

Revealing Procedural Reasoning Structures in Chain-of-Thought Training via Span-Level Gradient Organization

Anonymous ACL submission

Abstract

Chain-of-Thought (CoT) prompting enables large language models to produce multi-step reasoning, yet how such reasoning-related structure is expressed during training remains poorly understood. We present Gradient-based Structural Developer (GSD), an unsupervised framework with a principled gradient aggregation view that tracks span-level gradient during fine-tuning on reasoning benchmarks to understand how models develop structured, step-by-step reasoning capabilities. Our analysis shows that while gradients at the level of individual tokens are often noisy, aggregating gradients over contiguous reasoning-related spans reveals stable and recurring directional alignment across samples. We refer to these directionally aligned patterns as aligned sequential stresses, reflecting consistent gradient organization associated with similar reasoning procedures. Beyond capturing semantically similar reasoning instances, such gradient alignment also reveals structurally similar but semantically diverse cases that share common procedural organization. These findings position GSD as an explainability framework. It makes the internal formation of procedural reasoning during training transparent, enabling human-understandable analysis and diagnosis of how reasoning-oriented behaviors emerge in language models.

1 Introduction

Large language models (LLMs) exhibit strong reasoning behavior when guided by structured intermediate steps, as exemplified by Chain-of-Thought (CoT) prompting (Zhang et al., 2024). While CoT-style prompts reliably elicit multi-step reasoning in model outputs, little is known about how reasoning-related structure is reflected during training (Lee et al., 2025; Lanham et al., 2023; Ren et al., 2022). Most existing studies focus on generated reasoning traces, prompt effectiveness, or downstream performance (Lee et al., 2025; Lanham et al., 2023;

Ren et al., 2022), leaving open the question of how shared reasoning procedures are reflected in training-time dynamics (Zhang and Parkes, 2023; Ren et al., 2024; Wang et al., 2024b; Ranaldi et al., 2025; Guo et al., 2024).

Prior work on implicit and latent reasoning further suggests that similar reasoning procedures do not necessarily correspond to a single explicit or semantically homogeneous surface form (Deng et al., 2023; Li et al., 2025; Zelikman et al., 2024). Different problem instances may be expressed with distinct lexical content while following comparable procedural organization, such as identifying relevant information before performing stepwise transformation or computation. Such a gap between surface-level semantic similarity and underlying reasoning structure motivates approaches that probe training-time organization beyond forward-pass representations.

In this work, rather than inferring reasoning patterns from model outputs, we directly analyze gradients produced during fine-tuning. This allows us to examine how training-time signals associated with reasoning procedures are reflected in parameter-space updates. Based on this perspective, we introduce **Gradient-based Structural Developer (GSD)**, a framework that analyzes the relationship between training data gradient organization and the emergence of structured, step-by-step reasoning in models. GSD extracts gradients from training examples with consistent reasoning patterns and aggregates them over short contiguous token spans. We refer to the resulting consistency in gradient direction across samples as span-level gradient alignment. This design is motivated by the observation that individual token gradients are often noisy, while gradients of short spans reveal strong and consistent directional alignment across samples. GSD does not rely on output supervision, predefined reasoning templates, or semantic annotations, enabling fully unsupervised analysis.

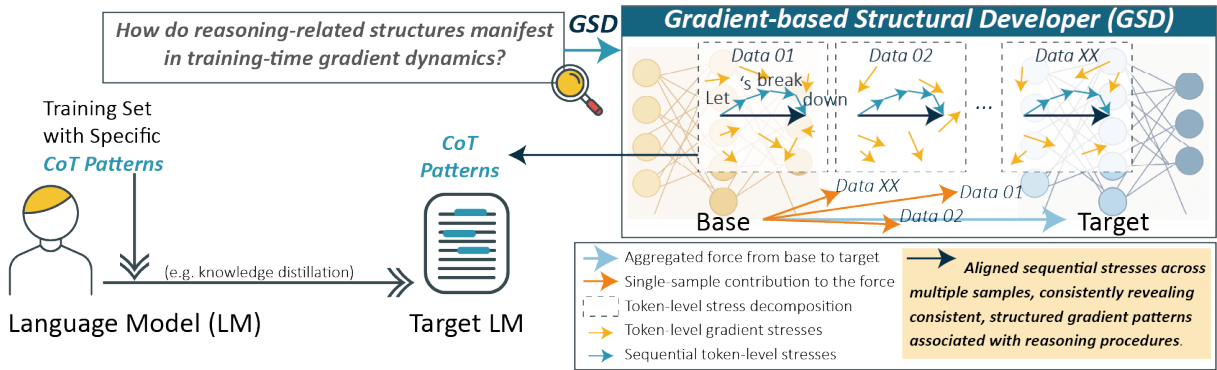


Figure 1: **Problem description and method overview.** We study how reasoning-related structures are expressed during language model (LM) training through span-level gradient dynamics. Chain-of-Thought (CoT) exemplars induce localized gradient signals that, when aggregated across sequences and samples, form consistent directional patterns in parameter space. We propose the **Gradient-based Structural Developer (GSD)** to detect aligned sequential stresses across samples and analyze structured reasoning-related patterns reflected in training-time gradients.

Building on this formulation, we observe that span-level gradient alignment often coincides with semantic similarity, but also appears in cases where text segments with substantially different lexical content follow comparable reasoning procedures. This indicates that span-level gradient alignment captures shared step-level organization that is not directly recoverable from surface semantic representations alone. Figure 1 provides an overview of this analysis. Training on reasoning exemplars produces localized span-level gradient signals, which, when aggregated across sequences and samples, form structured patterns in parameter space. GSD makes these patterns explicit and directly observable, offering a concrete view of how procedural reasoning structure manifests during fine-tuning.

Our contributions are summarized as follows:

- We identify a training-time phenomenon in which dispersed and noisy token-level gradients, when aggregated over contiguous spans and compared across samples, exhibit consistent directional alignment, enabling the unsupervised discovery of latent procedural structure in reasoning data under a gradient aggregation view.
- We demonstrate that gradient-based structural alignment is related to, yet fundamentally distinct from, surface-level semantic similarity: strong gradient alignment can emerge between semantically dissimilar spans that play similar procedural roles, while semantic overlap alone is insufficient to guarantee aligned training dynamics.

- We introduce Gradient-based Structural Developer (GSD) as a diagnostic framework that makes this internal organization observable in a fully unsupervised manner, supported by a signal-noise analysis that explains why span-level aggregation is the minimal effective scale.

2 Related Work

2.1 Chain-of-Thought Reasoning

Explicit CoT prompting has revealed reasoning capabilities in LLMs, but most work focuses on prompt design rather than internal structural effects (Wei et al., 2022; Kojima et al., 2022). Recent studies begin to probe CoT’s influence on model dynamics (Wu et al., 2023). Gradient-based analyses show CoT improves the robustness of token saliency patterns rather than amplifying specific tokens (Wu et al., 2023). Other work finds CoT induces layer-wise functional shifts (Dutta et al., 2024) and that hidden states encode coherent reasoning paths (Wang et al., 2024a). Some studies model latent reasoning using memory modules or continuous hidden-state chaining, avoiding explicit token-level outputs (Orlicki, 2025; Hao et al., 2024). On the learning side, theory shows CoT simulates multi-step optimization (Huang et al., 2025), while empirical evidence links CoT supervision to improved generalization and modular internal circuits. Together, these findings suggest that CoT influences attention patterns, hidden states, and aspects of training dynamics, motivating further investigation into how reasoning-related structure is re-

flected during learning.

2.2 Training-Time Analysis of LLMs

Recent research on fine-tuning LLMs emphasizes aligning outputs through supervised instruction-tuning (SFT) and preference-tuning. SFT on diverse instruction–response datasets has been shown to enable effective zero-shot generalization (Wei et al., 2021). Preference-tuning often employs reinforcement learning from human feedback (RLHF) using proximal policy optimization to align model behavior with human preferences (Ouyang et al., 2022), while alternative approaches perform direct reward-model optimization in a purely supervised fashion (Rafailov et al., 2023). Complementary analyses of training dynamics use gradient-based influence methods to trace how individual examples shape predictions, such as the TracIn technique for quantifying example-level influence (Pruthi et al., 2020). Other studies examine neural tangent kernel evolution to explain network training behavior (Jacot et al., 2018) and employ coreset selection strategies for robust model fitting (Sener and Savarese, 2017). These investigations connect algorithmic choices with token-level effects: instruction tuning tends to reduce token perplexities and enhance general reasoning (Wei et al., 2021), whereas off-policy preference updates can compress probability distributions, lowering confidence on less likely outputs, an effect observed in DPO fine-tuning (Rafailov et al., 2023). Empirical evaluations further demonstrate that PPO-based RLHF can substantially improve human-judged alignment (Ouyang et al., 2022). Understanding these token-level shifts is crucial, as they directly impact overall alignment performance (Ouyang et al., 2022; Rafailov et al., 2023). However, most existing analyses operate at the example or distribution level, leaving the organization of token-level gradient structure across reasoning sequences largely unexplored.

3 Problem Definition and Preliminary Observations

In this work, we examine how reasoning-related structure, encompassing both explicit and implicit forms of CoT, is reflected in span-level gradient dynamics induced by the training objective during LLM fine-tuning. Rather than assessing reasoning capability itself, our objective is to characterize how structured reasoning patterns present in the

training data are expressed through fine-grained gradient organization. Although gradients are computable at a fixed model state, they are not forward-pass representations. Gradients depend explicitly on the training objective and target tokens, and characterize how supervision shapes parameter updates, rather than how the model represents inputs at test time.

To facilitate controlled observation, we adopt a distillation-style training setting. Our focus is not on the teacher–student framework per se, but on how recurring reasoning patterns embedded in training sequences give rise to consistent gradient responses within the student model. We consider token-level gradients as local indicators of how training signals associated with reasoning procedures are distributed and aligned in parameter space.

As a preliminary analysis, we extract token-level gradient contributions corresponding to the first ten tokens of each training example, which typically cover the reasoning prefix, and analyze them at a fixed model state to isolate the structure of training signals. We visualize these gradients using t-SNE to examine whether structured patterns emerge across samples.

Figure 2 presents two complementary views. The left panel illustrates how token-level gradients originating from reasoning prefixes aggregate into coherent directional span-level patterns. The right panel shows clustering among training examples based on prefix gradients, revealing that examples with similar procedural organization tend to exhibit similar gradient structure.

These observations motivate a working hypothesis: **when continuous span-level gradients exhibit consistent directional alignment across multiple samples, such aligned sequential stresses reflect recurring procedural structure in the training data.** The prevalence of such alignment provides an empirical signal of how consistently a particular reasoning procedure is reinforced during training.

4 Methodology

We study how reasoning-related structure is expressed during language model training by analyzing span-level gradient dynamics. We introduce the **Gradient-based Structural Developer (GSD)**, a training-time analysis framework that characterizes how gradient-induced update directions align

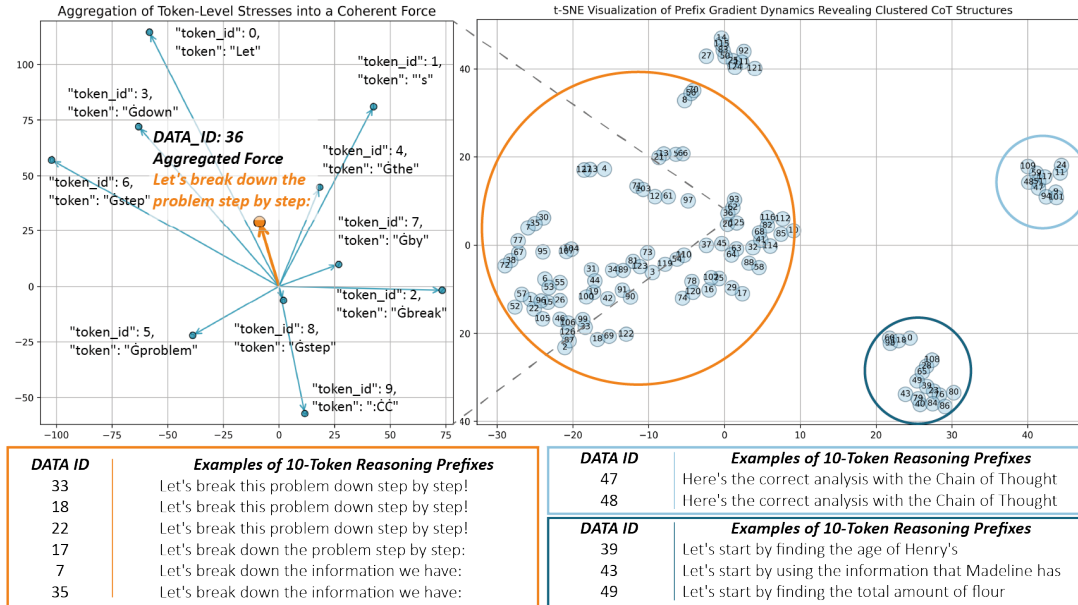


Figure 2: **Visualization of Span-Level Stress Aggregation and Reasoning Prefix Clustering.** Left: Aggregated directional patterns formed by span-level gradients originating from reasoning prefixes. Right: t-SNE visualization of prefix gradients, showing clustering among examples with similar procedural organization.

across samples. Unlike prior work that focuses on generated reasoning traces or forward-pass representations, GSD examines optimization dynamics to reveal shared procedural structure. All analyses are conducted during a single forward pass with gradients computed on a per-sample basis; this ensures each gradient reflects the model's immediate response to an individual example, enabling fine-grained comparison of training signals across sequences.

4.1 Training-time gradient decomposition and interpretation.

Consider an autoregressive language model with parameters θ . Given an input sequence $X = (x_1, \dots, x_L)$, the negative log-likelihood decomposes across positions:

$$\mathcal{L}(\theta; X) = - \sum_{l=1}^L \log \pi_{\theta}(x_l | x_{1:l-1}). \quad (1)$$

The corresponding parameter gradient admits a tokenwise decomposition,

$$\nabla_{\theta} \mathcal{L}(\theta; X) = \sum_{l=1}^L f_l, \quad (2)$$

$$f_l \triangleq -\nabla_{\theta} \log \pi_{\theta}(x_l | x_{1:l-1}), \quad (3)$$

and we interpret each f_l as a *gradient stress* exerted on parameter space by token x_l . This per-token

view emphasizes training-time *update geometry* (how examples push parameters), which is complementary to forward-pass representations (what internal activations encode). GSD's analysis operates on these stresses to surface recurring update directions associated with procedural roles in reasoning tasks.

4.2 Span-level aggregation.

Token-level stresses are often noisy and can be insufficient in isolation to reveal weak but consistent procedural signals. GSD therefore aggregates gradients across contiguous token spans: for a span $[a : b]$ we define the span-level stress

$$\mathcal{F}_{[a:b]} \triangleq \sum_{l=a}^b f_l. \quad (4)$$

To justify this choice formally, we adopt a simple signal-noise decomposition across samples:

$$f_l^{(n)} = s_r + \varepsilon_l^{(n)}, \quad n = 1, \dots, N, \quad (5)$$

$$f_l^{(n)} \triangleq -\nabla_{\theta} \log \pi_{\theta}(x_l^{(n)} | x_{1:l-1}^{(n)}), \quad (6)$$

where $n = 1, \dots, N$ is the sample index, $l = 1, \dots, L^{(n)}$ is the position index within the sequence and $s_r \in \mathbb{R}^P$ is a deterministic vector associated with procedural role r , shared across all tokens (across all samples) that instantiate the same role. And $\varepsilon_l^{(n)}$ is a zero-mean noise term capturing

sample-specific variations. For the aggregated span of length $m = b - a + 1$,

$$\mathcal{F}_{[a:b]}^{(n)} = S_{[a:b]} + E_{[a:b]}^{(n)}, \quad (7)$$

$$S_{[a:b]} = \sum_{l=a}^b s_{r_l}, \quad (8)$$

$$E_{[a:b]}^{(n)} = \sum_{l=a}^b \varepsilon_l^{(n)}. \quad (9)$$

Under this model, the expected cross-sample inner product of aggregated stresses equals $\|S_{[a:b]}\|^2$, while noise-induced fluctuations grow more slowly with span length. When the per-role signals s_{r_l} are directionally aligned across the span, $\|S_{[a:b]}\| = \Theta(m)$ and the signal-to-noise ratio improves roughly as \sqrt{m} , making coherent procedural directions detectable only after aggregation. Conversely, aggregating random tokens yields $\|S_{[a:b]}\| = O(\sqrt{m})$ and provides no signal-to-noise ratio (SNR) advantage. (We state the formal theorem and concentration bounds in Appendix A.) This shows span aggregation is not a heuristic smoothing step but the minimal aggregation scale at which weak, consistent training-time procedures become observable.

4.3 Measuring alignment and discovering structured stresses.

To compare how different spans influence training dynamics we focus on directional agreement in parameter space. Given two spans $[a : b]$ and $[c : d]$ (possibly from different samples), GSD measures

$$\text{GFA}([a : b], [c : d]) \triangleq \frac{\langle \mathcal{F}_{[a:b]}, \mathcal{F}_{[c:d]} \rangle}{\|\mathcal{F}_{[a:b]}\| \|\mathcal{F}_{[c:d]}\|}, \quad (10)$$

which we call Gradient Force Alignment (GFA). High GFA indicates the spans induce parameter updates along similar directions, independent of magnitude.

GSD identifies aligned sequential stresses by finding span sets that (i) occupy comparable relative positions within sequences and (ii) exhibit consistently high pairwise GFA across samples. Empirically, such aligned stresses recur even when surface-level text differs: comparing GFA to embedding-based cosine similarity shows many high-GFA span pairs have low semantic similarity, implying that gradient alignment captures training-time procedural organization that is not reducible to forward-pass semantics. To support the unsupervised discovery of aligned sequential stresses,

the corresponding implementation procedures are provided in Appendix E.

5 Experiments

The experiments in this section examine how reasoning-related structure is reflected in token-level gradient dynamics during language model training. Rather than proposing a new training algorithm, we empirically characterize the patterns revealed by the GSD across different settings. Specifically, we study (i) the unsupervised discovery of gradient-aligned reasoning structures, including semantically diverse cases, (ii) their qualitative correspondence to recognizable reasoning components, and (iii) a controlled diagnostic experiment relating gradient alignment to fine-tuning effectiveness. Together, these experiments provide evidence that aligned sequential stresses offer a meaningful lens for analyzing training-time reasoning dynamics.

5.1 Experimental Setup

We investigate token-level gradient dynamics under controlled Chain-of-Thought (CoT) fine-tuning settings, aiming to understand how reasoning-related structure manifests during training. GSM8K (Cobbe et al., 2021) is used as the primary benchmark due to its explicit multi-step reasoning format, which provides a clear basis for analyzing both surface CoT patterns and latent procedural organization.

To assess the generality of the observed gradient structures, we additionally conduct structural analyses on representative commonsense and logical reasoning benchmarks, including CommonsenseQA (CSQA) (Talmor et al., 2019) and LogiQA (Yu et al., 2020). These datasets are used for qualitative comparison, with detailed results reported in the Appendix B.

All experiments are conducted using the LLaMA-3 8B Base model (Grattafiori et al., 2024) under a controlled CoT-style fine-tuning setup. To enable fine-grained gradient inspection while keeping the parameter space tractable, we adopt LoRA-based adaptation and record token-level gradients during training. Unless otherwise specified, all implementation details—including adaptation modules, gradient extraction procedures, and hyperparameters—are provided in the Appendix F.

All gradient analyses are conducted at a fixed model state, corresponding to the initial step of fine-tuning. This design isolates the structure of

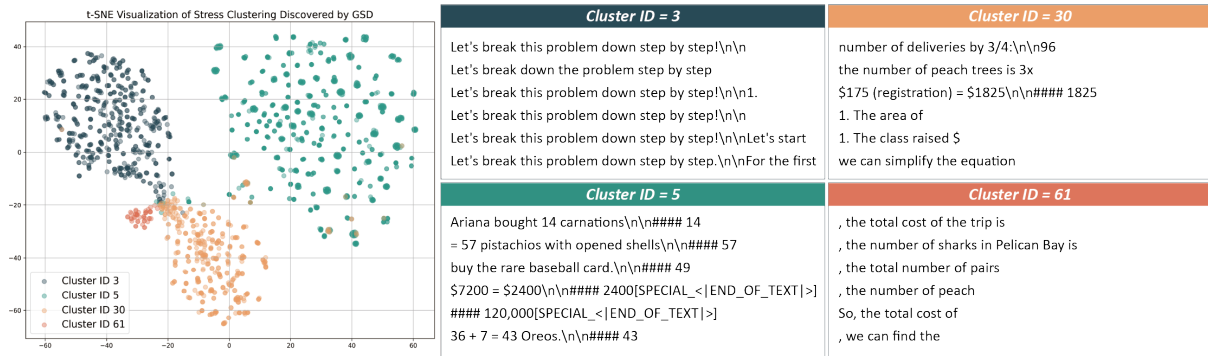


Figure 3: **Unsupervised Discovery of Gradient Structure in Reasoning.** Left: t-SNE visualization of gradient representations derived from continuous token spans during fine-tuning, with points colored by clusters discovered without semantic labels or task annotations. Right: Decoded token spans from selected clusters, illustrating functional consistency within each cluster, corresponding to distinct stages of the reasoning process (e.g., initiation, transition, and conclusion).

training signals induced by reasoning data, without conflating them with feedback effects introduced by parameter updates in later steps.

5.2 GSD Analysis of Reasoning Structure

This section examines whether the similarity patterns observed in gradient space reflect underlying reasoning structure during training, or whether they can be attributed to surface-level semantic similarity. Through unsupervised analysis and controlled contrasts, we show that gradient alignment reflects procedural organization that is neither reducible to semantic overlap nor recoverable by forward-pass similarity measures.

Unsupervised discovery of gradient structure. We examine whether training-time gradients exhibit intrinsic organization without semantic labels or task-specific supervision. Specifically, we extract continuous token spans during fine-tuning, compute their averaged token-level gradient representations, and visualize them using t-SNE. As shown in Figure 3 (left), the resulting gradient space forms multiple well-separated clusters rather than a diffuse distribution. Similar clustering patterns are observed on CommonsenseQA and LogiQA (Appendix B.2, B.1), indicating that this structure is not specific to GSM8K.

Each point represents a span from a different sample, and the separation therefore suggests that certain gradient patterns recur consistently across diverse inputs. Importantly, this organization emerges without access to lexical content, semantic embeddings, or reasoning annotations, implying that it reflects shared update dynamics rather than surface-form similarity.

Decoding representative spans from selected clusters (Figure 3, right) reveals clear functional coherence, corresponding to distinct stages of reasoning such as initiation, transition, and conclusion. Together, these results indicate that fine-tuning induces recurrent, role-specific gradient patterns that are discoverable in a fully unsupervised manner. In order to quantitatively validate the reliability of the observed gradient clusters and the statistical significance of span-level alignment, we provide detailed quantitative evaluations in Appendix C, which include clustering quality metrics such as the Silhouette Score, the Calinski–Harabasz Index and the Davies–Bouldin Index and distributional statistics of GFA scores, thereby confirming the robustness of the gradient-based structural patterns.

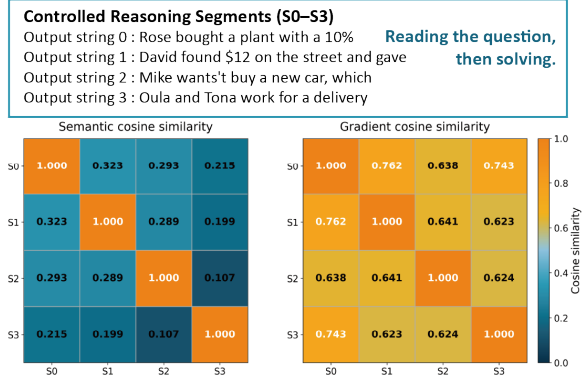


Figure 4: **Semantic similarity vs. gradient similarity for controlled reasoning segments.** Top: four question-reading segments from different GSM8K problems. Bottom: cosine similarity matrices based on semantic embeddings (left) and token-level gradients (right). Despite low semantic similarity, gradient similarity remains high, indicating procedural alignment beyond lexical overlap.

Semantic-different but structurally-aligned reasoning segments. To disentangle structural alignment from surface-level semantic similarity, we construct a controlled comparison using four input segments (S0–S3) drawn from distinct GSM8K problems. Each segment corresponds to the question-reading portion preceding explicit reasoning, sharing a common functional role but differing substantially in lexical content. As shown in Figure 4, semantic cosine similarity between these segments remains low (0.11–0.32), reflecting limited surface overlap. In contrast, cosine similarity computed from span-level gradient vectors is consistently high (0.62–0.76) across all pairs. This discrepancy indicates that segments with divergent semantics can nonetheless induce aligned parameter-space update directions during training, suggesting that gradient similarity reflects procedural commonality beyond what is recoverable from forward-pass semantic representations.

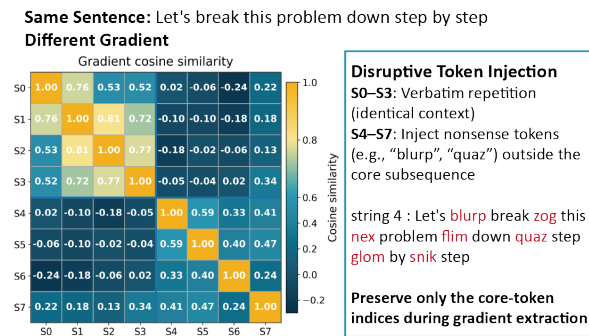


Figure 5: **Identical semantic cores with disrupted gradient alignment.** Gradient cosine similarity for sequences sharing the same core sentence but differing in surrounding tokens. Although the semantic core is preserved, gradient alignment collapses across groups, showing that semantic overlap alone is insufficient to induce aligned updates.

Semantic-identical cores with disrupted gradient organization. We next test whether semantic overlap alone is sufficient to induce gradient alignment. All eight sequences in Figure 5 contain the identical core sentence (“Let’s break this problem down step by step”) and share the same core-token indices during gradient extraction. Sequences S0–S3 repeat the sentence verbatim under identical context, whereas S4–S7 inject nonsensical tokens outside the core subsequence. Although the analyzed token spans are semantically identical across all sequences, gradient similarity between the two groups collapses to near zero or negative

values. This indicates that context perturbations outside the preserved core span are sufficient to reorganize gradient dynamics, breaking alignment even when the analyzed semantic content remains unchanged. Consequently, surface-level token or sentence similarity alone cannot account for the observed gradient structure, highlighting a limitation of forward-pass similarity as a proxy for internal reasoning organization.

5.3 Gradient Structure as a Diagnostic Signal of Training Influence

Building on the preceding analyses of gradient-level structure, we now summarize how such structure manifests at the instance level and how its consistency correlates with downstream fine-tuning behavior. Rather than proposing optimization procedures, this subsection consolidates empirical observations that position gradient alignment as a diagnostic signal of training influence.

Instance-level manifestations of gradient-aligned reasoning structure. Across qualitative case studies, gradient-aligned structure consistently emerges at the token-span level, even without human annotation or explicit reasoning labels. As illustrated in Figure 6, canonical reasoning outputs exhibit both sparse and dense patterns of gradient-aligned spans: in sparse cases, annotations cover as little as ~25% of tokens yet still achieve high cosine similarity to the full training force trajectory (e.g., 0.77–0.86), while dense cases span larger portions of the output (exceeding 40–60%) and yield even stronger alignment (up to 0.95). In contrast, for noisy or non-canonical reasoning outputs, global alignment weakens (e.g., cosine similarity dropping to 0.37–0.71), but localized canonical triggers such as standard reasoning initiators remain consistently highlighted. These examples demonstrate how gradient-level structure manifests at different granularities and under varying degrees of noise.

Correlation between gradient alignment and fine-tuning outcomes. As a complementary signal, we observe that gradient-force aggregation consistency is correlated with downstream fine-tuning behavior in extremely low-resource settings. When fine-tuning with small subsets of data, subsets whose aggregated gradients exhibit higher cosine similarity to a reference force (computed from 800 samples over three epochs) tend to achieve higher GSM8K accuracy. For instance, with 8, 12,

Sparse Annotation on Canonical CoT	Dense Annotation on Canonical CoT	Noisy or Non-Canonical Reasoning
<p>[Let's break this down step by step! Colton had 72 dolphin stickers initially]. He [gave 4 stickers each to 3 friends], so he gave a total of $4 \times 3 = 12$ stickers to his three friends. He gave his friend Mandy 2 more than he gave his three friends, so he gave Mandy $12 + 2 = 14$ stickers. He gave Justin 10 less than Mandy, so he gave Justin $14 - 10 = 4$ stickers. In total, Colton gave away $12 + 14 + 4 = 30$ stickers. To find out how many stickers Colton has left, we subtract the number of stickers he gave away from the initial number of stickers: $72 - 30 = 42$. Colton has 42 stickers left. ##### 42</p> <p>Annotation Ratio: 25.7% Cosine Similarity to Full SFT: 0.86</p>	<p>[Let's break down the problem step by step: 1. Megan's grandma gave her \$125 to start a savings account. 2. She] increased the account by 25% from funds she earned babysitting. To find the amount she added, we can calculate 25% of the [starting balance: $25\% \text{ of } \\$125 = 0.25 \times \\$125 = \\$31.25$. So, she added \$31.25 to the account. 3]. The new balance is: $\\$125 + \\$31.25 = \\$156.25$. 4. Then, it] decreased by 20% when she bought a new pair of shoes. To find the amount she subtracted, we can calculate 20% of the new balance: $20\% \text{ of } \\$156.25 = 0.20 \times \\$156.25 = \\$31.25$. So, she subtracted \$31.25 from the account. 5. The] final balance is: $[\\$156.25 - \\$31.25 = \\$125]$. The final balance is [the same as the starting balance, \$125. To] find the percentage of the starting balance, we can divide the final balance by the starting balance and multiply by 100: $(\\$125 \div \\$125) \times 100 = 100\%$. So, the final balance is 100% of the starting balance. ##### 100</p> <p>Annotation Ratio: 62.3% Cosine Similarity to Full SFT: 0.95</p> <p>[Let's break down the problem step by step: 1. Jim starts with \$80 in his investment portfolio. 2. After 1] year, the portfolio grows by 15%. To find the new value, we calculate: $\\$80 \times (1 + 0.15) = \\$80 \times 1.15 = \\$92$. So, the portfolio] is now worth \$92. 3. Jim then adds another \$28 to his portfolio, making the new total: $[\\$92 + \\$28 = \\$120]$. 4]. After 1 more year, the combined portfolio grows by 10%. To find the new value, we calculate: $\\$120 \times (1 + 0.10) = \\$120 \times 1.10 = \\$132$. So, [the final portfolio value after 2 years is \$132. ##### 132</p> <p>Annotation Ratio: 42.9% Cosine Similarity to Full SFT: 0.89</p>	<p>Here's the correct analysis with the chain of thought: ### Question: Working together, four pugs can clean their house in 45 minutes. In how many minutes will 15 pugs working together be able to clean their house? #### The Numerical Answer] Let's break this problem down step by step! 1. Four] pugs can clean the house in 45 minutes. This means that one pug will take 4 times as long to clean the house, which is: 1 pug : 4 \times 45 minutes = 180 minutes So, one pug will take 180 minutes to clean the house. 2. Since we want to find out how many minutes it will take 15 pugs working together to clean the house, we can divide the time it takes one pug by the number of pugs: 180 minutes (time taken by one pug) \div 15 pugs = 12 minutes ##### 12</p> <p>Annotation Ratio: 7.1% Cosine Similarity to Full SFT: 0.37</p> <p>[Let's break down the time Jake spent watching the show on each day: Monday: Half] a day = 12 hours (since there [are 24 hours in a] day) Tuesday: 4 hours Wednesday: A quarter of a day = 6 hours Total time spent watching the show on Monday, Tuesday, and Wednesday: $12 + 4 + 6 = 22$ hours On Thursday, Jake spent half as much time watching the show as he had done in total throughout the previous few days. Since he spent 22 hours on Monday, Tuesday, and Wednesday, he spent: $22 \text{ hours} / 2 = 11$ hours on Thursday Total time spent watching the show on Monday, Tuesday, Wednesday, and Thursday: $12 + 4 + 6 + 11 = 33$ hours Since the entire show is 52 hours long, Jake still needs to watch: $52 \text{ hours} - 33 \text{ hours} = 19$ hours] on Friday to finish the show. ##### 19</p> <p>Annotation Ratio: 17.7% Cosine Similarity to Full SFT: 0.71</p>

Figure 6: Instance-level manifestations of gradient-aligned reasoning structure. Representative examples illustrating how gradient-aligned token spans appear under different conditions. Left: sparse annotations on canonical CoT outputs, where a small fraction of tokens yields high alignment with the full training force trajectory. Center: dense annotations on canonical CoT outputs, showing extended spans with strong gradient consistency. Right: noisy or non-canonical reasoning outputs, where global alignment weakens but localized canonical triggers remain detectable. Each example reports the annotation ratio and cosine similarity to the full SFT force trajectory.

and 16 training samples, higher alignment corresponds to accuracy differences of approximately 7.0, 6.0, and 3.1 percentage points, respectively. We emphasize that this analysis is not intended to propose a data selection or optimization strategy, but rather to illustrate that gradient alignment provides a measurable diagnostic signal that covaries with adaptation outcomes. Detailed experimental setup and additional results are provided in Appendix D. For additional empirical support, Appendix G provides expanded results. These supplementary results reinforce the robustness of our core findings.

6 Conclusion

This work examines how reasoning-related structure is reflected during Chain-of-Thought fine-tuning by analyzing span-level gradients in parameter space. Rather than relying on generated reasoning traces, we study gradients induced by training examples and find that different stages of reasoning correspond to recurring, role-specific patterns of gradient organization.

We observe that such gradient alignment persists across samples even when surface-level semantic

similarity is low, while semantic overlap alone does not guarantee aligned gradients under contextual perturbations. This contrast suggests that gradient space captures procedural organization not directly accessible through forward-pass semantic representations. These findings support a geometric view of reasoning internalization, where repeated procedural roles during fine-tuning exert consistent directional influences on model parameters. Span-level gradients thus provide an interpretable diagnostic signal for understanding how reasoning-related regularities are reinforced during adaptation.

We emphasize that gradient alignment is not a definition of reasoning itself, nor a replacement for behavioral evaluation. Instead, it serves as an interpretable diagnostic signal that makes aspects of training-induced reasoning organization explicit. By grounding analysis in parameter-space structure rather than surface outputs, this work offers a complementary explainability-oriented perspective on how structured reasoning behavior is internalized during model adaptation.

7 Limitations

This work is intended as a diagnostic study of how reasoning-related procedures are reflected in training-time gradient organization, and its limitations primarily stem from this analytical focus.

First, while GSD does not aim to establish full interventional causality between specific training manipulations and downstream reasoning behavior, it goes beyond purely correlational observation. Our analyses are conducted under controlled conditions, including a fixed model state and targeted structural perturbations, which allow us to isolate how procedural organization influences gradient alignment independently of surface semantics. In this sense, the results support a weak but structured causal interpretation: procedural coherence is necessary for the emergence of aligned gradient patterns, whereas semantic similarity alone is insufficient. Nevertheless, we do not claim that these patterns constitute a complete causal model of reasoning acquisition. Establishing necessity and sufficiency through explicit training-time interventions or counterfactual optimization trajectories remains an important direction for future work.

Second, the framework focuses on the consistency of internalized procedural structure rather than the correctness or optimality of the resulting reasoning outcomes. High gradient alignment indicates convergent update directions induced by recurring procedural roles, but it does not guarantee that the corresponding reasoning paths are valid, faithful, or desirable. Evaluating reasoning correctness and faithfulness requires complementary behavioral and task-level analyses.

Finally, GSD emphasizes linearly aggregatable gradient structure by design, leveraging the additive nature of gradients to expose dominant procedural regularities. While this yields a minimal and interpretable diagnostic signal, it may underrepresent highly distributed, nonlinear, or weakly aligned effects that do not manifest as coherent gradient directions. Exploring richer aggregation schemes or combining gradient-based diagnostics with intervention-based analyses is left for future work.

In addition, this work does not introduce new datasets, collect user data, or involve human subjects. All analyses are conducted on publicly available reasoning benchmarks, and no personally identifying information is used or generated. Since the study focuses on training-time gradient diagnostics

rather than data curation or model deployment, it does not introduce new risks related to data privacy, data misuse, or offensive content generation beyond those already associated with the underlying benchmarks.

A potential practical risk lies in over-interpreting gradient alignment signals as direct causal explanations or optimization prescriptions. To mitigate this, we explicitly frame GSD as a diagnostic and analysis tool rather than a training algorithm or intervention strategy. The results are intended to support careful inspection of training-time structure, not to justify automated data selection, model editing, or deployment decisions without further validation. We therefore emphasize that any downstream use of gradient-based diagnostics should be accompanied by task-level evaluation and, where appropriate, explicit intervention-based analysis.

References

- Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. 2013. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press.
- Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- David L Davies and Donald W Bouldin. 2009. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.
- Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. 2023. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*.
- Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. 2024. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *arXiv preprint arXiv:2402.18312*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Shangmin Guo, Yi Ren, Stefano V Albrecht, and Kenny Smith. 2024. Iptnk: Better generalisation with less data via sample interaction during learning. *arXiv preprint arXiv:2401.08808*.

672	Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li,	Leonardo Ranaldi, Marco Valentino, and Andre Fre-	726
673	Zhiting Hu, Jason Weston, and Yuandong Tian. 2024.	itas. 2025. Improving chain-of-thought reasoning via	727
674	Training large language models to reason in a contin-	quasi-symbolic abstractions. pages 17222–17240.	728
675	uous latent space. <i>arXiv preprint arXiv:2412.06769</i> .		
676	Jianhao Huang, Zixuan Wang, and Jason D Lee. 2025.	Yi Ren, Shangmin Guo, Linlu Qiu, Bailin Wang, and	729
677	Transformers learn to implement multi-step gradi-	Danica J Sutherland. 2024. Bias amplification in	730
678	ent descent with chain of thought. <i>arXiv preprint</i>	language model evolution: An iterated learning per-	731
679	<i>arXiv:2502.21212</i> .	spective. <i>Advances in Neural Information Processing</i>	732
		<i>Systems</i> , 37:38629–38664.	733
680	Arthur Jacot, Franck Gabriel, and Clément Hongler.	Yi Ren, Shangmin Guo, and Danica J Sutherland. 2022.	734
681	2018. Neural tangent kernel: Convergence and gen-	Better supervisory signals by observing learning	735
682	eralization in neural networks. <i>Advances in neural</i>	paths. <i>arXiv preprint arXiv:2203.02485</i> .	736
683	<i>information processing systems</i> , 31.		
684	Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yu-	Peter J Rousseeuw. 1987. Silhouettes: a graphical aid	737
685	taka Matsuo, and Yusuke Iwasawa. 2022. Large lan-	to the interpretation and validation of cluster analysis.	738
686	guage models are zero-shot reasoners. <i>Advances in</i>	<i>Journal of computational and applied mathematics</i> ,	739
687	<i>neural information processing systems</i> , 35:22199–	20:53–65.	740
688	22213.		
689	Tamera Lanham, Anna Chen, Ansh Radhakrishnan,	Ozan Sener and Silvio Savarese. 2017. Active learn-	741
690	Benoit Steiner, Carson Denison, Danny Hernandez,	ing for convolutional neural networks: A core-set	742
691	Dustin Li, Esin Durmus, Evan Hubinger, Jackson	approach. <i>arXiv preprint arXiv:1708.00489</i> .	743
692	Kernion, and 1 others. 2023. Measuring faithful-		
693	ness in chain-of-thought reasoning. <i>arXiv preprint</i>	Alon Talmor, Jonathan Herzig, Nicholas Lourie, and	744
694	<i>arXiv:2307.13702</i> .	Jonathan Berant. 2019. Commonsenseqa: A question	745
695	Ayeong Lee, Ethan Che, and Tianyi Peng. 2025.	answering challenge targeting commonsense knowl-	746
696	How well do llms compress their own chain-of-	edge. In <i>Proceedings of the 2019 Conference of</i>	747
697	thought? a token complexity approach. <i>arXiv</i>	<i>the North American Chapter of the Association for</i>	748
698	<i>preprint arXiv:2503.01141</i> .	<i>Computational Linguistics: Human Language Tech-</i>	749
		<i>nologies, Volume 1 (Long and Short Papers)</i> , pages	750
699	Jindong Li, Yali Fu, Li Fan, Jiahong Liu, Yao Shu,	4149–4158.	751
700	Chengwei Qin, Menglin Yang, Irwin King, and Rex		
701	Ying. 2025. Implicit reasoning in large language	Roman Vershynin. 2018. <i>High-Dimensional Prob-</i>	752
702	models: A comprehensive survey. <i>arXiv preprint</i>	<i>ability: An Introduction with Applications in Data</i>	753
703	<i>arXiv:2509.02350</i> .	<i>Science</i> . Cambridge Series in Statistical and Prob-	754
		abilistic Mathematics. Cambridge University Press.	755
704	Jia Liu, Yue Wang, Zhiqi Lin, Min Chen, Yixue Hao,	Yiming Wang, Pei Zhang, Baosong Yang, Derek F	756
705	and Long Hu. 2024. Natural language fine-tuning.	Wong, and Rui Wang. 2024a. Latent space chain-of-	757
706	<i>arXiv preprint arXiv:2412.20382</i> .	embedding enables output-free LLM self-evaluation.	758
		<i>arXiv preprint arXiv:2410.13640</i> .	759
707	José I Orlicki. 2025. Beyond words: A latent memory	Yu Wang, Shiwan Zhao, Zhihu Wang, Heyuan Huang,	760
708	approach to internal reasoning in llms. <i>arXiv preprint</i>	Ming Fan, Yubo Zhang, Zhixing Wang, Haijun Wang,	761
709	<i>arXiv:2502.21030</i> .	and Ting Liu. 2024b. Strategic chain-of-thought:	762
		Guiding accurate reasoning in llms through strategy	763
710	Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida,	elicitation. <i>arXiv preprint arXiv:2409.03271</i> .	764
711	Carroll Wainwright, Pamela Mishkin, Chong Zhang,		
712	Sandhini Agarwal, Katarina Slama, Alex Ray, and 1	Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin	765
713	others. 2022. Training language models to follow in-	Guu, Adams Wei Yu, Brian Lester, Nan Du, An-	766
714	structions with human feedback. <i>Advances in neural</i>	drew M Dai, and Quoc V Le. 2021. Finetuned lan-	767
715	<i>information processing systems</i> , 35:27730–27744.	guage models are zero-shot learners. <i>arXiv preprint</i>	768
		<i>arXiv:2109.01652</i> .	769
716	Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten	770
717	Sundararajan. 2020. Estimating training data influ-	Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,	771
718	ence by tracing gradient descent. <i>Advances in Neural</i>	and 1 others. 2022. Chain-of-thought prompting elic-	772
719	<i>Information Processing Systems</i> , 33:19920–19930.	its reasoning in large language models. <i>Advances</i>	773
		<i>in neural information processing systems</i> , 35:24824–	774
720	Rafael Rafailov, Archit Sharma, Eric Mitchell, Christo-	24837.	775
721	pher D Manning, Stefano Ermon, and Chelsea Finn.	Skyler Wu, Eric Meng Shen, Charumathi Badrinath,	776
722	2023. Direct preference optimization: Your lan-	Jiaqi Ma, and Himabindu Lakkaraju. 2023. Ana-	777
723	guage model is secretly a reward model. <i>Advances in</i>	lyzing chain-of-thought prompting in large language	778
724	<i>Neural Information Processing Systems</i> , 36:53728–	models via gradient-based feature attributions. <i>arXiv</i>	779
725	53741.	<i>preprint arXiv:2307.13339</i> .	780

781 Weihao Yu, Zihang Jiang, Yanfei Dong, and Ji-
782 ashi Feng. 2020. [Reclor: A reading comprehension dataset requiring logical reasoning](#). *Preprint*,
783 arXiv:2002.04326.
784

785 Eric Zelikman, Georges Harik, Yijia Shao, Varuna
786 Jayasiri, Nick Haber, and Noah D Goodman. 2024.
787 Quiet-star: Language models can teach themselves
788 to think before speaking.

789 Hugh Zhang and David C Parkes. 2023. Chain-of-
790 thought reasoning is a policy improvement operator.
791 *arXiv preprint arXiv:2309.08589*.

792 Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei
793 Gao, and Min Lin. 2024. Chain of preference opti-
794 mization: Improving chain-of-thought reasoning in
795 llms. *Advances in Neural Information Processing*
796 *Systems*, 37:333–356.

797	Contents			
798	1 Introduction	1		
799	2 Related Work	2		
800	2.1 Chain-of-Thought Reasoning	2		
801	2.2 Training-Time Analysis of LLMs .	3		
802	3 Problem Definition and Preliminary Observations	3		
803				
804	4 Methodology	3		
805	4.1 Training-time gradient decomposition and interpretation.	4		
806	4.2 Span-level aggregation.	4		
807	4.3 Measuring alignment and discovering structured stresses.	5		
808				
809				
810	5 Experiments	5		
811	5.1 Experimental Setup	5		
812	5.2 GSD Analysis of Reasoning Structure	6		
813	5.3 Gradient Structure as a Diagnostic Signal of Training Influence	7		
814				
815				
816	6 Conclusion	8		
817	7 Limitations	9		
818	Appendix	12		
819	A Theoretical foundations and methodological clarifications for GSD	13		
820	A.1 Notation and standing assumptions	13		
821	A.2 Signal-noise model for span aggregation	13		
822	A.3 Cross-sample alignment and SNR improvement	14		
823	A.4 Exact identities and relation to Jacobian geometry	14		
824				
825				
826				
827				
828	B Generalization to Broader Reasoning Tasks	15		
829	B.1 Logical Deduction (LogiQA)	15		
830	B.1.1 Construction of the Distilled Logical Deduction Dataset (LogiQA)	15		
831	B.1.2 Structural Stress Patterns in Logical Deduction Tasks	15		
832	B.1.3 Qualitative Analysis of Clustered Stress Patterns .	16		
833	B.1.4 Per-Sample Token Trace Examples	17		
834				
835				
836				
837				
838				
839				
	B.2 Commonsense Reasoning (CSQA)	19		840
	B.2.1 Construction of the Distilled Commonsense Reasoning Dataset (CSQA) . .	19		841
	B.2.2 Structural Stress Patterns in Commonsense Reasoning Tasks	19		842
	B.2.3 Qualitative Analysis of Clustered Stress Patterns .	19		843
				844
				845
				846
				847
				848
	C Quantitative Evaluation of Gradient-Based Alignment	21		849
	C.1 Clustering Quality of GSD Patterns	21		850
	C.2 Distributional Statistics of GFA Scores	21		851
				852
				853
	D Gradient Force Aggregation and Fine-Tuning Behavior	23		854
	D.1 Construction of the Target Force and Candidate Pool	24		855
	D.2 Subset Aggregation and Alignment Measurement	24		856
	D.3 Fine-Tuning Setup and Evaluation	24		857
	D.4 Interpretation and Scope	24		858
				859
				860
				861
	E Algorithmic Details	25		862
	E.1 Unsupervised Clustering of Sequential Stresses	25		863
	E.2 Greedy Force-Based Data Selection Algorithm	25		864
				865
				866
	F Implementation Details	25		867
	F.1 Training Hyperparameters and Infrastructure	25		868
	F.2 Token-Level Gradient Tracing Protocol	26		869
	F.3 LoRA Configuration and Optimization Settings	27		870
				871
				872
				873
	G Additional Results	27		874
	G.1 Expanded Visualizations of Stress Clusters	27		875
	G.1.1 Strongly Separated Clusters	27		876
	G.1.2 Moderately Cohesive Clusters	30		877
	G.1.3 Low-Cohesion or Noisy Clusters	33		878
	G.2 Per-Sample Token Trace Examples	34		879
				880
				881
				882

A Theoretical foundations and methodological clarifications for GSD

This appendix provides formal justification, analytic identities, and detailed clarifications supporting the methodology introduced in Section 3. Its purpose is to make explicit the assumptions, mathematical structure, and statistical reasoning underlying Gradient-based Structural Developer (GSD), without introducing additional modeling components beyond those used in the main text.

A.1 Notation and standing assumptions

We fix notation used throughout the analysis. Let $\theta \in \mathbb{R}^P$ denote the full parameter vector of the language model. For a training example indexed by n , we denote the conditioning prefix by $X^{(n)}$ and the autoregressive target sequence by $Y^{(n)} = (y_1^{(n)}, \dots, y_{L^{(n)}}^{(n)})$. When unambiguous, we omit the superscript.

The per-token negative log-likelihood loss at position l is

$$\ell_l(\theta) \triangleq -\log \pi_\theta(y_l \mid y_{1:l-1}, X), \quad (11)$$

and the per-sample loss decomposes as

$$\mathcal{L}(\theta) = \sum_{l=1}^L \ell_l(\theta). \quad (12)$$

The corresponding token-level gradient (stress) is defined as

$$f_l \triangleq \nabla_\theta \ell_l(\theta) \in \mathbb{R}^P, \quad (13)$$

and for a contiguous span $[a : b]$ we define the span-level stress

$$\mathcal{F}_{[a:b]} \triangleq \sum_{l=a}^b f_l. \quad (14)$$

For clarity in analytic derivations, we denote the model logits at position l by $z_l(\theta) \in \mathbb{R}^{|V|}$ and the corresponding Jacobian by

$$J_l(\theta) \triangleq \frac{\partial z_l(\theta)}{\partial \theta} \in \mathbb{R}^{|V| \times P}. \quad (15)$$

By the chain rule, the token gradient admits the exact decomposition

$$f_l = J_l(\theta)^\top g_l, \quad (16)$$

where $g_l \in \mathbb{R}^{|V|}$ is the derivative of ℓ_l with respect to z_l .

A.2 Signal–noise model for span aggregation

To formalize why span-level aggregation reveals procedural structure that is obscured at the token level, we adopt a structured signal–noise decomposition that directly models cross-sample role sharing.

Let \mathcal{R} denote the set of procedural roles (e.g., “define variable”, “perform arithmetic”, “conclude”). For each sample n and token position l we denote by $r_l^{(n)} \in \mathcal{R}$ the procedural role instantiated by that token. We model the token-level gradient as

$$f_l^{(n)} = s_{r_l^{(n)}} + \varepsilon_l^{(n)}, \quad n = 1, \dots, N, \quad (17)$$

where $s_r \in \mathbb{R}^P$ is a deterministic vector associated with role r (shared across all tokens and samples that instantiate role r), and $\varepsilon_l^{(n)} \in \mathbb{R}^P$ is a zero-mean noise term capturing sample-specific variation and surface-form differences.

For analytic transparency we impose the following standing conditions:

1. **Noise independence across samples:** $\varepsilon_l^{(n)}$ and $\varepsilon_{l'}^{(n')}$ are independent for $n \neq n'$.

2. **Bounded within-sample accumulation:** for any span $[a : b]$ of length $m = b - a + 1$,

$$\mathbb{E} \left\| \sum_{l=a}^b \varepsilon_l^{(n)} \right\|^2 \leq \sigma^2 m, \quad (18)$$

allowing for limited within-sample correlations while controlling cumulative variance.

3. **Signal–noise orthogonality:** $\mathbb{E} \langle s_r, \varepsilon_l^{(n)} \rangle = 0$ for all r, l, n .

For a contiguous span $[a : b]$ with role sequence (r_a, \dots, r_b) the aggregated stress decomposes as

$$\mathcal{F}_{[a:b]}^{(n)} = S_{[a:b]} + E_{[a:b]}^{(n)}, \quad (19)$$

$$S_{[a:b]} = \sum_{l=a}^b s_{r_l}, \quad (20)$$

$$E_{[a:b]}^{(n)} = \sum_{l=a}^b \varepsilon_l^{(n)}. \quad (21)$$

Note that $S_{[a:b]}$ is deterministic and shared across samples whose span instantiates the same role sequence, whereas $E_{[a:b]}^{(n)}$ varies across samples. The simpler, position-indexed model $f_l^{(n)} = s_l + \varepsilon_l^{(n)}$

is recovered as a special case when each absolute position corresponds deterministically to a unique procedural role.

A.3 Cross-sample alignment and SNR improvement

Building on the signal–noise decomposition introduced above, consider two independent samples $n \neq m$ whose spans $[a : b]$ instantiate the same procedural role sequence (r_a, \dots, r_b) . Recall

$$\mathcal{F}_{[a:b]}^{(k)} = S_{[a:b]} + E_{[a:b]}^{(k)}, \quad k \in \{n, m\}. \quad (22)$$

Linearity and the zero-mean / independence assumptions yield the exact expectation

$$\mathbb{E}[\langle \mathcal{F}_{[a:b]}^{(n)}, \mathcal{F}_{[a:b]}^{(m)} \rangle] = \|S_{[a:b]}\|^2. \quad (23)$$

Define the deviation

$$\begin{aligned} \Delta &\triangleq \langle \mathcal{F}_{[a:b]}^{(n)}, \mathcal{F}_{[a:b]}^{(m)} \rangle - \|S_{[a:b]}\|^2 \\ &= A_n + A_m + B, \end{aligned} \quad (24)$$

where $A_k = \langle S_{[a:b]}, E_{[a:b]}^{(k)} \rangle$ and $B = \langle E_{[a:b]}^{(n)}, E_{[a:b]}^{(m)} \rangle$. By the bounded-accumulation assumption,

$$\mathbb{E}\|E_{[a:b]}^{(k)}\|^2 \leq \sigma^2 m, \quad (25)$$

and therefore

$$\text{Var}(A_k) = \mathbb{E}[A_k^2] \leq \|S_{[a:b]}\|^2 \sigma^2 m, \quad (26)$$

$$\mathbb{E}[B^2] \leq \sigma^4 m^2. \quad (27)$$

Using independence across samples and zero-mean of each term, we obtain the conservative variance bound

$$\text{Var}(\Delta) \leq 2\|S_{[a:b]}\|^2 \sigma^2 m + \sigma^4 m^2. \quad (28)$$

Weak (moment-based) concentration. Without further distributional assumptions, Chebyshev’s inequality yields for any $t > 0$

$$\mathbb{P}(|\Delta| \geq t) \leq \frac{\text{Var}(\Delta)}{t^2}, \quad (29)$$

with $\text{Var}(\Delta)$ bounded by (28).

Strong (exponential) concentration under sub-Gaussian noise. If additionally each scalar projection $\langle v, \varepsilon_l^{(n)} \rangle$ is sub-Gaussian with parameter

σ (uniformly over unit v), then one obtains exponential tails. Concretely, there exists an absolute constant $c > 0$ such that for all $t > 0$

$$\begin{aligned} \mathbb{P}(|\Delta| \geq t) &\leq 2 \exp\left(-c \right. \\ &\left. \min\left\{\frac{t^2}{\sigma^2 m \|S_{[a:b]}\|^2 + \sigma^4 m^2}, \frac{t}{\sigma^2 m}\right\}\right). \end{aligned} \quad (30)$$

(Proof sketch: linear terms are controlled via standard sub-Gaussian concentration bounds for linear forms of independent noise, while the bilinear term is handled by conditional linearization and decoupling arguments; see (Boucheron et al., 2013; Vershynin, 2018) for standard results.)

SNR scaling under signal alignment. We define the effective signal-to-noise ratio of the inner product as $\text{SNR} \triangleq \|S_{[a:b]}\|^2 / \sqrt{\text{Var}(\Delta)}$. To quantify $\|S_{[a:b]}\|$ we adopt a concrete sufficient condition for directional coherence: suppose there exists a unit vector u and $\mu_{\min} > 0$ such that $\langle s_{r_l}, u \rangle \geq \mu_{\min}$ for every $l \in [a : b]$. Then

$$\|S_{[a:b]}\| \geq \mu_{\min} m, \quad (31)$$

hence by (28) the variance is $O(m^3)$ and the SNR of the inner product grows as $\Omega(\sqrt{m})$. If instead the s_{r_l} are incoherent so that $\|S_{[a:b]}\| = O(\sqrt{m})$, no asymptotic SNR gain occurs.

A.4 Exact identities and relation to Jacobian geometry

Using the chain-rule decomposition $f_l = J_l^\top g_l$, the inner product between two token gradients can be written as

$$\langle f_i, f_j \rangle = \langle J_i^\top g_i, J_j^\top g_j \rangle = g_i^\top (J_i J_j^\top) g_j, \quad (32)$$

where $J_\ell(\theta) \in \mathbb{R}^{|V| \times P}$ denotes the Jacobian of the logits at position ℓ and $g_\ell \in \mathbb{R}^{|V|}$ is the derivative of the per-token loss with respect to the logits. In particular, $J_i J_j^\top \in \mathbb{R}^{|V| \times |V|}$ is the token-level NTK-like matrix between positions i and j ; we emphasize that in practice we do not form this large $|V| \times |V|$ matrix but compute the inner product via the parameter-space vectors $J_i^\top g_i \in \mathbb{R}^P$ and $J_j^\top g_j \in \mathbb{R}^P$.

By linearity, for two spans $[a : b]$ and $[c : d]$ the span-level inner product admits the exact decomposition

$$\langle \mathcal{F}_{[a:b]}, \mathcal{F}_{[c:d]} \rangle = \sum_{i=a}^b \sum_{j=c}^d g_i^\top (J_i J_j^\top) g_j. \quad (33)$$

This identity shows that Gradient Force Alignment (GFA) measures a *gradient-weighted* NTK similarity: each kernel entry $J_i J_j^\top$ is weighted by the model’s error vectors g_i and g_j . Accordingly, high GFA between spans can arise from:

1. **aligned error vectors** g_i, g_j in output space,
2. **shared Jacobian subspaces** so that J_i and J_j map errors to similar parameter directions, or
3. **a consistent interaction** of both effects.

These three sources correspond naturally to the SNR mechanisms analyzed in the previous subsection: for example, repeated occurrence of the same procedural role tends to produce coherent g -patterns and/or coherent Jacobian actions, which yields $\|S_{[a:b]}\| = \Theta(m)$ and thus the span-level SNR amplification previously described. This decomposition therefore motivates the attribution diagnostics in the main text that separate error-driven alignment from representational (Jacobian) alignment.

B Generalization to Broader Reasoning Tasks

B.1 Logical Deduction (LogiQA)

B.1.1 Construction of the Distilled Logical Deduction Dataset (LogiQA)

We construct LogiQA, a logical reasoning dataset built upon The ReClor Dataset (Yu et al., 2020), in which each instance is augmented with a step-by-step Chain-of-Thought explanation generated by a large language model. This dataset aims to improve model explainability and reasoning transparency in complex logical deduction tasks.

Specifically, we take the original questions, answer choices, and correct answers from The ReClor Dataset, and prompt the DeepSeek-R1 model to generate detailed reasoning steps for each instance. Each data sample in LogiQA includes four components: the question text, candidate options, the correct answer label, and the generated CoT explanation.

B.1.2 Structural Stress Patterns in Logical Deduction Tasks

We apply the GSD framework to the LogiQA dataset to examine whether structured stress patterns also emerge in tasks involving complex logical reasoning. By extracting token-level stress vectors across 330 samples and performing t-SNE

projection, we identify 7 prominent clusters with size ≥ 40 , as visualized in Figure 7.

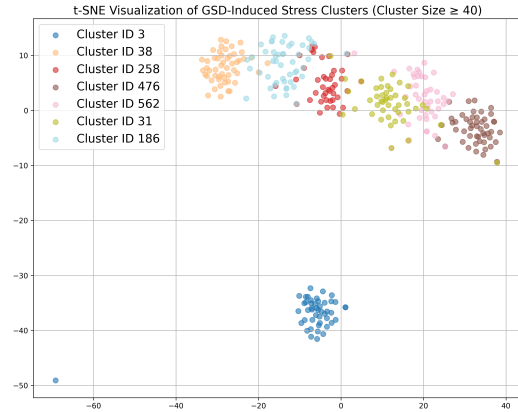


Figure 7: t-SNE visualization of GSD-induced token-level stress clusters on the LogiQA dataset. Only clusters with size ≥ 40 are shown. Each point represents a token-level stress vector.

We evaluate the structural coherence of these clusters using standard metrics:

- **Silhouette Score:** 0.4503 (higher is better), indicating clear intra-cluster consistency.
- **Calinski-Harabasz Index:** 810.60, reflecting good separation between clusters.
- **Davies-Bouldin Index:** 0.8563 (lower is better), confirming inter-cluster distinctiveness.

Table 1 presents the average Silhouette Scores for each cluster:

Cluster ID 3 stands out with a very high silhouette score (0.8431), indicating tight internal consistency. It predominantly captures template-driven reasoning initiators such as “Let’s analyze the argument step by step...” and “To determine the logical conclusion...”. Clusters 38 and 476 also exhibit high coherence (≥ 0.58), typically involving conditional reasoning markers and core logical deductions. Lower-scoring clusters (e.g., 31 and 562) tend to contain more diverse intermediate or paraphrased logical constructs.

The emergence of interpretable clusters in LogiQA confirms the robustness of GSD in capturing logic-centric stress alignment. The alignment structure remains consistent with our findings in math and commonsense domains, suggesting that logical deduction—despite higher abstraction—exhibits recurrent stress dynamics across token spans, especially in transitions between assumptions, logical bridges, and conclusions.

Table 1: Average Silhouette Score per Cluster (LogiQA, Cluster Size ≥ 40)

Cluster ID	3	38	258	476	562	31	186
Silhouette	0.8431	0.6192	0.3883	0.5880	0.2716	0.1245	0.2805

B.1.3 Qualitative Analysis of Clustered Stress Patterns

To further understand the internal reasoning structure captured by GSD, we analyze representative token spans within each major cluster (size ≥ 40). The results show clear functional differentiation across clusters, revealing distinct roles in the reasoning chain from initiation to flaw detection and argumentative evaluation.

Cluster ID 3. This cluster is highly consistent (Silhouette score: 0.8431), dominated by standardized reasoning initiators such as “Let’s think step by step.” and “First, let’s analyze the problem.” Tokens in this cluster mark the beginning of structured deduction processes and occur frequently in both model-generated CoT and ground truth templates. Many samples include enumerated premises (e.g., “1.”, “2.”) and logical decomposition phrases (e.g., “Let’s break this down logically.”), confirming this cluster’s role as a canonical CoT preamble anchor.

step by step. First, let’s analyze the problem.

step by step.

Truax was sent before Friday

think step by step. First,

First, let’s analyze the problem.

think step by step. First

this down logically. **

ignore negative consequences of actions

Let’s break this down logically. The argument concludes

The conclusion is that only three of Miguel’s four family members will

Cluster ID 38. This cluster (Silhouette score: 0.6192) captures causal evaluation and inference bridging. Common tokens include numeric data, statistical references, and conditional phrases like “for the conclusion to hold,” and “we must assume.” The cluster is rich in middle-stage reasoning where the model is required to test logical validity, align evidence with premises, or identify hidden assumptions. This indicates stress alignment around epistemic control points in logical

arguments.

body to injury/infection

follow logically, we must

at least one shipment to

construction tools are not returnable for store credit.

justify offering the service.

histidine production cannot explain

with suppression; it focuses on

owners belong to groups that

let’s analyze the problem

completion, we must find a

Cluster ID 258. Tokens in Cluster 258 (Silhouette score: 0.3883) span diverse argumentative purposes, including counter-example handling, claim evaluation, and real-world analogies. Frequent patterns include “we need an assumption that...”, “therefore...”, and domain references (e.g., “PRP,” “satellites,” “injuries”). This cluster represents transitional reasoning units often linking evidence with claims or identifying comparison flaws.

think step by step.

need an assumption that links

think step by step.

. Therefore, the theorists

contents) or could be

returnable for store credit. Since

returnable for a refund

claims cost-saving benefits of double hulls

think step by step.

creatures symbolize horses rather

Cluster ID 476. This cluster (Silhouette score: 0.5880) reflects structured counter-argumentation and weakening strategies. Tokens include “introducing a confounding variable,” “alternative explanation,” “fails to address...” and “Option X weakens the argument...”. Its contents are well-aligned with samples involving option testing, assumption negation, and causal interference detection. This cluster highlights the model’s ability to simulate skeptical evaluation and alternative hypothesis generation.

1203	but doesn't refute the causal link	weaken the argument by implying	1253
1204	people ignore objectionable	conditions in Ireland must be	1254
1205	consequences of actions supporting	argument praises space-based satellites	1255
1206	inherent downsides.	for enabling early environmental problem	1256
1207	potential ozone layer damage from	a moderate scenario that could	1257
1208	due to the absence of	without special exhibitions**,	1258
1209	government regulation due to the	attendance	1259
1210	Analyzing the options:- **	reasoning to hold, the	1260
1211	highlighting the severity of water	alone doesn't explain why	1261
1212	scarcity		
1213	the recovery must stem from	Cluster ID 186. Cluster 186 (Silhouette score:	1262
1214	express unconscious thoughts but doesn	0.2805) aligns with interpretive explanation and	1263
		pragmatic support statements. Examples include	1264
1215	Cluster ID 562. Cluster 562 (Silhouette score:	"critics' ability to help," "assumes	1265
1216	0.2716) contains tokens involved in premises	accurate identification," "citizen	1266
1217	reweighting and nuanced critique. Tokens like	action," "security principle," and similar	1267
1218	"points to consider," "just the lack of	evaluative or supportive units. These tokens help	1268
1219	reward," "anarchists commit violence,"	to reinforce the conclusion or highlight potential	1269
1220	and "another ideology..." reflect subtle argu-	flaws in assumptions, consistent with the cluster's	1270
1221	ment evaluation and hypothesis differentiation. The	mid-to-late role in argumentative structure.	1271
1222	lower silhouette score corresponds to semantic vari-	is flawed), we need an assumption that	1272
1223	ability, as tokens encode more heterogeneous argu-	critics' ability to help	1273
1224	mentative forms.	is undermined. If it does not decline,	1274
1225	evaluation, which is irrelevant	the argument is supported.	1275
1226	firms with PRP had 13% higher	think step by step.	1276
1227	productivity.	directly ties citizen action to	1277
1228	points to consider:	this down logically. The	1278
1229	Since *selected* construction	Correct. The argument never	1279
1230	satellites for enabling early	's think step by step.	1280
1231	environmental problem detection but	lifestyle) might explain both	1281
1232	overlooking potential ozone layer	B.1.4 Per-Sample Token Trace Examples	1282
1233	damage from	SAMPLE 1	1283
1234	negative consequences of actions that	annotation ratio = 24.5%	1284
1235	benefit their cause	GFA score = 0.7597	1285
1236	the conclusion (showing the issue is	Let 's think step by step. First, let's	1286
1237	lack of rewards	analyze the problem. The manager ar-	1287
1238	points to consider:	gues against offering free gift wrapping	1288
1239	just the lack of reward	because:	1289
1240	Cluster ID 31. This cluster (Silhouette score:	- **If most customers use it**, it €	1290
1241	0.1245) has the lowest internal coherence, likely	expensive and time-consuming.	1291
1242	due to its coverage of syntactically and se-	- **If few customers use it**, there's no	1292
1243	mantically diverse middle-stage reasoning. To-	advantage.	1293
1244	kens include "without special exhibitions,"	The conclusion is that offering it has no	1294
1245	"modernized equipment," "analogies to	benefit in either scenario.	1295
1246	hold," and "counterexamples." This group ap-		1296
1247	pears to encode domain-specific justifications and	To identify the assumption, we	1297
1248	factual grounding, functioning as semantic fillers	need a premise that bridges the gap	1298
1249	in broader logical chains.	between the two scenarios and the	1299
1250	building supplies). Thus,	conclusion. The argument implicitly	1300
1251	productivity boost could stem from	assumes that **only two outcomes	1301
1252	modernized equipment and adopted	are possible**:	1302

1303	customers use the service. This excludes	between environmental and financial	1354
1304	a middle ground (e.g., a moderate	priorities, but the argument focuses	1355
1305	number of customers) where the costs	on <i>*who*</i> should act, not <i>*why*</i> one	1356
1306	might be manageable <i>*and*</i> the service	priority outweigh s another. - **Option	1357
1307	provides a net advantage.	2** claims cost-saving benefits of	1358
1308		double hulls, but the environmentalist's	1359
1309	<i>*Analysis of options**:</i>	stance doesn't depend on long-term	1360
1310	- **Option 0**: Discusses cost of	savings only on preventing spills.	1361
1311	<i>*informing*</i> customers, which the	- **Option 3** addresses the exclusivity	1362
1312	argument never mentions. Irrelevant.	of enforcement sources, directly sup-	1363
1313	- **Option 1**: Compares costs to	porting the environmental ist's premise	1364
1314	previous seasons, but the argument	that government regulation is the only	1365
1315	focuses on uptake (most/few), not	viable alternative to industry action.	1366
1316	historical costs. Irrelevant.		1367
1317	- **Option 2**: States that either few	Thus, the environmentalist assumes no	1368
1318	or most customers would want it. This	other effective sources exist besides the	1369
1319	matches the manager's dichotomy and	industry or government.	1370
1320	is necessary to rule out a moderate		1371
1321	scenario that could justify offering	##### 3	1372
1322	the service. ** Required assumption	=====	1373
1323	**. - **Option 3**: Links slowing	SAMPLE 3	1374
1324	down shoppers to costs. While time-	annotation ratio = 39.3%	1375
1325	consuming is mentioned, the argument	GFA score = 0.8752	1376
1326	doesn't assume <i>*all*</i> slowdowns cost		
1327	money only those tied to high uptake.	Let 's think step by step. First, let's	1377
1328	Not required.	analyze the problem. The argument	1378
1329		begins by highlighting the severity	1379
1330	##### 2	of water scarcity and pollution, pre-	1380
1331	=====	dicting future conflicts. However , it	1381
1332	SAMPLE 2	shifts focus to solutions, emphasizing	1382
1333	annotation ratio = 38.9%	citizens' potential to address the crisis	1383
1334	GFA score = 0.7802	through activism (lobbying governments,	1384
1335		recycling, reducing biological footprint).	1385
1336	Let's break this down logically. The	The conclusion directly ties citizen	1386
1337	environmentalist's argument hinges on	action to mitig ating the crisis.	1387
1338	the necessity of government regulation		1388
1339	due to the industry's inaction. For this	Key points to consider:	1389
1340	reasoning to hold, the environmental	1. The argument identifies a problem	1390
1341	ist must assume that no other avenues	(water crisis) but prioritizes actionable	1391
1342	exist to enforce stricter safety stan-	solutions.	1392
1343	dards beyond the industry itself or the	2. Specific steps for citizens (lobbying,	1393
1344	government. If alternative effective	recycling) are central. 3. The conclu-	1394
1345	sources (e.g., international bodies,	sion explicitly states that active citizen	1395
1346	local governments) existed , the claim	response can mitigate the crisis.	1396
1347	that "national government regulation" is		1397
1348	required would be weakened.	Options analysis:	1398
1349		- **Option 0** (crit icize government)	1399
1350	Analyzing the options: - **Op-	is incorrect because the focus is on	1400
1351	tion 0** discusses negotiation processes,	citizen agency, not government failure.	1401
1352	which isn't directly tied to the necessity	- **Option 1** (spur activism) aligns	1402
1353	of government action.	perfectly with the call for citizens to	1403
	- **Option 1** introduces a comparison	lobby, recycle, and act.	1404

- **Option 2** (inform about consequences) is partially true but secondary; consequences set up the need for action, not the primary purpose.
- **Option 3** (promote recycling) is too narrow; recycling is one example of activism, not the main goal.

The argument's structure **problem followed by citizen-driven solutions indicates its primary** purpose is to motivate citizen activism.

1

B.2 Commonsense Reasoning (CSQA)

B.2.1 Construction of the Distilled Commonsense Reasoning Dataset (CSQA)

We construct a commonsense reasoning dataset based on CommonsenseQA (Talmor et al., 2019), where each sample is augmented with a step-by-step Chain-of-Thought explanation generated by a large language model. This dataset is designed to support training and evaluation of models with enhanced reasoning and explainability capabilities in commonsense tasks.

Specifically, we begin with the original multiple-choice questions from CommonsenseQA and employ the DeepSeek-R1 model to generate Chain-of-Thought explanations. Carefully designed prompts are used to elicit explicit multi-step reasoning paths for each question. The generated CoTs are then integrated with the original question, answer choices, and correct answer to form structured samples. Each data instance in CSQA consists of four components: the question text, answer choices, the correct answer label, and the model-generated CoT explanation.

B.2.2 Structural Stress Patterns in Commonsense Reasoning Tasks

To assess whether the proposed Gradient-based Structural Developer (GSD) generalizes beyond mathematical reasoning, we apply it to the CommonsenseQA-derived CSQA dataset. The goal is to investigate whether structurally aligned stress patterns—indicative of reasoning chains—also emerge in commonsense tasks.

We perform token-level stress extraction across CSQA samples and project the resulting stress vectors into a 2D space using t-SNE. We then apply

clustering on the 1920-dimensional token stress representations and visualize clusters with sizes ≥ 50 . As shown in Figure 8, five distinct clusters are identified, reflecting recurring structural patterns in commonsense reasoning traces.

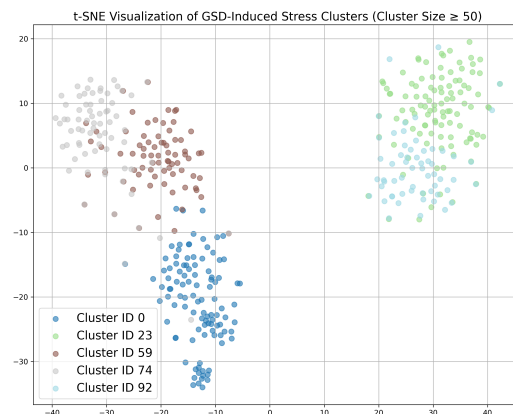


Figure 8: t-SNE visualization of GSD-induced token-level stress clusters on the CSQA dataset. Only clusters with size ≥ 50 are shown. Each point corresponds to a token stress vector.

We further evaluate clustering quality using three standard metrics:

- **Silhouette Score:** 0.3460 (higher is better), indicating moderate intra-cluster consistency.
- **Calinski-Harabasz Index:** 1258.60, showing distinct cluster separation.
- **Davies-Bouldin Index:** 1.1846 (lower is better), reflecting reasonable inter-cluster distinctiveness.

Table 2 presents the average Silhouette Score for each identified cluster:

Among these, Cluster ID 0 achieves the highest internal coherence with a silhouette score of 0.5676. This cluster corresponds to tokens frequently initiating the reasoning process (e.g., “Let’s think...” or “To solve this...”). Other clusters reflect intermediate transitions, distractor elimination, or conclusion patterns. These structured and semantically consistent stress clusters confirm the cross-domain generalizability of GSD for detecting internalized reasoning structures.

B.2.3 Qualitative Analysis of Clustered Stress Patterns

To further interpret the latent structure of GSD-induced stress clusters, we conduct a qualitative

Table 2: Average Silhouette Score per Cluster (CSQA)

Cluster ID	0	23	59	74	92
Avg. Silhouette	0.5676	0.2183	0.3209	0.2820	0.2939

1485	analysis of exemplar token spans from each major cluster in the CSQA dataset. These clusters are derived from token-level stress vectors and reflect recurring patterns in commonsense reasoning explanations.		step. First, let		1529
1486			let’s analyze the situation		1530
1487			step. First, let		1531
1488			let’s analyze the situation:		1532
1489			, let’s analyze the		1533
1490	Cluster ID 0. This cluster exhibits a remarkably consistent pattern centered around the initiation of reasoning sequences. A large proportion of samples include the canonical reasoning preamble “Let’s think step by step”, indicating that GSD successfully captures standardized rhetorical triggers for model-initiated deliberation. Many samples also begin with context-establishing phrases or problem restatements (e.g., “Sammy wants to be...”, “To share files...”, “The woman checked...”), reflecting early cognitive framing. The high silhouette score (0.5676) highlights the strong internal consistency of this cluster, which aligns with its lexical and structural regularity.		it’s implausible. Windowsill		1534
1491			this, we need to apply some everyday		1535
1492			let’s analyze the situation		1536
1493			Cluster ID 59. This cluster is characterized by tokens involved in evaluative or comparative reasoning. Examples include causal reasoning (“this aligns with...”, “bitter specifically captures...”), functional descriptions (“a crockpot is...”, “radio transmits signals...”), and pragmatic filtering (“ignore is the only option that...”). Such content typically arises in the middle or conclusion phases of explanation, reflecting the application of external knowledge to refine or eliminate distractors. The average silhouette score (0.3209) confirms moderate internal similarity.		1537
1494			by not acknowledging them. ”		1549
1495			aligns with the punitive		1550
1496			typically worn around the neck		1551
1497			with this reasoning, as		1552
1498			it the most strategic location		1553
1499			are possible, bitterness specifically		1554
1500			captures resentment		1555
1501			pet shops are the logical		1556
1502			of a national government, not a		1557
1503			business or		1558
1504	Let’s think step by step. Sammy wants to be		the president is the governing		1559
1505	is to find a choker		strain. To recover, the body and mind		1560
1506	The remaining logical option is a		need downtime		1561
1507	Let’s think step by step		Cluster ID 74. Cluster 74 shows a blend of domain-specific lexical grounding and logical conditionals. The cluster includes tokens like “choker”, “cable connections”, “cabinet”, “safe zone”, and “natural habitat”, suggesting it captures concept-defining terms or scenario-specific cues used to justify answer choices. Many of these tokens occur in statements that explain why one option is more plausible than others. Despite higher lexical diversity, the cluster maintains thematic consistency with an average silhouette score of 0.2820.		1562
1508	Let’s think step by step				1563
1509	Let’s think step by step;				1564
1510	Let’s think step by				1565
1511	Between ’shop’ (generic				1566
1512	Let’s think step by				1567
1513	’s think step by step				1568
1514	refers to the head of				1569
1515	Cluster ID 23. Cluster 23 includes a mix of planning-oriented and scenario-based reasoning patterns. Frequent use of phrases like “First, the goal is to...” and “Let’s analyze the situation...” suggests that tokens in this group often appear during the decomposition of commonsense questions into sub-goals or sub-events. Compared to Cluster 0, this group is more diverse in lexical content but still shows functional consistency in structuring step-wise explanations. The silhouette score (0.2183) indicates weaker coherence, likely due to semantic drift in intermediate steps of reasoning.				1570
1516					1571
1517					1572
1518					
1519					
1520					
1521					
1522					
1523					
1524					
1525					
1526					
1527					
1528					

1573	or public spaces. This aligns with	-1 to 1, where higher values indicate better-	1622
1574	common	defined boundaries.	1623
1575	gregate, like cities or public		
1576	by not acknowledging them.	• Calinski–Harabasz Index (Caliński and	1624
1577	Let’s think step by step.	Harabasz, 1974): Measures the ratio of	1625
1578	specifically designed to soak up	between-cluster dispersion to within-cluster	1626
1579	possible, bitterness specifically	cohesion; higher values reflect more compact	1627
1580	captures resentment	and well-separated clusters.	1628
1581	knowledge. What typically happens	• Davies–Bouldin Index (Davies and Bouldin,	1629
1582	place. The solar system	2009): Computes intra-cluster similarity ver-	1630
1583	time, it likely means he	sus inter-cluster dissimilarity, where lower val-	1631
1584	skills but doesn’t immediately	ues indicate higher-quality clustering.	1632
1585	Cluster ID 92. Cluster 92 aggregates tokens con-		
1586	tributing to nuanced or emotionally grounded inter-	These metrics confirm that the token-level clus-	1633
1587	pretations, often near the end of explanations. To-	ters induced by GSD are not only visually distinct	1634
1588	kens such as “grapes are typically...”, “resentment	but also statistically well-formed in latent space,	1635
1589	due to...”, “relaxed (C)...”, or “barbeque meat...”	offering robust evidence of reasoning-related align-	1636
1590	represent affective, environmental, or cultural con-	ment patterns.	1637
1591	text elements that influence human-centered rea-	To evaluate the structural coherence of stress	1638
1592	soning. These often involve multi-modal associa-	clusters discovered by GSD, we apply a cluster size	1639
1593	tions and pragmatic knowledge, making the cluster	threshold filter to retain only clusters containing	1640
1594	lexically varied but conceptually rich. The silhou-	more than a given number of samples. Table 4	1641
1595	ette score (0.2939) reflects this balance between	through Table 7 report the mean Silhouette Score	1642
1596	diversity and thematic focus.	of each retained cluster under thresholds $\geq 50, 55,$	1643
1597	string bag, which implies	60, and 65, respectively. A comparative summary	1644
1598	arises from resentment due to	of global clustering metrics is provided in Table 3.	1645
1599	meat on a grill aligns with “bar	We observe that as the minimum cluster size	1646
1600	involves direct heat, but	increases, the **average Silhouette Score and	1647
1601	Painting and movie hall are human-made	Calinski-Harabasz Index both improve** , while the	1648
1602	step by step. A thirsty	**Davies-Bouldin Index consistently decreases** ,	1649
1603	people hurry when they’re	indicating more compact and better-separated clus-	1650
1604	step by step: The	ters. Notably, Cluster ID 282 and 556 consistently	1651
1605	physical exercise, typically	achieve high internal consistency across all thresh-	1652
1606	associated with workouts or group	olds.	1653
1607	drills	To complement the quantitative clustering met-	1654
1608	stration is a common reaction	rics, we visualize the high-dimensional stress vec-	1655
1609		tors using t-SNE for each threshold configuration.	1656
1610	C Quantitative Evaluation of	Figures 9–12 display the clustering results for min-	1657
	Gradient-Based Alignment	imum cluster sizes $\geq 50, 55, 60,$ and 65, respec-	1658
1611	C.1 Clustering Quality of GSD Patterns	tively.	1659
1612	In addition to distributional and performance-based	As the threshold increases, less coherent clusters	1660
1613	analyses, we quantitatively assess the structural co-	are progressively filtered out, leading to improved	1661
1614	herence of token clusters revealed by the Gradient-	separation and tighter cohesion. This observation	1662
1615	based Structural Developer (GSD). We apply three	is consistent with the rising silhouette scores and	1663
1616	widely used unsupervised clustering evaluation	declining Davies-Bouldin indices, as discussed ear-	1664
1617	metrics to the t-SNE projections of stress-aligned	lier.	1665
1618	token representations:	C.2 Distributional Statistics of GFA Scores	1666
1619	• Silhouette Score (Rousseeuw, 1987): Eval-	To examine whether full supervision is necessary	1667
1620	uates how similar each token is to its own	for effective gradient alignment, we analyze the	1668
1621	cluster relative to other clusters, ranging from	GFA distribution under a pruning strategy where	1669

Table 3: Overall Clustering Metrics across Cluster Size Thresholds

Threshold	Silhouette Score	Calinski-Harabasz	Davies-Bouldin	# Samples
≥ 50	0.3546	2014.52	1.1722	939
≥ 55	0.3871	2143.91	1.0761	835
≥ 60	0.4692	6598.64	0.8243	680
≥ 65	0.4796	7074.65	0.8050	584

Table 4: Mean Silhouette Scores per Cluster (Cluster Size ≥ 50)

Cluster	12	18	175	176	177	282	293
Score	0.1183	-0.0043	0.6821	0.4012	0.2612	0.4955	0.1093
Cluster	294	479	531	556	558	697	744
Score	0.5638	0.3797	0.0687	0.8146	0.4265	0.2847	0.2338
Cluster	1189	1196	1198	1204	5588		
Score	0.4636	0.0991	0.9173	0.1396	-0.0188		

Table 5: Mean Silhouette Scores per Cluster (Cluster Size ≥ 55)

Cluster	12	18	175	176	177	282	293
Score	0.1155	0.1916	0.6759	0.3944	0.2675	0.4701	0.1067
Cluster	294	531	556	558	697	744	1196
Score	0.5646	0.3322	0.8156	0.4286	0.2841	0.2342	0.0763
Cluster	1198	5588					
Score	0.9018	0.1368					

Table 6: Mean Silhouette Scores per Cluster (Cluster Size ≥ 60)

Cluster	12	175	176	177	282	293	294
Score	0.3261	0.6694	0.5456	0.2433	0.7963	0.1057	0.5625
Cluster	531	556	558	697	744		
Score	0.7650	0.7386	0.4278	0.2951	0.2322		

Table 7: Mean Silhouette Scores per Cluster (Cluster Size ≥ 65)

Cluster	12	175	176	282	293	294	556
Score	0.5110	0.6617	0.5245	0.7883	0.1240	0.5682	0.7670
Cluster	558	697	744				
Score	0.4424	0.2808	0.2335				

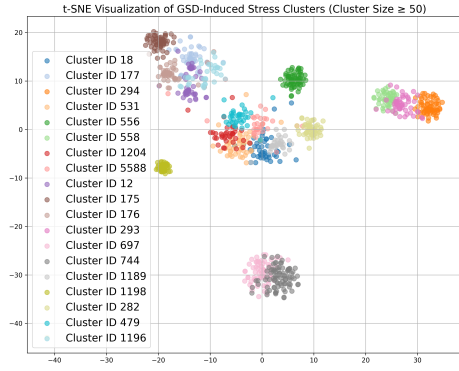


Figure 9: t-SNE visualization of stress clusters discovered by GSD with minimum cluster size ≥ 50 . A total of 19 clusters are retained, showing a wide variation in spatial separation and density. Some clusters, such as ID 556 and ID 1198, form tight and well-separated groups, while others like ID 18 and ID 5588 exhibit dispersed distributions.

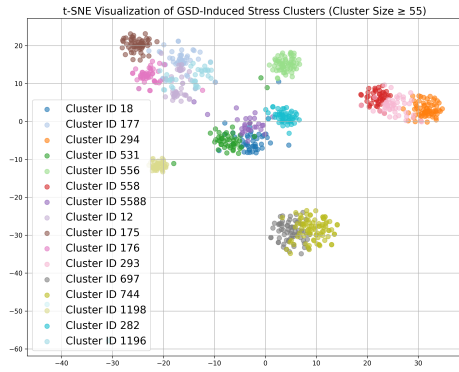


Figure 10: t-SNE visualization of stress clusters with size threshold ≥ 55 . After filtering, 16 clusters are retained. Spatial coherence improves compared to the previous setting, with visibly tighter clusters such as ID 294 and ID 1198.

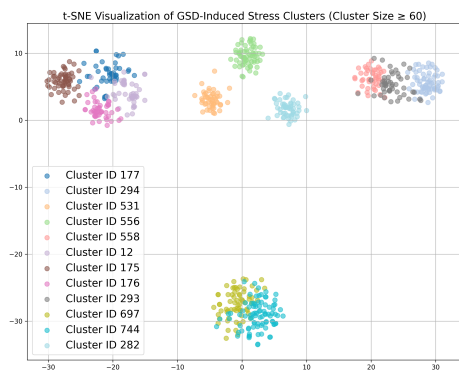


Figure 11: t-SNE visualization for clusters of size ≥ 60 . Only 12 clusters remain, demonstrating further consolidation of structure. Notably, clusters such as ID 282 and ID 556 are clearly separated, suggesting strong internal consistency.

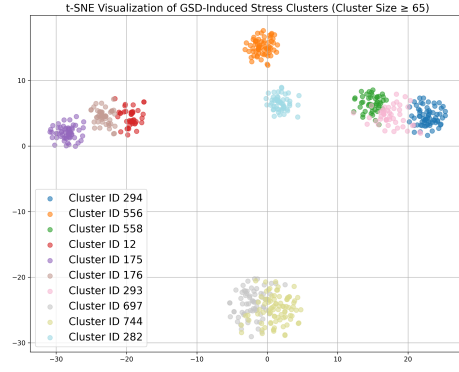


Figure 12: t-SNE plot with cluster size threshold ≥ 65 . Only 10 dense and well-separated clusters are retained. Most clusters exhibit strong internal cohesion and clear boundaries, highlighting the robustness of GSD clustering under stricter size constraints.

only tokens with importance ratio ≥ 0.1 are retained. Table 8 summarizes the mean and standard deviation of both marking ratio and GFA across varying cluster size thresholds. Despite a reduction in the average ratio from 28.1% to 21.5%, the average GFA remains high (from 0.669 to 0.653), showing minimal degradation in alignment quality.

We further examine the GFA central concentration by computing the central 90%, 80%, 70%, 60%, and 50% intervals (after trimming outliers). As shown in Table 9, the main body of GFA values remains consistently within [0.5, 0.8], regardless of pruning intensity. This indicates that alignment quality is preserved even under reduced annotation.

Finally, Table 10 explicitly compares the average marking ratio with average GFA to reveal their weak dependence. The shallow slope of this relation confirms that a higher marking ratio is not strictly required to maintain strong gradient alignment.

These results provide quantitative support for selective supervision strategies that reduce annotation cost while preserving alignment fidelity.

D Gradient Force Aggregation and Fine-Tuning Behavior

This appendix provides additional experimental details and visualizations supporting the correlation analysis reported in the main text. The purpose of this analysis is not to introduce a data selection or optimization method, but to examine whether consistency in gradient-force aggregation coincides with differences in fine-tuning behavior under extremely low-resource conditions.

Table 8: Basic statistics after pruning (Ratio ≥ 0.1)

Threshold	Samples	Mean Ratio	Std Ratio	Mean GFA	Std GFA
Size ≥ 50	103	0.2811	0.1009	0.6690	0.1612
Size ≥ 55	101	0.2565	0.1009	0.6601	0.1600
Size ≥ 60	93	0.2309	0.0978	0.6504	0.1651
Size ≥ 65	87	0.2154	0.1000	0.6526	0.1633

Table 9: GFA concentration intervals after pruning

Threshold	90% Range	80% Range	70% Range	60% Range	50% Range
Size ≥ 50	0.403–0.889	0.458–0.855	0.516–0.834	0.572–0.809	0.602–0.779
Size ≥ 55	0.398–0.881	0.430–0.855	0.492–0.822	0.563–0.800	0.597–0.761
Size ≥ 60	0.347–0.882	0.456–0.862	0.524–0.824	0.567–0.797	0.578–0.764
Size ≥ 65	0.396–0.883	0.473–0.857	0.506–0.821	0.552–0.799	0.572–0.771

Table 10: Average ratio vs. average GFA across cluster size thresholds

Threshold	Avg Marking Ratio	Avg GFA
Size ≥ 50	0.2811	0.6690
Size ≥ 55	0.2565	0.6601
Size ≥ 60	0.2309	0.6504
Size ≥ 65	0.2154	0.6526

D.1 Construction of the Target Force and Candidate Pool

We begin by constructing a reference *target force vector* by aggregating token-level gradients obtained from fine-tuning on 800 GSM8K samples for three epochs. This aggregate serves solely as a stable reference point for measuring force alignment and does not represent an optimal or oracle objective.

A candidate pool of 128 training samples is then collected from the same data distribution. For each candidate, we compute its associated gradient force using the same GSD-based extraction procedure as described in the main text.

D.2 Subset Aggregation and Alignment Measurement

To probe the relationship between force alignment and adaptation behavior, we form small training subsets of size 8, 12, and 16 samples. Starting from an empty set, samples are incrementally added following a simple greedy procedure that either increases or decreases the cosine similarity between the cumulative force of the selected subset and the

target force.

We emphasize that this procedure is intentionally simplistic and is not designed to approximate an optimal selection strategy. Instead, it provides a controlled setting in which force alignment can be varied while holding data size fixed.

D.3 Fine-Tuning Setup and Evaluation

For each selected subset, we perform lightweight supervised fine-tuning using the NLFT framework (Liu et al., 2024), which is specifically designed for extremely low-resource adaptation. All experiments are conducted with identical hyperparameters and evaluated on the GSM8K benchmark.

As summarized numerically in the main text, subsets with higher force alignment consistently achieve higher GSM8K accuracy across all tested data sizes. For completeness, Figure 13 visualizes the aggregation process, alignment evolution, and corresponding performance outcomes.

D.4 Interpretation and Scope

The results presented here should be interpreted as correlational rather than causal. They do not establish gradient-force alignment as a sufficient condition for improved generalization, nor do they propose a practical data selection method. Instead, they provide supporting evidence that internal gradient structure—revealed through GSD—coincides with measurable differences in fine-tuning outcomes, reinforcing its role as a diagnostic signal for training influence.

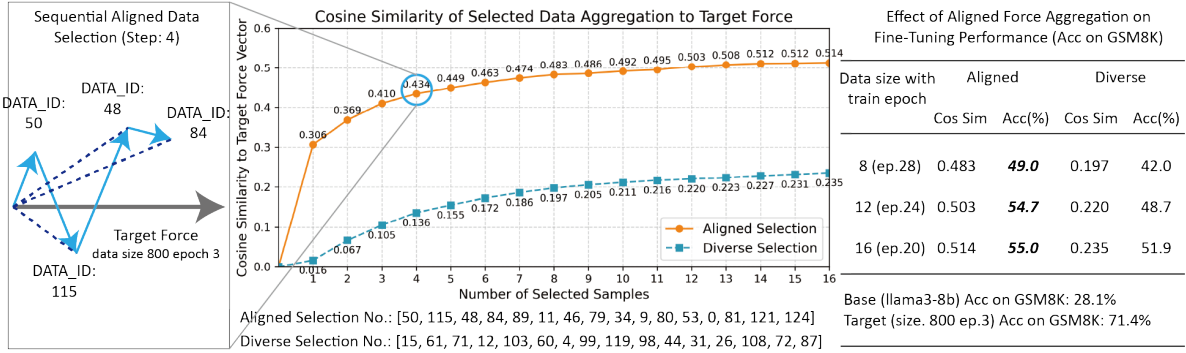


Figure 13: **Supplementary visualization of force aggregation consistency and fine-tuning behavior.** Left: illustrative depiction of sequential subset aggregation toward a reference target force. Center: cosine similarity between aggregated subset forces and the target force as a function of subset size, comparing aligned and diverse selection trajectories. Right: GSM8K accuracy obtained after fine-tuning with 8, 12, and 16 samples. These results complement the main-text analysis by visualizing how force consistency co-varies with adaptation behavior.

E Algorithmic Details

E.1 Unsupervised Clustering of Sequential Stresses

We introduce a framework named "Unsupervised Clustering of Sequential Stresses", designed for unsupervised discovery of recurring stress patterns in time-series data. The method first segments continuous stress signals into overlapping windows and computes the mean feature vector within each window. All input vectors, including 2D stress representations, are uniformly treated as 3D to maintain processing consistency. Window features are then processed in batches through an incremental clustering module: cosine similarity determines whether a window joins an existing cluster (if the similarity exceeds a threshold), or initiates a new cluster otherwise; cluster centers are updated dynamically via mean aggregation. To remove redundancy, each cluster undergoes an Intersection-over-Union (IoU)-based filtering step that selects only the most representative and non-overlapping windows. This framework does not require the number of clusters to be predefined, adaptively reveals repeating stress structures, and ensures compatibility across varying input dimensions by enforcing 3D vector consistency.

E.2 Greedy Force-Based Data Selection Algorithm

The Greedy Force-Based Data Selection Algorithm selects a subset from a pool of feature vectors to maximize representativeness while preserving diversity. It defines for each candidate f_i an attractive force toward the global center c_{global} and a repulsive force from the already selected set S . At each

iteration, the net force for candidate i is computed as

$$F_i = \alpha \cdot \text{sim}(f_i, c_{global}) - \beta \cdot \frac{1}{|S|} \sum_{j \in S} \text{sim}(f_i, f_j) \quad (34)$$

α , β are weighting parameters. The algorithm greedily picks the sample with the highest F_i into S , updates any necessary statistics, and repeats until the desired subset size or a force threshold is reached.

F Implementation Details

This section outlines the implementation protocol adopted for our gradient-based structural analysis framework, including training hyperparameters, per-token optimization mechanisms, and the LoRA configuration applied across transformer layers. While the full codebase will be made available upon publication, we describe all key implementation logic below to ensure reproducibility and clarity.

F.1 Training Hyperparameters and Infrastructure

All experiments are conducted using LLaMA-3 8B as the base model, fine-tuned on NVIDIA A100 GPUs with full 8192-token context windows. Each reasoning example is trained independently using single-sample batches to facilitate granular gradient tracing. The training pipeline leverages a distributed job launcher that concurrently dispatches multiple runs per GPU with dynamic allocation.

We use a learning rate of 5×10^{-5} and a cosine scheduler. Each training run proceeds for exactly one epoch, with no validation set. No end-of-

Input: Sequence set $\mathcal{S} = \{s_i\}_{i=1}^N$, window length range $[L_{min}, L_{max}]$, similarity threshold τ_{sim} , IoU threshold τ_{iou} , batch size B

Output: Representative non-overlapping windows \mathcal{R}

// Step 1: Compute mean feature vectors via sliding windows

Initialize list $\mathcal{W} = []$;

```

foreach sequence  $s \in \mathcal{S}$  do
  for  $w = L_{min}$  to  $L_{max}$  do
    for  $t = 1$  to  $|s| - w + 1$  do
       $\mathbf{f} \leftarrow \frac{1}{w} \sum_{j=t}^{t+w-1} s_j$ ; // treat
      2D as 3D for consistency
      Append  $(s, t, t + w, \mathbf{f})$  to  $\mathcal{W}$ ;
    end
  end
end

```

end

// Step 2: Cluster windows incrementally based on cosine similarity

Initialize clusters $\mathcal{C} = []$, centers $\mathcal{M} = []$;

```

for  $i = 1$  to  $\lceil |\mathcal{W}|/B \rceil$  do
   $\mathcal{B} \leftarrow \mathcal{W}[(i-1)B : iB]$ ;
  foreach window  $(\_, s, e, \mathbf{f}) \in \mathcal{B}$  do
    if  $\mathcal{M} \neq \emptyset$  then
      Compute similarities
       $\{\cos(\mathbf{f}, \mathbf{m}) : \mathbf{m} \in \mathcal{M}\}$ ;
    end
    if  $\max \text{similarity} \geq \tau_{sim}$  then
      Assign window to nearest
      cluster and update its center;
    else
      Create new cluster with  $\mathbf{f}$  as
      center;
    end
  end
end

```

end

// Step 3: Retain non-overlapping, most representative windows

Initialize $\mathcal{R} = []$;

```

for  $k = 1$  to  $|\mathcal{C}|$  do
  Compute cluster center  $\mathbf{c}_k$ ;
  Sort windows in  $\mathcal{C}_k$  by descending
   $\cos(\mathbf{f}, \mathbf{c}_k)$ ;
  Initialize  $\mathcal{S}_k = []$ ;
  foreach window  $(s, e)$  in sorted list do
    if  $\text{IoU}((s, e), (s', e')) < \tau_{iou}$  for all
     $(s', e') \in \mathcal{S}_k$  then
      Append  $(s, e)$  to  $\mathcal{S}_k$ ;
    end
  end
  Append  $\mathcal{S}_k$  to  $\mathcal{R}$ ;
end

```

end

return \mathcal{R}

Algorithm 1: Unsupervised Clustering of Sequential Stresses

Input: Candidate set $\mathcal{F} = \{f_1, \dots, f_N\}$, target size K , weights α, β

Output: Selected subset S

$c_{\text{global}} \leftarrow \frac{1}{N} \sum_{i=1}^N f_i$; // compute global center

$S \leftarrow \emptyset$;

for $t \leftarrow 1$ **to** K **do**

foreach $f_i \in \mathcal{F} \setminus S$ **do**

$A_i \leftarrow \text{sim}(f_i, c_{\text{global}})$;

$R_i \leftarrow |S| >$

$0 ? \frac{1}{|S|} \sum_{f_j \in S} \text{sim}(f_i, f_j) : 0$;

$F_i \leftarrow \alpha A_i - \beta R_i$;

end

$i^* \leftarrow \arg \max_{i: f_i \notin S} F_i$;

$S \leftarrow S \cup \{f_{i^*}\}$;

$c_{\text{global}} \leftarrow \frac{(t-1)c_{\text{global}} + f_{i^*}}{t}$;

 // incremental update

end

return S

Algorithm 2: Greedy Force-Based Data Selection

sequence (EOS) tokens are appended to avoid biasing token boundaries, and Alpaca-style prompts are used. 1821
1822
1823

Two hyperparameters, threshold and γ , are used to regulate token filtering and unlikelihood loss modulation. The threshold controls token inclusion based on confidence ratios, while γ scales the corrective alignment term. 1824
1825
1826
1827
1828

F.2 Token-Level Gradient Tracing Protocol 1829

To evaluate token-level contributions to LoRA modules, we implement a multi-stage backpropagation framework. For each training sample, we isolate the base branch’s hidden states and compute logits in isolation. Then, we iterate over all valid token positions (i.e., those with non-masked loss targets), and for each position, we: 1830
1831
1832
1833
1834
1835
1836

1. Compute the cross-entropy loss associated with that token. 1837
1838
2. Execute a single backward pass on that token’s loss while retaining the computation graph. 1839
1840
3. Extract gradients of the token-level loss with respect to all LoRA adapter parameters (both A and B). 1841
1842
1843
4. Compose the effective update matrix as: 1844

$$\Delta W = \frac{\alpha}{r} \cdot (\nabla B \cdot A + B \cdot \nabla A)$$
1845

1846 where A and B are the low-rank adaptation
1847 matrices, and α/r is the LoRA scaling factor.

- 1848 5. Compare the directional consistency of the
1849 computed ΔW against a reference stress
1850 alignment map, verifying whether the gradient
1851 supports or contradicts the expected optimiza-
1852 tion signal.

1853 This protocol allows fine-grained analysis of
1854 how each token affects adapter-level adaptation,
1855 enabling a structured understanding of reasoning
1856 internalization within parameter-efficient learning
1857 settings.

1858 F.3 LoRA Configuration and Optimization 1859 Settings

1860 All LoRA adapters are inserted into the MLP pro-
1861 jection layers of the transformer, specifically tar-
1862 geting the modules `gate_proj`, `down_proj`, and
1863 `up_proj`. The adapters are configured with rank
1864 $r = 16$, scaling factor $\alpha = 16$, and dropout rate
1865 0.05.

1866 To prevent numerical instability and preserve
1867 interpretability of adapter gradients, LoRA param-
1868 eters are excluded from 8-bit quantization. Instead,
1869 they are trained in full precision while other mod-
1870 ules are quantized using standard 8-bit techniques.
1871 Gradient checkpointing is enabled to reduce mem-
1872 ory usage and to support long-sequence backpropa-
1873 gation.

1874 Furthermore, we instrument all LoRA modules
1875 with internal forward hooks that explicitly track
1876 whether $A \cdot x$ operations are executed and contribute
1877 to downstream loss. This enables automatic verifi-
1878 cation of parameter usage during fine-tuning and
1879 ensures that gradients are not blocked or silently
1880 ignored during per-token analysis.

1881 G Additional Results

1882 G.1 Expanded Visualizations of Stress 1883 Clusters

1884 **Cluster Analysis Overview.** We analyze the in-
1885 ternal structure and cohesion of 21 discovered clus-
1886 ters, identified by their respective cluster IDs: 18,
1887 177, 294, 531, 556, 558, 1204, 1611, 5588, 12,
1888 175, 176, 293, 697, 744, 1189, 1198, 282, 479, and
1889 1196. Each cluster exhibits distinct gradient-level
1890 and structural characteristics based on token-level
1891 alignment patterns. Clusters such as 556, 1198,
1892 531, and 282 show strong silhouette scores, indicat-
1893 ing well-separated reasoning patterns, while others

1894 like 1611 and 12 demonstrate lower cohesion or
1895 potential misalignment. These clusters serve as
1896 the basis for our fine-grained analysis of reasoning
1897 stress localization across diverse samples.

1898 G.1.1 Strongly Separated Clusters

1899 Clusters exhibiting strong structural separation in-
1900 clude **556, 1198, 531, 175, 282**. These clusters
1901 present high silhouette scores (typically above 0.6),
1902 suggesting coherent and distinct internal reason-
1903 ing structures. The following are representative
1904 sequences sampled from each cluster:

1905 **Cluster ID 556.** student is:120 seconds -
1906 105 seconds = 15 seconds####
1907 vehicles) + \$175 (registration) =
1908 \$1825####
1909 60:6 hours x 60 = 360 minutes####
1910 they require \$49 more to buy the rare
1911 baseball card.####
1912 days of the week.####
1913 24 = \$64####
1914 42 = 28.####
1915 = 120,000 trees####
1916 24 = 2####
1917 74 questions right.####
1918 \$150 - \$140 = \$10####
1919 \$500 - \$74 = \$426####
1920 6 megabytes/picture = 4,000
1921 pictures####
1922 to spend 50 hours to complete the trip
1923 to the hotel.####
1924 + 80 + 60 = 180 days####
1925 57 pistachios with opened shells####
1926 were 15 individuals left at the zoo.####
1927 160 + 180 + 200 = 540 lbs####
1928 5.00####
1929 of juice left.####
1930 24 + 10 = 34 trash cans in Veteran's
1931 Park.####
1932 pages with text####
1933 purple flowers) + 7 (green flowers) =
1934 35####
1935 11 hours each day.####
1936 12 (birds) + 18 (fish) = 39####
1937 2 fillets/fish = 120 fillets####
1938 x 2 = 12 leaves####
1939 ow) + 10 (Cat Cafe Paw) = 40####
1940 \$110 = \$310####
1941 24 + 8 + 8 = 40 cases####
1942 apples left to eat raw.####
1943 Colton has 42 stickers left.####

1944	worth of organisms traveling together	by step:1.	1995
1945	for this walk.####	by step:1. Jaime saves \$50	1996
1946	18 years####	by step:1.	1997
1947	Ã· 20 ounces/steak = 4 steaks####	by step:1.	1998
1948	1423 more pages than Beatrix.####	by step:1.	1999
1949	of chocolates left.####	by step:1.	2000
1950	1.00.####	by step:1.	2001
1951	flower costs \$15,	by step:1.	2002
1952	divide both sides by \$15	by step:1.	2003
1953	is:20 points / 4 players = 5	by step:1.	2004
1954	points/player####	by step:1.	2005
1955	Ela's firewood = 13####	by step:1.	2006
1956	= 14####	by step:1.	2007
1957	2 years is \$132.####	by step:1.	2008
1958	25 = 10So, Purple Valley has 10	by step:1.	2009
1959	skirts.####	by step:1.	2010
1960	= 180 pages.####	by step:1.	2011
1961	is:120 fruits/tree Ã 8 trees = 960	by step:1.	2012
1962	fruits####	by step:1.	2013
1963	ane picked up 62 pounds of garbage.####	by step:1.	2014
1964	, Juan picked 61 oranges.####	by step:1.	2015
1965	540 - 104 = 436.####	by step:1.	2016
1966	360 + 90 = 450####	by step:1.	2017
1967	have 7 fish.####	by step:1.	2018
1968	5 = \$3####	:1. Bucky	2019
1969	22.5 - 10.5 = 12 centimeters####	by step:1. The	2020
1970	Area = 196 square inches####	by step:1.	2021
1971	6 + 20 + 1 = 27 slices.####	by step:1.	2022
1972	150 (part) = \$300####	by step:1.	2023
1973	960,000####		
1974	800 bricks to complete his fence.####	Cluster ID 531. last three students is 105	2024
1975	:360 drums / 6 drums per day = 60	seconds	2025
1976	days####	4 runs =	2026
1977	7 minutes (for marinara stain) = 19	much more money they require	2027
1978	minutes####	.He then brings the remaining	2028
1979	one amoeba to divide into 16	has 10 -	2029
1980	amoebae.####	keep 2 pieces of fruit	2030
1981	"TOP" books:\$104 - \$92 = \$12####	pay for the month is	2031
1982	the band has played 5 gigs.####	6 large pieces,	2032
1983	end of 2019.####	of Oreos Jordan has	2033
1984	Cluster ID 1198. by step:1.	the trip to the hotel	2034
1985	by step:1. Layla	lbs less than Jamison	2035
1986	by step:1.	12 bottles in 1	2036
1987	by step:1.	times as old as Aaron	2037
1988	by step:1.	is 15 years old	2038
1989	by step:1.	long doing accounting as calling	2039
1990	by step:1.	clients	2040
1991	by step:1.	, and 11 students	2041
1992	by step:1.	9, 10,	2042
1993	by step:1.	the average number of goals	2043
1994	by step:1.	has 400g of	2044
		so the amount of flour	2045

2046	2. Each fish gives	Let's break down the problem step by	2097
2047	many cats are there in	step:	2098
2048	30 cats in Cat Cafe	Let's break this problem down step	2099
2049	- 2000 pounds = 500 pounds	Let's break this problem down step by	2100
2050	{Mike wants to buy	step	2101
2051	8 customers bought 3 cases	Let's break this problem down step by	2102
2052	his friend Mandy	step!	2103
2053	4 stickers each to	Let's break this problem down step by	2104
2054	14 - 10 = 4 stickers	step!	2105
2055	total weight of plates per	Let's break this problem down step by	2106
2056	organisms traveling together for this	step!	2107
2057	game is \$900 per	Let's break this down step by step!	2108
2058	take his son to see	Let's break this problem down step by	2109
2059	Jane gave to the boys	step!	2110
2060	cookies for 20% more than it costs to	, we are asked to	2111
2061	make them	Let's break this problem down step by	2112
2062	as many skirts as Seafoam Valley.	step	2113
2063	67 = 40	Let's break down the problem	2114
2064	x 0.25 =	Let's break down the	2115
2065	she reads 20 pages	's break this down step by step!	2116
2066	picked up 4 times as many pounds of	Let's break down the problem	2117
2067	by step:1. Johny traveled South 40	Let's break it down step by step!	2118
2068	Del and Juan is	Let's break this down step by step!	2119
2069	.3. She then	Let's break it down step	2120
2070	by step:1.	Let's break down the problem step by	2121
2071	Each person's share x	step	2122
2072	third year, which is	Let's break down the	2123
2073	how much more money Mary has	's break this problem down step	2124
2074	add it to the height	shells - Number of shells	2125
2075	Cluster ID 175. Let's break this problem	Let's break this problem down step by	2126
2076	down step by step!	step!	2127
2077	Let's break this problem	Let's break this problem down step by	2128
2078	Let's break down the	step	2129
2079	Let's break this problem down step by	Let's break this problem down step by	2130
2080	step!	step:	2131
2081	Let's break this problem	Let's break this problem down step	2132
2082	Let's break this problem down step by	's the step-by-step	2133
2083	step.	Let's break this problem down step by	2134
2084	Let's break this problem down step by	step!	2135
2085	step!	Let's break this problem down step by	2136
2086	Let's break this problem down step by	step:1	2137
2087	step!	Let's break this problem down step by	2138
2088	hours2. They	step.	2139
2089	's break this problem down step	Let's break down the problem step by	2140
2090	Let's break down the problem step by	step:	2141
2091	step	Let's break it down step by	2142
2092	Let's break this problem down step by	Let's break this problem down step by	2143
2093	step!1	step!	2144
2094	Let's break this problem down step by	Let's break this problem down step by	2145
2095	Let's break down the	step!	2146
2096	Let's break this down step	Let's break it down step	2147
		Let's break down the	2148

2149	Let's break down the problem step by	= 120 miles North	2200
2150	step	. Lorenzo placed one thumbtack from	2201
2151	Pablo's mother agrees to pay him one	each of the	2202
2152	cent for every page he	. The three friends pay	2203
2153	Let's break down the problem step by	3 = 129	2204
2154	step:	rained 2.5 centimeters	2205
2155	Let's break down the problem step by	Mondays, we subtract the	2206
2156	step:	imeters, so the total	2207
2157	Let's break it down step	able to make 10	2208
2158	Let's break this problem down step	diaper for \$5,	2209
2159	Let's break this problem down step by	/pack = 192	2210
2160	step!Let	0 packs x 160	2211
2161	Cluster ID 282. - 10 = 14	by year:Year 1	2212
2162	analysis:Mark had 10 pieces	. Each wall is	2213
2163	20% of the	grass stains) + 7 minutes (for marinara	2214
2164	!1. Mary weighs	stain) =	2215
2165	10 trash cans were	Tommy has 3	2216
2166	Week 2) +	add 5.Tom	2217
2167	9 cakes x 100	ae every 2 days	2218
2168	2. There is a 20-minute advertisement	G.1.2 Moderately Cohesive Clusters	2219
2169	before the start of each	Clusters such as 294, 176, 744, 177 exhibit mod-	2220
2170	6 (dogs) +	erate cohesion, with silhouette scores between 0.2	2221
2171	original number by 2:	and 0.5. These clusters often encode partially	2222
2172	x 2 = 12 leaves	aligned reasoning patterns with some diversity	2223
2173	. Mr. Ray has 10 tuna, each weighing	across examples.	2224
2174	200 pounds	Cluster ID 294. bought 14 carnations.####	2225
2175	customers x 25 pounds	14[SPECIAL_< END_OF_TEXT >]	2226
2176	= 2500 pounds	.#### 20[SPECIAL_< END_OF_TEXT >]	2227
2177	customers#### Correct!	6 hours x 60 = 360 minutes####	2228
2178	\$3000Total	360[SPECIAL_< END_OF_TEXT >]	2229
2179	Colton had 72	.#### 7[SPECIAL_< END_OF_TEXT >]	2230
2180	silverware) + 24 ounces (final balance is 100% of	2231
2181	. Robie bought	the starting balance.####	2232
2182	= 4 bags	100[SPECIAL_< END_OF_TEXT >]	2233
2183	So, Robie	points: 70 - 42 = 28.####	2234
2184	. Jayden received	28[SPECIAL_< END_OF_TEXT >]	2235
2185	1.2 = 1.2	120,000[SPECIAL_< END_OF_TEXT >]	2236
2186	x 1.2 =	#### 2[SPECIAL_< END_OF_TEXT >]	2237
2187	by 60:(cost	.#### 74[SPECIAL_< END_OF_TEXT >]	2238
2188	.Since Joe sold	\$150 - \$140 = \$10####	2239
2189	(cost of making each cookie) = 1	10[SPECIAL_< END_OF_TEXT >]	2240
2190	6 lilies in	:100 x 20 = 2000####	2241
2191	. Kimberley collects 10 pounds of	2000[SPECIAL_< END_OF_TEXT >]	2242
2192	firewood	180 days####	2243
2193	, we calculate:\$120	180[SPECIAL_< END_OF_TEXT >]	2244
2194	. Jim then adds another \$28 to his	, there were 15 individuals left at the	2245
2195	portfolio	zoo.#### 15[SPECIAL_< END_OF_TEXT >]	2246
2196	10%. To find	lbs#### 540[SPECIAL_< END_OF_TEXT >]	2247
2197	= \$132So,	#### 370[SPECIAL_< END_OF_TEXT >]	2248
2198	. Daliah picked up 17.).#### 5[SPECIAL_< END_OF_TEXT >]	2249
2199	15.5 x		

2250	.00#### 1035[SPECIAL_< END_OF_TEXT >]	+ 60 + 120 = 220 miles####	2302
2251	.#### 150[SPECIAL_< END_OF_TEXT >]	220[SPECIAL_< END_OF_TEXT >]	2303
2252	180 (Henry) = 240####	now.#### 40[SPECIAL_< END_OF_TEXT >]	2304
2253	240[SPECIAL_< END_OF_TEXT >]	oranges.####	2305
2254	.#### 70[SPECIAL_< END_OF_TEXT >]	61[SPECIAL_< END_OF_TEXT >]	2306
2255	.#### 10[SPECIAL_< END_OF_TEXT >]	#### 2888[SPECIAL_< END_OF_TEXT >]	2307
2256	10 = 34 trash cans in Veteran's	540 - 104 = 436.####	2308
2257	Park.#### 34[SPECIAL_< END_OF_TEXT >]	436[SPECIAL_< END_OF_TEXT >]	2309
2258	+ 42 + 63 = 231####	after five years, Tony will have 7	2310
2259	231[SPECIAL_< END_OF_TEXT >]	fish.#### 7[SPECIAL_< END_OF_TEXT >]	2311
2260	.#### 32[SPECIAL_< END_OF_TEXT >]	centimeters####	2312
2261	92.#### 92[SPECIAL_< END_OF_TEXT >]	12[SPECIAL_< END_OF_TEXT >]	2313
2262	pages with text####	.#### 27[SPECIAL_< END_OF_TEXT >]	2314
2263	19[SPECIAL_< END_OF_TEXT >]	70 = 140 cents####	2315
2264	in his tank at the end of the heavy	140[SPECIAL_< END_OF_TEXT >]	2316
2265	rain.#### 280[SPECIAL_< END_OF_TEXT >]	#### 960,000[SPECIAL_< END_OF_TEXT >]	2317
2266	35#### 35[SPECIAL_< END_OF_TEXT >]	800 bricks to complete his fence.####	2318
2267	120 fillets####	800[SPECIAL_< END_OF_TEXT >]	2319
2268	120[SPECIAL_< END_OF_TEXT >]	books.#### 12[SPECIAL_< END_OF_TEXT >]	2320
2269	6 leaves x 2 = 12 leaves####	.#### -2[SPECIAL_< END_OF_TEXT >]	2321
2270	12[SPECIAL_< END_OF_TEXT >]	+ 3 + 11 = 17 cars.####	2322
2271	Correct!####	17[SPECIAL_< END_OF_TEXT >]	2323
2272	20[SPECIAL_< END_OF_TEXT >]	books:\$104 - \$92 = \$12####	2324
2273	#### 40[SPECIAL_< END_OF_TEXT >]	12[SPECIAL_< END_OF_TEXT >]	2325
2274	Lex will have 42 apples left to eat	So, the band has played 5 gigs.####	2326
2275	raw.#### 42[SPECIAL_< END_OF_TEXT >]	5[SPECIAL_< END_OF_TEXT >]	2327
2276) + 4 (cat) = 22####	.#### 21[SPECIAL_< END_OF_TEXT >]	2328
2277	22[SPECIAL_< END_OF_TEXT >]	Cluster ID 176. break this problem down	2329
2278	.#### 42[SPECIAL_< END_OF_TEXT >]	step by step	2330
2279	#### 5040[SPECIAL_< END_OF_TEXT >]	break this problem down step by step!	2331
2280	this walk.####	break this problem down step by step	2332
2281	12[SPECIAL_< END_OF_TEXT >]	break this problem down step by step	2333
2282	#### 1423[SPECIAL_< END_OF_TEXT >]	break down the information we	2334
2283	.#### 4[SPECIAL_< END_OF_TEXT >]	break this problem down step	2335
2284	1.00[SPECIAL_< END_OF_TEXT >]	break down the problem step	2336
2285) = 13 apples####	break this problem down step by	2337
2286	13[SPECIAL_< END_OF_TEXT >]	break this problem down step	2338
2287	- 4 = 8 fish.####	break this problem down step by step.	2339
2288	8[SPECIAL_< END_OF_TEXT >]	break this problem down step by step!	2340
2289	.#### 660[SPECIAL_< END_OF_TEXT >]	43 Oreos.	2341
2290	#### 5[SPECIAL_< END_OF_TEXT >]	break this problem down step by step	2342
2291	's firewood = 13####	break this problem down step by step	2343
2292	13[SPECIAL_< END_OF_TEXT >]	break this problem down step by	2344
2293	after 2 years is \$132.####	break this problem down step by step	2345
2294	132[SPECIAL_< END_OF_TEXT >]	break down the information we have:	2346
2295	375 pounds per square inch = 600	break this down step by	2347
2296	pounds#### 600[SPECIAL_< END_OF_TEXT >]	break this problem down step	2348
2297	= 180 pages.####	break this problem down step by step	2349
2298	180[SPECIAL_< END_OF_TEXT >]	break this problem down step by step!	2350
2299	people#### 80[SPECIAL_< END_OF_TEXT >]	this problem down step by	2351
2300	Zane picked up 62 pounds of garbage.####	break this problem down step by step!	2352
2301	62[SPECIAL_< END_OF_TEXT >]		

2353	break this down step by step!	\$2.00 =	2404
2354	this problem down step by	= \$18.00	2405
2355	problem down step by step	5.00 = \$225.00	2406
2356	this problem down step by	.00 + \$225.00 =	2407
2357	break this down step by step	\$810.006.	2408
2358	break this down step by	\$2.00	2409
2359	break this problem down step	:\$20.00 x	2410
2360	break this problem down step by step!	price by the discount percentage:	2411
2361	break this problem down step by step	\$200 - \$50 = \$150	2412
2362	this problem down step by step	Therefore, James has to pay \$150.	2413
2363	break this problem down step	.25 = \$50	2414
2364	break this problem down step by step	minus \$46 = \$54).3.	2415
2365	this problem down step by step	to save \$54.	2416
2366	break this problem down step by step:1	is \$50.2	2417
2367	break this problem down step by step.	2 = \$100, minus \$	2418
2368	break it down step by	to save \$54:	2419
2369	break this problem down step	:\$54 / \$	2420
2370	break this problem down step	%= \$20000 x 0.15	2421
2371	for every page he reads	00 + \$15000	2422
2372	problem down step by step	00 x 15%=	2423
2373	break it down step by step:	+ \$5=	2424
2374	break this problem down step by step	420= 0.05 x \$420= \$21	2425
2375	break this problem down step	\$420 - \$110	2426
2376	Cluster ID 744. + \$175 (registration) = \$	= 5% of \$420=	2427
2377	- \$51 = \$49	= \$84 + \$21 +	2428
2378	2 = \$15	= 0.20 x \$420= \$84	2429
2379	x \$156.25 = \$31.25So, she subtract	900= \$27,000	2430
2380	:20% of \$156.25 = 0.20 x \$	\$0.35 =	2431
2381	\$31.25 = \$156.25	flower = \$2160	2432
2382	ed \$31.25	:144 flowers x \$	2433
2383	31.25So	92 + \$28 = \$	2434
2384	\$31.25 to	6 = \$60.3.	2435
2385	\$125) x 100 =	x \$6 = \$36.	2436
2386	\$156.25 - \$31.25 = \$	\$202.	2437
2387	:\$125 + \$	\$135,000	2438
2388	the account.5.	Ã· 4 = \$600,000 Ã· 4 = \$	2439
2389	to the account.3	= \$150,000	2440
2390	balance, \$125.	1 - 0.10) = \$	2441
2391	9600 - \$7200 = \$2400	= \$600,0002. James splits the revenue	2442
2392	the month is:72 deliveries x \$100 per	with his	2443
2393	delivery = \$	= \$150,000 x 0.	2444
2394	= \$7200	1.2 = \$	2445
2395	\$100 per delivery =	5 = \$3	2446
2396	= \$22* Senior citizen	square foot= 56 square feet x \$40 per	2447
2397	\$9 = \$18	square foot	2448
2398	per student = \$140	foot= \$2240	2449
2399	\$100 = \$150	\$75 per hour =	2450
2400	student = \$100	is \$150.3.	2451
2401	3. The total	:\$15 + \$8 = \$23	2452
2402	\$27*	4. Add the	2453
2403	20 + \$15 =	\$23 = \$58	2454
		- \$58 = \$2	2455

2456	-gill x \$4	it down step by step	2507
2457	-gill is:2	down step by step!	2508
2458	has \$3 leftover.	down step by step!	2509
2459	.2. Pablo always	break it down step by step	2510
2460	\$5 = \$22	problem step by step!	2511
2461	\$12 = \$27.	This means that he earned	2512
2462	\$24 by 2:\$24 ÷ 2 = \$	earned a total of \$	2513
2463	" books is:4 x \$23 = \$92	Let's break it down step by step:Tommy	2514
2464	- \$92 = \$12	has	2515
2465	:13 x \$8 = \$	down step by step!	2516
2466	4 x \$20	end of 2019.	2517
2467	Cluster ID 177. this problem down step by	G.1.3 Low-Cohesion or Noisy Clusters	2518
2468	step!	Clusters such as 12, 293 demonstrate relatively	2519
2469	this problem down step by step	low silhouette scores, suggesting weaker structural	2520
2470	down step by step!	alignment or noisier optimization signals. These	2521
2471	down step by step!	clusters may contain sequences that deviate from	2522
2472	problem down step by step!We	common reasoning patterns or suffer from semantic	2523
2473	down step by step!	ambiguity.	2524
2474	this problem down step by step.	Cluster ID 12. problem step by step:	2525
2475	down step by step!	down step by step!1.	2526
2476	!Let x be the number of Oreos	down step by step!	2527
2477	number of Oreos James has	this problem down step by step:1. The	2528
2478	down step by step!	3/4:96	2529
2479	problem down step by step	down step by step!	2530
2480	parents and 2 teachers	down the problem step by step	2531
2481	down step by step!	down step by step!1.	2532
2482	on in 10 minutes	problem down step by step.	2533
2483	down step by step!	/picture = 24	2534
2484	, James has to pay	Carla has:500 ml	2535
2485	spend the money (\$50	down step by step:1.	2536
2486	problem down step by step!	down step by step!1.	2537
2487	this problem down step by step!	down step by step:1.	2538
2488	problem down step by step!	down step by step:1.	2539
2489	down step by step!	problem down step by step:1.	2540
2490	down step by step!	problem down step by step	2541
2491	down step by step!	down step by step!1.	2542
2492	break it down step by step!	break down the problem step by step:1.	2543
2493	down step by step!	The	2544
2494	Let's break it down step by step:1.	down step by step!1.	2545
2495	Robie bought	down step by step!1.	2546
2496	4 bags of chocolates left.	problem down step by step!	2547
2497	many apples Simone ate.	step by step:1	2548
2498	down step by step!	's break it down step by step:1. Robie	2549
2499	2: 90	bought	2550
2500	's journey step by step	break down the information given:	2551
2501	' expenses step by step	step by step:1. The	2552
2502	number of kangaroos	down step by step!1.	2553
2503	problem down step by step!	down step by step:	2554
2504	problem down step by step!	problem down step by step:1.	2555
2505	this problem down step by step	problem down step by step:1	2556
2506	problem down step by step.		

2557	step by step:1.	show.#### 19	2608
2558	problem down step by step	left.#### 42	2609
2559	step by step:1	= 18 years####	2610
2560	problem down step by step!1.	18[SPECIAL_< END_OF_TEXT >]	2611
2561	down step by step:1.	Beatrix.####	2612
2562	= 5 fish. However	1423[SPECIAL_< END_OF_TEXT >]	2613
2563	However, one of them	#### 1.00[SPECIAL_< END_OF_TEXT >]	2614
2564	problem down step by step:1. James	5 apples (Lauri) = 13 apples#### 13	2615
2565	sells	- 4 = 8 fish.#### 8	2616
2566	break down the problem step by step:1.	's firewood = 13#### 13	2617
2567	The three friends pay an	final portfolio value after 2 years is	2618
2568	down step by step!1.	\$132.#### 132[SPECIAL_< END_OF_TEXT >]	2619
2569	step by step:1.	0.375 pounds per square inch = 600	2620
2570	break down the problem step by step:1.	pounds#### 600	2621
2571	The paint	= 180 pages.#### 180	2622
2572	break down the problem step by step:1.	people#### 80	2623
2573	step by step:1.	fruits/tree Ã 8 trees = 960 fruits####	2624
2574	problem down step by step	960[SPECIAL_< END_OF_TEXT >]	2625
2575	Cluster ID 293. , Ariana bought 14	Zane picked up 62 pounds of garbage.####	2626
2576	carnations.#### 14	40 + 60 + 120 = 220 miles####	2627
2577	Cove.#### 20	the same number of kangaroos as Kameron	2628
2578	baseball card.####	has now.####	2629
2579	49[SPECIAL_< END_OF_TEXT >]	picked 61 oranges.#### 61	2630
2580	22 + \$18 + \$24 = \$64####	540 - 104 = 436.#### 436	2631
2581	64[SPECIAL_< END_OF_TEXT >]	7 fish.#### 7	2632
2582	: \$150 - \$140 = \$10#### 10	3 = 129####	2633
2583	amount: \$500 - \$74 = \$426####	129[SPECIAL_< END_OF_TEXT >]	2634
2584	426[SPECIAL_< END_OF_TEXT >]	she now has: 70 + 70 = 140 cents####	2635
2585	+ 7 = 43 Oreos.####	960,000#### 960	2636
2586	50 hours to complete the trip to the	, Cooper needs 800 bricks to complete	2637
2587	hotel.#### 50[SPECIAL_< END_OF_TEXT >]	his fence.#### 800	2638
2588	lbs#### 540	books.#### 12	2639
2589	120 + 100 + 150 = 370 bottles#### 370	Marco.#### -2	2640
2590	5.00#### 1035	+ 3 + 11 = 17 cars.#### 17	2641
2591	James has to pay \$150.#### 150	has played 5 gigs.#### 5	2642
2592	ister) + 180 (Henry) = 240#### 240		
2593	spent 70 minutes calling clients.####	G.2 Per-Sample Token Trace Examples	2643
2594	70	Let's break this problem down step by	2644
2595	126 + 42 + 63 = 231#### 231	step! 1. Ariana bought a bunch of 40	2645
2596	32 coconuts left after selling 10 of	flowers.	2646
2597	them.#### 32	2. 2/5 of the flowers were roses. To find	2647
2598	in is 92.####	the number of roses, we can multiply 40	2648
2599	92[SPECIAL_< END_OF_TEXT >]	by 2/5:	2649
2600	rainwater in his tank at the end of the		2650
2601	heavy rain.####	40 Ã (2/5) = 40 Ã 0.4 = 16	2651
2602	each day.####		2652
2603	x 2 = 12 leaves#### 12	So, Ariana bought 16 roses.	2653
2604	= \$420 - \$110 = \$310####	3. She also bought 10 tulips.	2654
2605	310[SPECIAL_< END_OF_TEXT >]	4. The rest of the flowers were carnations.	2655
2606	, Lex will have 42 apples left to eat	To find the number of carnations,	2656
2607	raw.#### 42	we subtract the number of roses and	2657
		tulips from the total number of flowers:	2658

2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706

40 - 16 - 10 = 14 Therefore, Ariana bought 14 carnations. #### 14

=====

Let's break this problem down step by step: 1. The average completion time of the last three students was 35 seconds, so the total time taken by the last three students is:

35 seconds/stud * 3 studs = 105 seconds

2. The average completion time of all four runners was 30 seconds, so the total time taken by all four runners is:

30 seconds/run * 4 runs = 120 seconds

3. Since the total time taken by the last three students is 105 seconds, the time taken by the first student is: 120 seconds - 105 seconds = 15 seconds #### 15

=====

Let's break down the cost of the dirt bikes and off-road vehicles:

Dirt bikes: 3 x \$150 = \$450
Off-road vehicles: 4 x \$300 = \$1200
Total cost of vehicles : \$450 + \$1200 = \$1650
Registration cost: 3 (dirt bikes) + 4 (off-road vehicles) = 7 x \$25 = \$175
Total cost: \$1650 (vehicles) + \$175 (registration) = \$1825 #### 1825

=====

Let's break this problem down step by step! Let's start by identifying the given information:

It takes 8 hours to clean the room. A quarter of the time spent cleaning was by Lilly. Fiona was responsible for the rest of the cleaning.

Since Lilly cleaned for a quarter of the time, that means Lilly cleaned for:

8 hours x (1/4) = 2 hours

This means that Fiona cleaned for the remaining time, which is:

8 hours - 2 hours = 6 hours

Since we need to convert the time from hours to minutes, we multiply the number of hours by 60: 6 hours x 60 = 360 minutes #### 360

=====

Let's break down the information we have :

Patricia has \$6 * Lisa has five times Patricia's money, which is 5 x \$ 6 = \$30 * Lisa has double Charlotte's money, so Charlotte has \$30 ÷ 2 = \$15 * The total amount they have is \$6 + \$30 + \$15 = \$51 * The rare baseball card costs \$100

To find out how much more money they require, we subtract the amount they have from the cost of the card:

\$100 - \$51 = \$49 Therefore , they require \$49 more to buy the rare baseball card. #### 49

=====

Here's the corrected analysis: Mark had 10 pieces of fruit initially.

On Friday, he decided to keep 2 pieces of fruit for next week, which means he has 10 - 2 = 8 pieces of fruit left . He then brings the remaining 3 pieces of fruit to school for the day, which means he has 8 - 3 = 5 pieces of fruit left.

The key point is that Mark ate the 3 pieces of fruit he brought to school on Friday. So, we need to subtract the number of pieces he ate on Friday from the total number of pieces he had

2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755

2756 initially:
 2757
 2758 10 (initial pieces) - 3 (pieces eaten
 2759 on Friday) = 7
 2760
 2761 So, Mark ate 7 **pieces of fruit in**
 2762 **the first four days of the week. ##### 7**

=====

2764 Let's **break down the problem step by**
 2765 **step:**

- 2766 1. Megan's grandma gave her \$125 to
- 2767 start a savings account.
- 2768 2. She increased the account by 25%
- 2769 from funds she earned babysitting. To
- 2770 find the amount she added, we can
- 2771 calculate 25% of the starting balance:

2772
 2773
 2774 25% of \$125 = 0.25 x \$125 = \$
 2775 **31.25** So, she added **\$31.25 to the**
 2776 **account. 3.** The new balance is : **\$125 +**
 2777 **\$31.25 = \$156.25 4** . Then, it decreased
 2778 by 20% when she bought a new pair of
 2779 shoes. To find the amount she subtracted,
 2780 we can calculate **20% of the new**
 2781 **balance : 20% of \$156.25 = 0.20 x**
 2782 **\$156.25 = \$31.25 So, she subtracted**
 2783 **\$31.25 from the account. 5. The final**
 2784 **balance is:**

2785
 2786 **\$156.25 - \$31.25 = \$125**

2787
 2788 The final balance is the same as
 2789 the starting **balance, \$125. To** find
 2790 the percentage of the starting balance,
 2791 we can divide the final balance by the
 2792 starting balance and multiply by 100:

2793
 2794 $(\$125 \div \$125) \times 100 = 100 \%$

2795
 2796 So, the **final balance is 100% of**
 2797 **the starting balance. ##### 100**

=====

2799 {Oula and Tona work for a delivery
 2800 service company, **supplying different**
 2801 **parts of their state with milk products.**
 2802 **They are paid \$100 for each delivery**
 2803 **made in a day** . In a particular month,
 2804 Oula made 96 deliveries while Tona

2805 made 3/4 times as many deliveries as
 2806 Oula. Calculate their difference in pay
 2807 that month.}

2808
 2809 Oula made 96 deliveries, and each
 2810 delivery is paid \$100. So, Oula's total
 2811 pay for the month is:

2812
 2813 96 deliveries x **\$100 per delivery**
 2814 **= \$ 9600**

2815
 2816 Tona made 3/4 times as many de-
 2817 liveries as Oula. To find Tona's number
 2818 of deliveries, multiply **O ula's number**
 2819 **of deliveries by 3/4: 96 deliveries x 3/4**
 2820 **= 72 deliveries**

2821
 2822 Tona's total **pay for the month is:**
 2823 **72 deliveries x \$100 per delivery =**
 2824 **\$7200 To** find the difference in pay
 2825 between Oula and Tona, subtract Tona's
 2826 pay from Oula's pay:

2827
 2828 **\$9600 - \$7200 = \$2400 ##### 2400**

=====

2829
 2830 Let's break **down the problem step by**
 2831 **step:** 1. The area of the street is given
 2832 by the length of one side squared, which
 2833 is 100 meters squared (100^2 = 10,000
 2834 square meters).

2835 2. The area of the forest is three times
 2836 the area of the street, so it is 3 x 10,000
 2837 = 30,000 square meters.

2838 3. The number of trees per square meter
 2839 in the forest is 4, so the total number of
 2840 trees in the forest is the area of the forest
 2841 multiplied by the number of trees per
 2842 square meter:

2843
 2844 Total number of trees = Area of
 2845 forest x Trees per square meter
 2846 = 30,000 square meters x 4 trees/square
 2847 meter
 2848 **= 120,000 trees ##### 120,000**

=====

2849
 2850 Let's break down the problem step by
 2851 step:

- 2852 1. The class raised \$50 initially.
- 2853

2854 2. Each of the 20 students contributed
 2855 \$5 each, so the total amount contributed
 2856 by the students is:
 2857 20 students x **\$5 per student = \$100**
 2858 **3. The total amount** in the class fund
 2859 is the initial amount raised (\$50) plus
 2860 the amount contributed by the students
 2861 (\$100):
 2862 **\$50 + \$100 = \$150** 4. The cost of the
 2863 trip is \$7 per student, and there are 20
 2864 students. So, the total cost of the trip is:
 2865 20 students x \$7 **per student = \$140** 5
 2866 . After paying for the trip, **the amount**
 2867 **left in the class** fund is the total amount
 2868 in the fund minus the cost of the trip :
 2869 **\$150 - \$140 = \$10 ##### 10**

2870 =====

2871 Let's break down the cost of each item:
 2872
 2873 6 kilos of potatoes for **\$2 per kilo: 6**
 2874 **x \$2 = \$12 * 9 kilos of tomatoes for**
 2875 **\$3 per kilo: 9 x \$3 = \$27 * 5 kil** os of
 2876 cucumbers for **\$4 per kilo: 5 x \$4 = \$20**
 2877 *** 3 kilos of bananas for \$5 per kilo: 3 x**
 2878 **\$5 = \$15**

2880 Total cost: \$12 + **\$27 + \$20 +**
 2881 **\$15 = \$74**

2883 Peter started with \$500, so he spent \$74.
 2884 To find his remaining money, subtract
 2885 the cost from the initial **amount: \$500 -**
 2886 **\$74 = \$426 ##### 426**

2887 =====

2888 **Let's break this problem down step by**
 2889 **step! Let x be the number of Oreos**
 2890 **Jordan** has.

2892 Since James has 7 more than 4
 2893 times the number **of Oreos Jordan has,**
 2894 James has $4x + 7$ Oreos.

2896 We know that the total number of
 2897 Oreos is 52, so we can set up the
 2898 equation:

2900 $x + (4x + 7) = 52$

2902 Combine like terms:

$5x + 7 = 52$

Subtract 7 from both sides:

$5x = 45$

Divide both sides by 5:

$x = 9$

So Jordan has 9 Oreos.

Now, we can find the **number of**
Oreos James has: James has $4x + 7 =$
 $4(9) + 7 = 36 + 7 = 43$ **Oreos. ##### 43**

=====

Let's break down the problem step by
 step:

1. Pearl wants to order stockings
 for 5 grandchildren and 4 children,
 which is a total of $5 + 4 = 9$ children.

2. She wants to order 5 monogrammed
 Christmas stockings for each child,
 which is a total of $9 \times 5 = 45$ stockings.

3. The stockings are currently 10% off,
 which means the discount is 10% of the
 original price of \$20.00 each. To find
 the discount, multiply the original **price**
by the discount percentage: \$20.00 x 0
 $.10 = \$2.00$ 4 . The discounted price of
 each stocking is the original price minus
 the discount:

\$20. 00 - \$2.00 = \$18.00 5. **The**
total cost of the stockings before
monogramming is the number of
 stockings multiplied by the discounted
 price:

45 stockings x \$18.00 = **\$810.00** 6.
The monogramming cost is \$5.00 per
 stocking, so the total monogramming
 cost is:

45 stockings x **\$5.00 = \$225.00** 7. The
 total cost of the stockings, including
 monogramming, is the total cost of the
 stockings before monogramming plus
 the monogramming cost:

$\$810.00 + \$225.00 = \$1035.00$ **####**
1035

2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000

=====
Let's break this down step by step!
The adoption fee is \$200, and James' friend agrees to pay 25% of that. To find 25% of \$200, we can multiply \$200 by 0.25 (since 25% is equal to 0.25 as a decimal).
$$\$200 \times 0.25 = \$50$$
 So , James' friend will pay \$50. To find out how much James has to pay, we subtract the amount his friend is paying from the total adoption fee : $\$200 - \$50 = \$150$
Therefore, James has to pay \$150. #### 150
=====
Let's start by finding the age of Henry's sister. Since she is three **times as old as Aaron**, who **is 15 years old**, we can multiply Aaron's age by 3:
$$15 \times 3 = 45$$

So, Henry 's **sister is 45 years old**. Next, we can find Henry's age by multiplying his sister's age **by 4**: $45 \times 4 = 180$
So, Henry is 180 years old.
Now, we can find the combined **age of the siblings by adding** their ages:
$$15 \text{ (Aaron)} + 45 \text{ (sister)} + 180 \text{ (Henry)} = 240$$
 #### 240
=====
Let's break this problem down step by step! 1. Luke catches 2 fish every day for 30 days, so the total number of fish he catches is:
$$2 \text{ fish/day} \times 30 \text{ days} = 2 \times 30 = 60 \text{ fish}$$

2. Each fish gives him 2 fillets, so the total number of fillets he gets from each fish is:

2 fillets/fish 3001
3002
3. Since Luke catches 60 fish, the total number of fillets he gets is: 3003
3004
3005
$$60 \text{ fish} \times 2 \text{ fillets/fish} = 120 \text{ fillets}$$
 #### 120 3006
3007
=====
3008
Let's break down the problem step by step: 3009
3010
3011
1. Anderson adds 3 sprigs of mint to every batch of mud. 3012
3013
2. For every sprig of mint, he adds 2 green tea leaves. 3014
3015
3. So, the total number of green tea leaves added to a batch of mud is 3 sprigs x 2 leaves/sprig = 6 leaves. 3016
3017
4. Since the new mud makes the ingredients half as effective, Anderson needs to add twice as many green tea leaves to get the same efficacy as before. 3018
3019
3020
3021
3022
3023
To find the new number of green tea leaves, **we can multiply the original number by 2**: $6 \text{ leaves} \times 2 = 12 \text{ leaves}$ **#### 12** 3024
3025
3026
3027
=====
3028
Let's break down the problem step by step: 3029
3030
3031
1. Vikki's weekly earnings are calculated by multiplying her hourly pay rate by the number of hours she worked: 3032
3033
3034
3035
Weekly earnings = Hourly pay rate x Number of hours 3036
= \$10 x 42 3037
= \$420 3038
3039
2. 20% of the weekly earnings is **deducted as tax: Tax = 20 % of \$420 = 0.20 x \$420 = \$84** 3040
3041
3042
3043
3044
3045
Insurance cover = **5% of \$420 = 0.05 x \$420 = \$21** 3046
3047
3048
3049
\$5 is deducted for union dues: 3049

3050 Union dues = \$5
 3051
 3052 5. To find the total amount de-
 3053 ducted, add the tax, insurance cover, and
 3054 union dues:
 3055
 3056 Total deductions = Tax + Insur-
 3057 ance cover + Union dues
 3058 = \$84 + \$21 + \$5 = \$110 6 . To find
 3059 the amount Vikki takes home, subtract
 3060 the total deductions from her weekly
 3061 earnings:
 3062
 3063 Amount taken home = Weekly earnings -
 3064 Total deductions
 3065 = \$420 - \$110 = \$310 ##### 310
 3066
 =====
 3067 Let's start by counting the number
 3068 of legs of each member of Javier's
 3069 household:
 3070
 3071 Javier: 2 legs
 3072 Wife: 2 legs
 3073 Children (3): 3 x 2 = 6 legs
 3074 Dogs (2): 2 x 4 = 8 legs
 3075 Cat: 4 legs
 3076
 3077 Now, **let's add up the total** num-
 3078 ber of legs:
 3079
 3080 2 (Javier) + 2 (wife) + 6 (children)
 3081 + 8 (dogs) + 4 (cat) = 22 ##### 22
 3082
 =====
 3083 Let 's break this down step by step!
 3084 Colton had 72 dolphin stickers initially.
 3085
 3086 He gave 4 stickers each to 3 friends, so
 3087 he gave a total of 4 x 3 = 12 stickers to
 3088 his three friends.
 3089
 3090 He gave his friend Mandy 2 more than
 3091 he gave his three friends, so he gave
 3092 Mandy 12 + 2 = 14 stickers.
 3093
 3094 He gave Justin 10 less than Mandy, so
 3095 he gave Justin 14 - 10 = 4 stickers. In
 3096 total, Colton gave away 12 + 14 + 4 = 30
 3097 stickers.
 3098

To find out how many stickers Colton
 has left, we subtract the number of
 stickers he gave away from the initial
 number of stickers:
 72 - 30 = 42
 Colton has 42 stickers left. #####
 42
 =====
 Let's break it down step by step: 1.
 Robie bought 3 bags of chocolates ini-
 tially.
 2. She gave 2 bags to her brothers and
 sisters, leaving her with:
 3 - 2 = 1 bag
 3. Then, she bought another 3 bags of
 chocolates.
 4. Now, she has the 1 bag she had left
 initially, plus the 3 new bags:
 1 + 3 = 4 bags So, Robie has 4 bags of
 chocolates left. ##### 4
 =====
 Let's break down the problem step by
 step: 1. The cost of the magazine is
 \$0.85.
 2. The cost of the pencil is \$0.50.
 3. The total cost before the coupon is
 \$0.85 + \$0.50 = \$ 1.35.
 4. The coupon gives Daniel \$0.35 off,
 which means he gets a discount of \$0.35.
 5. To find the total amount Daniel spends,
 we subtract the discount from the total
 cost: \$1.35 - \$0.35 = \$1.00. ##### 1.00
 =====
 Let's break down the information
 given: * Ollie caught 5 fish.
 Angus caught 7 more fish than Ollie, so
 Angus caught 5 + 7 = 12 fish.
 Angus caught 4 more fish than Patrick,
 so Patrick caught 12 - 4 = 8 fish. ##### 8
 =====
 Let's start by using the information that
 Joe sells cookies for 20% more than it
 costs to make them. This means that
 the selling price of each cookie is 1.2

3145 times the cost of making each cookie .
 3146 **Since Joe sold 50** cookies and earned
 3147 \$60, the total selling price is \$60. We
 3148 can set up an equation to represent this:
 3149
 3150 $50 \times (\text{cost of making each cookie})$
 3151 $\times 1.2 = 60$
 3152
 3153 To solve for the cost of making
 3154 each cookie, we can divide both sides of
 3155 **the equation by 60: (cost of making**
 3156 **each cookie) $\times 1.2 = 60 / 50$**
 3157 **(cost of making each cookie) $\times 1.2 = 1.2$**
 3158 Now, we can divide both sides of the
 3159 equation by 1.2 to solve for the cost of
 3160 making each cookie:
 3161
 3162 $(\text{cost of making each cookie}) =$
 3163 $1.2 / 1.2$
 3164 **(cost of making each cookie) = 1 So ,**
 3165 the cookies cost \$1 to make.

3166
 =====
 3167 Let 's **break this problem down step**
 3168 **by step: 1. The team** scored a total of
 3169 270 points.
 3170 2. 5 players averaged 50 points each, so
 3171 the total points scored by these 5 players
 3172 is:
 3173
 3174 $5 \text{ players} \times 50 \text{ points/player} = 250$
 3175 points
 3176
 3177 3. Since the total points scored by
 3178 the team **is 270, the remaining** points
 3179 scored by the 4 other players ($270 - 250 = 20$)
 3180 is:
 3181
 3182 $270 - 250 = 20 \text{ points}$
 3183
 3184 4. Since there are 4 players, the
 3185 average points scored by each of these
 3186 players **is: 20 points / 4 players = 5**
 3187 **points/player ##### 5**

3188
 =====
 3189 Let's break down the problem step by
 3190 step:
 3191
 3192 1. Jim starts with \$80 in his in-
 3193 vestment portfolio.

3194 2. After 1 year, the portfolio grows by
 3195 15%. To find the new value, we calculate:
 3196
 3197 $\$80 \times (1 + 0.15) = \$80 \times 1.15 =$
 3198 $\$92$
 3199
 3200 So, the portfolio is now worth \$92.
 3201 3 . **Jim then adds another \$28 to his**
 3202 **portfolio, making the new total:**
 3203
 3204 $\$92 + \$28 = \$120$
 3205
 3206 4. After 1 more year, the com-
 3207 bined portfolio grows by **10%. To find**
 3208 **the new value , we calculate: $\$120 \times (1$**
 3209 **$+ 0.10) = \$120 \times 1.10 = \132 So, the**
 3210 **final portfolio value after 2 years is**
 3211 **$\$132. ##### 132$**

3212
 =====
 3213 Let's start by finding the area of the
 3214 sandbox. Since it's a **square, we can**
 3215 **use the formula: Area = side \times side**
 3216 $= 40 \text{ inches} \times 40 \text{ inches}$
 3217 $= 1600 \text{ square inches}$
 3218
 3219 Since a 30-pound bag of sand is
 3220 enough to fill 80 square inches, we can
 3221 find the amount of sand needed to fill 1
 3222 square inch:
 3223
 3224 $30 \text{ pounds} \div 80 \text{ square inches} =$
 3225 $0.375 \text{ pounds per square inch}$
 3226
 3227 Now, we can find the total amount
 3228 of sand needed to fill the entire sandbox:
 3229
 3230 $1600 \text{ square inches} \times 0.375 \text{ pounds}$
 3231 **per square inch = 600 pounds #####**
 3232 **600**

3233
 =====
 3234 Let's break this problem down step by
 3235 step! 1. Azure Valley has 60 skirts.
 3236 2. Seafoam Valley has two-thirds as
 3237 many skirts as Azure Valley. To find
 3238 the number of skirts in Seafoam Valley,
 3239 we can multiply the number of skirts in
 3240 Azure Valley by two-thirds:
 3241
 3242 $60 \times (2/3) = 60 \times 0.67 = 40$ So ,

3243
3244
3245
3246
3247
3248
3249
3250
3251
3252

3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280

3281

3282
3283
3284
3285
3286
3287
3288
3289
3290
3291

Seafoam Valley has 40 skirts.
3. Purple Valley has one-quarter **as many skirts as Seafoam Valley**. To find the number of skirts in Purple Valley, we can multiply the number of skirts in Seafoam Valley by one-quarter:

$$40 \times (1/4) = 40 \times 0.25 = 10$$

So, Purple Valley has 10 skirts. ##### 10

=====

Let's break this problem down step by step: Initially, each car has 3 people (driver and 2 passengers). There are 20 cars, so the total number of people is:

3 people/car \times 20 cars = **60 people** When the cars reach the halfway point, each car gains another passenger. This means each car now has 4 people (driver and 3 passengers). The total number of people remains the same, but the distribution **changes: 60 people (initially) = 60 people (still)**

However, the number of people **in each car changes: 3 people/car** \times 20 cars = 60 people (initially)
4 people/car \times 20 cars = 80 people (after gaining another passenger)

Since the race continues, and every single car has to pass **the finish line, the number of people** in each car remains the same (4 people/car). Therefore, the total number of people in the cars by the end of the race is:

80 people ##### 80

=====

Let's break this problem down step by step: 1. Daliah picked up 17.5 pounds of garbage.
2. Dewei picked up 2 pounds less than Daliah, so Dewei picked up $17.5 - 2 = 15.5$ pounds of garbage.
3. Zane **picked up 4 times as many pounds of garbage** as Dewei. To find out how many pounds Zane picked up, we multiply Dewei's amount by 4:

$$15.5 \times 4 = 62 \text{ pounds}$$

So, **Zane picked up 62 pounds of garbage. ##### 62**

=====

Let's break this problem down step by step: 1. Daliah picked up 17.5 pounds of garbage.
2. Dewei picked up 2 pounds less than Daliah, so Dewei picked up $17.5 - 2 = 15.5$ pounds of garbage.
3. Zane **picked up 4 times as many pounds of garbage** as Dewei. To find out how many pounds Zane picked up, we multiply Dewei's amount by 4:

$$15.5 \times 4 = 62 \text{ pounds}$$

So, **Zane picked up 62 pounds of garbage. ##### 62**

=====

Let's break this problem down step by step! First, we know that Del picked a total of 23 oranges per day for 2 days, so we can calculate the total number of oranges Del picked:

$$23 \text{ oranges/day} \times 2 \text{ days} = 46 \text{ oranges}$$

Now, we know that the total number of oranges picked by **Del and Juan is 107**. Since Del picked 46 oranges, we can subtract that from the total to find out how many oranges Juan picked:

$$107 \text{ oranges} - 46 \text{ oranges} = 61 \text{ oranges}$$

Therefore, **Juan picked 61 oranges. ##### 61**

=====

Let's break down Robert's earnings **step by step: 1. Basic salary: \$1250 per month 2. Commission: 10% of total sales = 0.10** \times \$23600 = \$2360
3. Total earnings: Basic salary +

3292
3293
3294
3295
3296

3297

3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312

3313

3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333

3334

3335
3336
3337
3338
3339

3340 Commission = $\$1250 + \$2360 = \$3610$
 3341
 3342 Now, Robert allocates 20% of his
 3343 total earnings to savings:
 3344
 3345 4. Savings: 20% of $\$3610 = 0.20$
 3346 $\times \$3610 = \722
 3347
 3348 The rest of the money goes to his
 3349 monthly expenses:
 3350
 3351 5. Monthly expenses: Total earn-
 3352 ings - Savings = $\$3610 - \$722 = \$2888$
 3353 ##### 2888

3354 =====

3355 **Let's break this problem down step by**
 3356 **step. Initially** , Tony has 2 fish.

3358 In the first year, his parents buy
 3359 him 2 more fish, making a total of $2 +$
 3360 $2 = 4$ fish. However, one **of them dies,**
 3361 **leaving him** with $4 - 1 = 3$ fish.

3363 In the second year, his parents buy
 3364 him 2 more fish, making a total of $3 + 2$
 3365 **= 5 fish. However, one of them dies** ,
 3366 leaving him with $5 - 1 = 4$ fish.

3368 This pattern continues for the next
 3369 three years:

3371 Year 3: $4 + 2 = 6$ fish, $6 - 1 = 5$
 3372 fish
 3373 Year 4: $5 + 2 = 7$ fish, $7 - 1 = 6$ fish
 3374 Year 5: $6 + 2 = 8$ fish, $8 - 1 = 7$ fish

3376 So, **after five years, Tony will**
 3377 **have 7 fish. ##### 7**

3378 =====

3379 Here's the correct analysis with the
 3380 chain of thought:

3382 ### Question: James sells a \$500,000
 3383 house for 20% over market value. He
 3384 splits the revenue with his 3 brothers.
 3385 How much does each person get after
 3386 taxes take away 10%?

3388 Let's break this **problem down**

step by step: 1. James sells the house 3389
 for 20% over market value. To find the 3390
 selling price, we multiply the market 3391
 value by 1.2 (since 20% is the same as 3392
 0.20, and $1 + 0.20 = 1.20$). 3393
 3394

Selling price = Market value $\times 1.2$ 3395
 $= \$500,000 \times 1.2 = \$600,000$ **2. James** 3396
splits the revenue with his 3 brothers. 3397
 Since there are 4 people (James and his 3398
 3 brothers), they will split the revenue 3399
 into 4 equal parts. 3400

Each person's share = Total rev- 3402
 enue $\tilde{A} \cdot 4 =$ Selling price $\tilde{A} \cdot 4 =$ 3403
 $\$600,000 \tilde{A} \cdot 4 = \$150,000$ **3. After taxes** 3404
 take away 10%, each person's share is 3405
 reduced by 10%. 3406

Each person's share after taxes = 3408
Each person's share $\times (1 - 0.10) =$ 3409
 $\$150,000 \times (1 - 0.10) = \$150,000 \times 0.90$ 3410
 $= \$135,000$ ##### 135,000 3411

3412 =====

Let's break down the problem step 3413
by step: 1. The three friends pay an 3414
equal amount of money, which is \$5 3415
 each. 3416

2. Since they are paying an equal 3417
 amount, the total amount they pay is $3 \times$ 3418
 $\$5 = \15 . 3419

3. This total amount is for 5 bags of 3420
 chips, so to find the cost of each bag, 3421
 we can divide the total amount by the 3422
 number of bags: 3423

$\$15 \tilde{A} \cdot 5 = \3 ##### 3 3424
 3425

3426 =====

Let's break it down step by step: 3427
 On each of 7 Mondays, it rained 1.5 3428
 centimeters, so the total amount **of** 3429
rain on Mondays is: $7 \times 1.5 = 10.5$ 3430
 centimeters 3431
 3432

On each of 9 Tuesdays, it **rained** 3433
2.5 centimeters, so the total amount of 3434
 rain on Tuesdays is: 3435

$9 \times 2.5 = 22.5$ centimeters 3436
 3437

3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485

To find out how many more centimeters it rained on Tuesdays than **Mondays, we subtract the amount** of rain on Mondays from the amount of rain on Tuesdays:

22.5 - 10.5 = 12 centimeters ##### 12

=====

Let's break it down step by step: Lizzy's mother gave her 80 cents, and her father gave her 40 cents, so she has a total of:

80 + 40 = 120 cents

She spent 50 cents buying candy for her little **cousin, so she has:** 120 - 50 = 70 cents

Her uncle gave her another 70 cents, so **she now has: 70 + 70 = 140 cents ##### 140**

=====

Let's break down the problem step by step:

1. The mechanic charges \$75 per hour, and the work takes 2 hours, so the labor cost is:

2 hours x **\$75 per hour = \$ 150**

2. The part cost **is \$150.** **3. To find** the total cost, we add the labor cost and the part cost:

\$150 (labor) + **\$150 (part) = \$300 ##### 300**

=====

Let's break down the problem step by step:

1. Meadow orders 30 boxes of diapers weekly, with each box containing 40 packs. So, the total number of packs is:

30 boxes x 40 packs/box = 120 **0 packs 2. Each pack contains 160 diapers,** so the total number of diapers is:

120 **0 packs x 160 diapers/pack = 192,000** diapers

3. Meadow sells each **diaper for \$5, so** the total revenue is:

192,000 diapers x \$5/diaper = \$ **960,000 ##### 960,000**

=====

Let's break down the problem step by step: 1 . Each wall is 20 bricks long, 5 bricks high, and 2 bricks deep. To find the total number of bricks in each wall, we multiply the length, height, and depth:

20 bricks (length) Ã 5 bricks (height) Ã 2 bricks (depth) = 200 bricks

So, each wall requires 200 bricks.

2. Cooper needs to build a fence around all four sides of his property, which means he needs to build four walls. To find the total number of bricks needed, we multiply the number of bricks in each wall by the number of walls:

200 bricks (per wall) Ã 4 walls = 800 bricks

Therefore , **Cooper needs 800 bricks to complete his fence. ##### 800**

=====

Let's break down the problem step by step:

1. Bucky earned \$35 last week-end, so he already has \$35.

2. This Sunday, he caught 5 fish, and 60% of them were trout. To find the number of trout, multiply 5 by 0.6:

3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534

3535 $5 \times 0.6 = 3$ trout
 3536
 3537 The remaining fish are blue-gill:
 3538
 3539 $5 - 3 = 2$ blue-gill
 3540
 3541 3 trout and 2 blue-gill means he
 3542 caught a total of 5 fish.
 3543
 3544 3. Trout earn \$5 each, so the total
 3545 amount earned from trout is:
 3546
 3547 $3 \text{ trout} \times \$5 = \15
 3548
 3549 2 blue-gill earn \$4 each, so the to-
 3550 tal amount earned from blue **-gill is: 2**
 3551 **blue-gill x \$4 = \$8**
 3552
 3553 The total amount earned this Sun-
 3554 day is : **\$15 + \$8 = \$23 4. Add the**
 3555 **amount** earned this Sunday to the
 3556 amount earned last weekend:
 3557
 3558 $\$35 + \$23 = \$58$ 5. Bucky needs
 3559 \$60 to buy the game, so he still needs:
 3560
 3561 $\$60 - \$58 = \$2$ ##### 2
 3562
 =====
 3563 Here's the correct solution:
 3564
 3565 Let's break this **problem down**
 3566 **step by step: 1. Marco** has \$24 and
 3567 gives Mary half of what he has. To find
 3568 out how much he gives Mary, we can
 3569 divide **\$24 by 2: $\$24 \div 2 = \12**
 3570
 3571 So Marco gives **Mary \$12. 2.** Mary now
 3572 has $\$15 + \$12 = \$27$. **3 . Mary spends**
 3573 **\$5, so** we subtract \$5 from \$27:
 3574
 3575 $\$27 - \$5 = \$22$ 4. Marco still has
 3576 \$24, and Mary now has \$22. To find
 3577 out **how much more money Mary**
 3578 **has than** Marco, we subtract Marco's
 3579 amount from Mary's amount:
 3580
 3581 $\$22 - \$24 = -\$2$
 3582
 3583 However, the problem states that
 3584 Mary has more than Marco, so we need
 3585 to find the correct difference. Since

Mary has \$22 and Marco has \$24, Mary
 has \$2 less than **Marco. ##### -2** 3586
 3587
 ===== 3588
Let's break down the problem step by
step: 1. The paint mixer takes three
 days to make 18 drums of paint. This
 means that the number of drums of paint
 made per day is: 3598
 3599
 3590
 3591
 3592
 3593
 3594
 3595
 3596
 3597
 3598
 2. To find the number of days it
 will take to make 360 drums of paint,
 we can divide the total number of drums
 by the number of drums made per day
 : **$360 \text{ drums} / 6 \text{ drums per day} = 60$**
days ##### 60 3599
 3600
 3601
 3602
 3603
 ===== 3604
 {Let's break down the information given
 in the problem.} 3605
 3606
 3607
 3608
 3609
 3610
 3611
 3612
 3613
 3614
 3615
 3616
 3617
 3618
 3619
 3620
 3621
 3622
 3623
 3624
 3625
 3626
 3627
 3628
 3629
 3630
 3631
 3632
 3633
 3634
 {First, let's find Aubrey's age on
 her 8th birthday.}
 {Now, let's find the age of Max.}
 On Aubrey's 8th birthday, we need to
 find the age of Max.
 Luka turned 4 years old on the day Max
 was born.
 Luka is exactly 2 years older than
 Aubrey.
 On Aubrey's 8th birthday, we need to
 find the age of Max.
 {First, let's find Aubrey's age on
 her 8th birthday.}
 {Now, let's find the age of Max.}
 On Aubrey's 8th birthday, we need to
 find the age of Max.
 Luka turned 4 years old on the day Max
 was born.
 Luka is exactly 2 years older than
 Aubrey.
 On Aubrey's 8th birthday, we need to
 find the age of Max.
 {First, let's find Aubrey's age on
 her 8th birthday.}
 {Now, let's find the age of Max.}

3635 is the same as the number of years that
 3636 have passed since his birth.
 3637 The number of years that have passed
 3638 since Max's birth is Aubrey's age on her
 3639 8th birthday minus her age on the day
 3640 Max was born, which is $8 - 2 = 6$ years.
 3641 Therefore, Max is 6 years old on
 3642 Aubrey's 8th birthday.

3643 ##### 6
 3644

3645 =====

3646 **Let's break it down step by step:**
 3647 **Tommy has 3 toy cars.**
 3648 Jessie has 3 toy cars, just like Tommy.

3649
 3650 Tommy's neighbor, Jessie's older
 3651 brother, has 5 more cars than Tommy
 3652 and Jessie combined. To find out how
 3653 many cars he has, we need to add
 3654 Tommy's and Jessie's cars together and
 3655 then **add 5. Tommy** and Jessie have $3 +$
 3656 $3 = 6$ cars in total.
 3657 Adding 5 more cars to that, Jessie's older
 3658 brother has $6 + 5 = 11$ cars.

3659
 3660 Now, let's count the total number
 3661 of cars the three of them have:
 3662 Tommy has 3 cars.
 3663 Jessie has 3 cars.
 3664 Jessie's older brother has 11 cars.

3665
 3666 In total, they have $3 + 3 + 11 =$
 3667 **17 cars. ##### 17**

3668 =====

3669 Let's break down the problem step by
 3670 step:

- 3671
- 3672 1. The cost of the best-selling book
 3673 "TOP" is \$8, and 13 books were sold.
 3674 So, the total revenue from "TOP" books
 3675 is : **$13 \times \$8 = \104**
 - 3676
 - 3677 2. The cost of the least-selling
 3678 book "ABC" is \$23, and 4 books were
 3679 sold. So, the total revenue from "ABC "
 3680 **books is: $4 \times \$23 = \92** 3. To find the
 3681 difference in the bookstore's earnings on
 3682 these two books last week, we subtract
 3683 the revenue from "ABC" books from

the revenue from "TOP" books: **$\$104 -$**
 $\$92 = \12 ##### 12

3684
 3685
 3686 =====

3687 **Let's break this problem down step by**
 3688 **step! Let the** number of gigs played be
 3689 x.
 3690

3691 Each member earns \$20 per gig,
 3692 so the total earnings per gig is **$4 \times \$20 =$**
 3693 **$\$80.$**

3694
 3695 The band has earned a total of
 3696 \$400. We can set up an equation **based**
 3697 **on this information: $80x = 400$**

3698
 3699 To solve for x, we can divide both
 3700 sides of **the equation by 80:** $x = 400 \div$
 3701 80
 3702 $x = 5$

3703
 3704 **So, the band has played 5 gigs.**
 3705 **##### 5**