 0.5 * 1 * (6)^2 = 0.5 * 1 * 36 = 18$ Joules. ## Step 4: Determine the change in kinetic energy Change in KE (ΔKE), difference between final, initial KEs: $\Delta KE = KE_2 - KE_1 = 18 - 2 = 16$ Joules. ## Step 5: Consider the relevance of the reference frame Calculation of KE, ΔKE depends on reference frame; KE is frame-dependent. However, calculating ΔKE for an object accelerating within same reference frame (implied), frame does not affect energy needed. Energy needed is same regardless of observer's frame because initial, final velocities are relative to same frame. ANSWER: $\boxed{16}$

Table 21: Case study: generative rewriting at 50% compression. For the same prompt, we report summaries generated without a redundancy signal and under different structure constraints (Structure Light/Medium/Strong), illustrating the effect of format control.

L Inference-Side Failure Cases

question	**If $f(x) = \frac{3x-2}{x-2}$, find $f(-2) + f(-1) + f(0)$.**
Method	**Generated Output**
Structural Preservation (✓)	$\cdots$ Step 4: $f(-2) = \frac{-8}{-4} = 2$. Step 5: $f(-1) = \frac{-5}{-3} = \frac{5}{3}$. Step 6: $f(0) = \frac{-2}{-2} = 1$. $\cdots$ Step 9: $\frac{9}{3} + \frac{5}{3} = \frac{14}{3}$. **ANSWER:** $\boxed{\frac{14}{3}}$
Framework Replacement (✗)	**Analysis:** The problem requires calculating $f(-2) + f(-1) + f(0)$ for $f(x) = \frac{3x-2}{x-2}$. The expression is a rational function, meaning it is undefined at $x = 2$. This means $f(-2)$, $f(-1)$, $f(0)$ are undefined. Since the sum of undefined values is also undefined, the final answer is undefined. **Summary:** The function $f(x)$ is undefined for $x = 2$. Their sum is also undefined. **Final Answer:** $\boxed{\text{undefined}}$

Table 22: Case 1: The Structure None student bypasses actual substitution by latching onto a superficially relevant concept—the undefined point of the function—a direct consequence of losing step-level computational scaffolding during training.

question	**The solution to $-4 < 2(x-1) < 8$ is expressed in the form $a < x < b$. Find the value of $a + b$.**
Method	**Generated Output**
LLMLingua-2 word-level pruning (✓)	Step 1: Solve inequality $-4 < 2(x-1) < 8 \cdots -2 < x - 1 < 4$, add 1 to all sides inequality $-2 + 1 < x - 1 + 1 < 4 + 1$. Step 2: Simplify inequalities simplify $-1 < x < 5$. Step 3: Determine values a and b: $a = -1$, $b = 5$ from inequality $-1 < x < 5$. Step 4: Calculate $a + b$: $a + b = -1 + 5 = 4$. **The final answer is:** $\boxed{4}$
NLL word-level pruning (✗)	Step 1: Solve for x in the inequality $-4 < 2(x1) < 8$. $\cdots$ First, we have $-4 < 2(x1)$, then we have $2(x1) < 8$. $\cdots$ Next, solve the second part of the inequality $2(x1) < 8$. Dividing both sides by 2, we get $x < 8$. Combine the solutions of the two inequalities. The combined solution is $-1 < x < 8$. $\cdots$ Calculate $a + b$. $-1+ = 7$. **The final answer is:** $\boxed{7}$

Table 23: Case 2: Token-Level Signal Sensitivity. As shown above, NLL-based pruning removes constraint-bearing tokens at substantially higher rates than LLMLingua-2. This propagates directly to inference: the NLL student drops the critical key symbols in $2(x-1) < 8$, leading to final calculation errors, while the LLMLingua-2 student preserves all key numerical steps despite similarly fragmented prose.