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

Jia Liu[1,2], Jiaxin Luo[1,3], Weiwen Xu, Jonathan M. Garibaldi[4], Xiao-Kun Wu[5], Yixue Hao[2], Min Chen[1,3]

[1]School of Computer Science and Engineering, South China University of Technology
[2]School of Computer Science and Technology, Huazhong University of Science and Technology
[3]Pazhou Laboratory, Guangzhou
[4]Provost Office, University of Nottingham Ningbo China
[5]School of Journalism & Communication, Renmin University of China

Correspondence: minchen@ieee.org

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 a diagnostic framework for analyzing how procedural reasoning structures emerge during training, with downstream selection results serving as auxiliary evidence correlating gradient alignment with adaptation efficiency.

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.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 37799–37845
July 2-7, 2026 ©2026 Association for Computational Linguistics

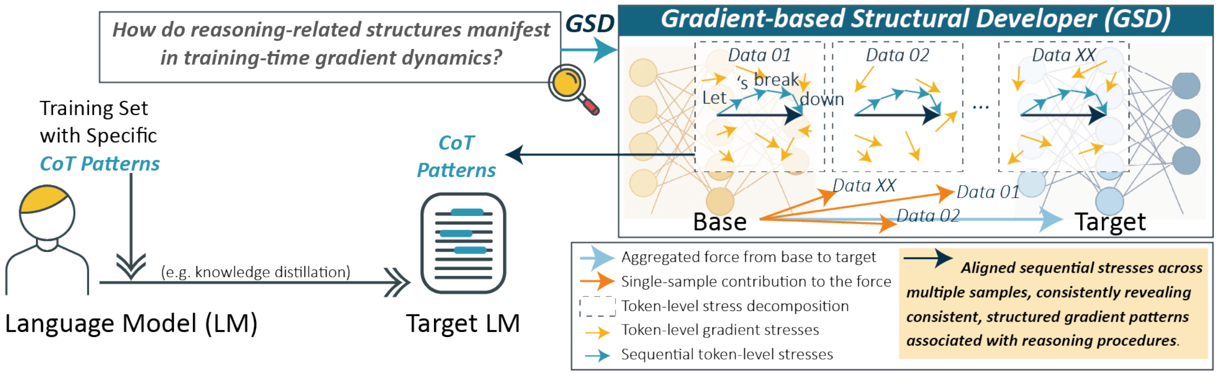


Figure 1: **Problem description and method overview.** We study how reasoning-related structures are expressed during language model (LM) training through span-level gradient dynamics. Chain-of-Thought (CoT) exemplars induce localized gradient signals that, when aggregated across sequences and samples, 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, while downstream correlations are presented solely to validate this signal rather than propose optimization strategies.

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 reflected 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.

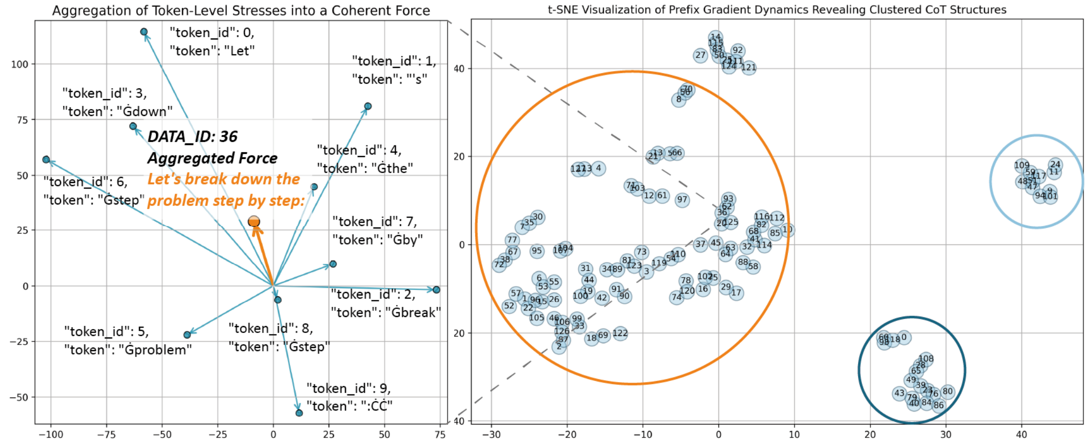
As a preliminary analysis, we extract token-level gradient contributions corresponding to the first ten tokens of each training example, which typically cover the reasoning prefix, and analyze them at a fixed model state to isolate the structure of training signals. We visualize these gradients using 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)**,



DATA ID	Examples of 10-Token Reasoning Prefixes
33	Let's break this problem down step by step!
18	Let's break this problem down step by step!
22	Let's break this problem down step by step!
17	Let's break down the problem step by step:
7	Let's break down the information we have:
35	Let's break down the information we have:

DATA ID	Examples of 10-Token Reasoning Prefixes
47	Here's the correct analysis with the Chain of Thought
48	Here's the correct analysis with the Chain of Thought

DATA ID	Examples of 10-Token Reasoning Prefixes
39	Let's start by finding the age of Henry's
43	Let's start by using the information that Madeline has
49	Let's start by finding the total amount of flour

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.

a training-time analysis framework that characterizes how gradient-induced update directions align 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 $X = (x_1, \ldots, x_L)$, the negative log-likelihood decomposes across positions:

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

The corresponding parameter gradient admits a tokenwise decomposition,

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

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

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:

$$f_l^{(n)} = s_{r_l} + \varepsilon_l^{(n)}, \qquad n = 1, \ldots, N, \quad (5)$$

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

where $n = 1, \ldots, N$ is the sample index, $l = 1, \ldots, L^{(n)}$ 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 $m = b - a + 1$,

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

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

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

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

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

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. Here, positional consistency serves as a weak cross-sample prior that helps organize comparable spans for analysis. Empirically, such aligned stresses recur even when surface-level text differs: comparing GFA to embedding-based cosine similarity shows that many high-GFA span pairs have low semantic similarity, suggesting that gradient alignment captures training-time procedural organization beyond what is recoverable from 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 mod-

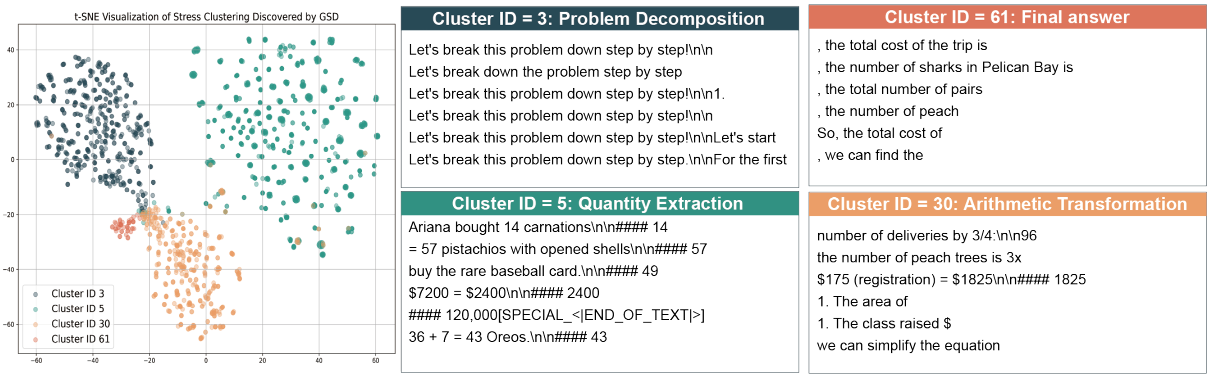


Figure 3: **Unsupervised Discovery of Gradient Structure in Reasoning.** Left: t-SNE visualization of gradient representations derived from continuous token spans during fine-tuning, with points colored by clusters discovered without semantic labels or task annotations. Right: Decoded token spans from selected clusters, illustrating functional consistency within each cluster. The selected clusters correspond to distinct procedural roles in reasoning, including reasoning initiation (Cluster 3), quantity extraction (Cluster 5), arithmetic transformation (Cluster 30), and final answer consolidation (Cluster 61).

ules, 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 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 functional coherence corresponding to procedural roles. Cluster 3 captures reasoning initiation through problem decomposition, illustrated by phrases such as “Let’s break this problem down.” Cluster 5 corresponds to quantity extraction, identifying numerical entities including “14 carnations” and “57 pistachios.” Cluster 30 encodes arithmetic transformation involving explicit operations, while Cluster 61 consolidates final answers with expressions such as “the total cost is ####.” This functional differentiation confirms that gradient alignment reveals procedural roles rather than superficial topics. 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.

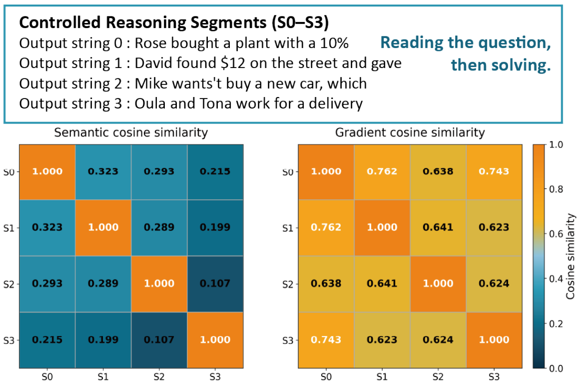


Figure 4: **Semantic similarity vs. gradient similarity for controlled reasoning segments.** Top: four question-reading segments from different GSM8K problems. Bottom: cosine similarity matrices based on semantic embeddings (left) and token-level gradients (right). Despite low semantic similarity, gradient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

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

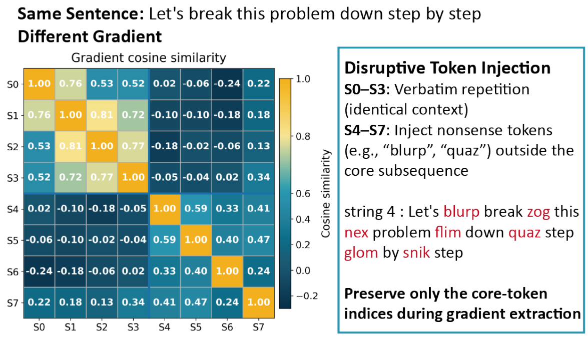


Figure 5: **Identical semantic cores with disrupted gradient alignment.** Gradient cosine similarity for sequences sharing the same core sentence but differing in surrounding tokens. Although the semantic core is preserved, gradient alignment collapses across groups, showing that semantic overlap alone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

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.

Ruling Out Positional Confounds. A natural alternative hypothesis is that gradient alignment merely reflects positional regularities rather than procedural roles. To decouple absolute position from procedural context, we conduct controlled interventions with four conditions: (A) baseline original spans; (B) strict position control where reasoning lines are shuffled but only spans with negligible positional drift ($d_{tok} \leq 0.02$) are retained, yielding 84 position-stable spans with disrupted context; (C) position-matched unrelated baseline with random content at the same position; (D) token-identical context disruption where exact token spans survive shuffling but surrounding procedural context is randomized.

Table 1 reports Gradient Force Alignment against original cluster centroids. Even under strict position control (B), alignment drops substantially from 0.557 to 0.371, demonstrating that positional stability alone cannot sustain the signal. Condition C confirms that unrelated content at the same position collapses to near-zero alignment (0.008), effec-

Sparse Annotation on Canonical CoT

[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 x 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]

Annotation Ratio: 25.7%
Cosine Similarity to Full SFT: 0.86

[Let's break down the problem step by step:
1. A fifth of the apples have worms, which] means 1/5 of 85 apples have worms[. To find this number, we can multiply] 85 by 1/5:
85 ÃĹ (1/5) = 17
So, 17 apples have worms and will be thrown away.
2. Nine more than one fifth of the apples are bruised. This means that 1/5 of the apples are bruised, and then we add 9 to that number. To find this number, we can multiply 85 by 1/5 and then add 9:
85 ÃĹ (1/5) = 17
17 + 9 = 26
So, 26 apples are bruised and will be saved for applesauce.
3. We know that Lex picked 85 apples in total. We've already accounted for 17 apples with worms and 26 apples that are bruised. To find the number of apples left to eat raw, we subtract these numbers from the total:
85 - 17 - 26 = 42
[Therefore, Lex will have 42 apples left to eat raw.
42]

Annotation Ratio: 19.1%
Cosine Similarity to Full SFT: 0.77

Dense Annotation on Canonical CoT

[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% of $125 = 0.25 x $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% of $156.25 = 0.20 x $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 Ã[· $125) x 100 = 100]%
So[, the final balance is 100% of the starting balance.
100]

Annotation Ratio: 62.3%
Cosine Similarity to Full SFT: 0.95

[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 x (1 + [0.15) = $80 x 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 x (1 + 0.10) = $120 x 1.10 = $132
So,[the final portfolio value after 2 years is $132.
132]

Annotation Ratio: 42.9%
Cosine Similarity to Full SFT: 0.89

Noisy or Non-Canonical Reasoning

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 ÃĹ 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) Ã· 15 pugs = 12 minutes
12

Annotation Ratio: 7.1%
Cosine Similarity to Full SFT: 0.37

[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 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 hours - 33 hours = 19 hours[
on Friday to finish the show.
19]

Annotation Ratio: 17.7%
Cosine Similarity to Full SFT: 0.71

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.

Table 1: Gradient Alignment under Positional Controls. Condition B maintains position but disrupts context; C maintains position with unrelated content; D maintains exact tokens but disrupts context.

Condition	N	GFA (mean±std)	Drop (Δ)
A (Base)	557	0.557 ± 0.195	—
B (Pos. Controlled)	84	0.371 ± 0.205	-0.186
C (Unrelated)	557	0.008 ± 0.230	-0.549
D (Token-Identical)	167	0.250 ± 0.291	-0.283

tively ruling out pure positional effects. Crucially, Condition D shows that token-identical spans lose significant alignment when embedded in randomized procedural contexts (0.250 vs. 0.533 originally, $p < 0.001$), proving that surface semantics are insufficient without procedural coherence.

We further quantify positional spread within clusters. If clusters were position-driven, we would expect minimal spread ($\Delta r \approx 0$). Instead, we observe median $\Delta r = 0.62$ (max 0.79), indicating that typical clusters span 62% of the sequence length, directly contradicting a purely positional explanation.

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 from the same 800-sample pool, subsets whose aggregated gradients exhibit higher cosine similarity to a reference force (computed from 800 samples over three epochs) consistently achieve higher GSM8K accuracy. Table 2 shows this advantage persists from $N = 16$ to $N = 128$, though margins decrease as larger subsets dilute the most consistent samples and increase structural overlap between target and control sets.

Table 2: Scalability of Gradient-Consistent Selection. Performance gap (in percentage points) between High-GFC (Target) and Low-GFC (Control) subsets across selection budgets.

N	Low GFC (Control)	High GFC (Target)	Δ (pp)
16	49.36%	59.36%	+10.00
32	51.25%	59.74%	+8.49
64	58.68%	62.70%	+4.02
128	60.35%	64.37%	+4.02

Training on the full 800-sample set reaches 65.05%, close to the 64.37% achieved with 128 high-consistency samples, suggesting diminishing returns beyond $N = 128$. We emphasize that this analysis serves as auxiliary validation that gradient alignment provides a measurable diagnostic signal correlated with adaptation outcomes, not as a claim of a practical data-selection algorithm. Detailed experimental setup is provided in Appendix D.

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.

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.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (2025YFE0213400), in part by the National Natural Science Foundation of China (NSFC) under Grant 62276109, and in part by the Interdisciplinary Research Program of HUST under Grant 5003210069. The authors thank the anonymous reviewers for their constructive feedback. AI assistants were used in a limited manner for language polishing and editorial refinement of the manuscript, but not for generating results, conducting analyses, or making scientific decisions.

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. lpntk: Better generalisation with less data via sample interaction during learning. arXiv preprint arXiv:2401.08808.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. arXiv preprint arXiv:2412.06769.

Jianhao Huang, Zixuan Wang, and Jason D Lee. 2025. Transformers learn to implement multi-step gradient descent with chain of thought. arXiv preprint arXiv:2502.21212.

Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35:22199–22213.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. Measuring faithfulness in chain-of-thought reasoning. arXiv preprint arXiv:2307.13702.

Ayeong Lee, Ethan Che, and Tianyi Peng. 2025. How well do llms compress their own chain-of-thought? a token complexity approach. arXiv preprint arXiv:2503.01141.

Jindong Li, Yali Fu, Li Fan, Jiahong Liu, Yao Shu, Chengwei Qin, Menglin Yang, Irwin King, and Rex Ying. 2025. Implicit reasoning in large language models: A comprehensive survey. arXiv preprint arXiv:2509.02350.

Jia Liu, Yue Wang, Zhiqi Lin, Min Chen, Yixue Hao, and Long Hu. 2024. Natural language fine-tuning. arXiv preprint arXiv:2412.20382.

José I Orlicki. 2025. Beyond words: A latent memory approach to internal reasoning in llms. arXiv preprint arXiv:2502.21030.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. Advances in neural information processing systems, 35:27730–27744.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. Advances in Neural Information Processing Systems, 33:19920–19930.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36:53728–53741.

Leonardo Ranaldi, Marco Valentino, and Andre Freitas. 2025. Improving chain-of-thought reasoning via quasi-symbolic abstractions. pages 17222–17240.

Yi Ren, Shangmin Guo, Linlu Qiu, Bailin Wang, and Danica J Sutherland. 2024. Bias amplification in language model evolution: An iterated learning perspective. Advances in Neural Information Processing Systems, 37:38629–38664.

Yi Ren, Shangmin Guo, and Danica J Sutherland. 2022. Better supervisory signals by observing learning paths. arXiv preprint arXiv:2203.02485.

Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 20:53–65.

Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. arXiv preprint arXiv:1708.00489.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4149–4158.

Roman Vershynin. 2018. High-Dimensional Probability: An Introduction with Applications in Data Science. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Yiming Wang, Pei Zhang, Baosong Yang, Derek F Wong, and Rui Wang. 2024a. Latent space chain-of-embedding enables output-free LLM self-evaluation. arXiv preprint arXiv:2410.13640.

Yu Wang, Shiwan Zhao, Zhihu Wang, Heyuan Huang, Ming Fan, Yubo Zhang, Zhixing Wang, Haijun Wang, and Ting Liu. 2024b. Strategic chain-of-thought:

Guiding accurate reasoning in llms through strategy elicitation. arXiv preprint arXiv:2409.03271.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824–24837.

Skyler Wu, Eric Meng Shen, Charumathi Badrinath, Jiaqi Ma, and Himabindu Lakkaraju. 2023. Analyzing chain-of-thought prompting in large language models via gradient-based feature attributions. arXiv preprint arXiv:2307.13339.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. Preprint, arXiv:2002.04326.

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

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

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

Contents

1 Introduction 1

2 Related Work 2
- 2.1 Chain-of-Thought Reasoning . . . 2
- 2.2 Training-Time Analysis of LLMs . 3

3 Problem Definition and Preliminary Observations 3

4 Methodology 3
- 4.1 Training-time gradient decomposition and interpretation. 4
- 4.2 Span-level aggregation. 4
- 4.3 Measuring alignment and discovering structured stresses. 5

5 Experiments 5
- 5.1 Experimental Setup 5
- 5.2 GSD Analysis of Reasoning Structure 6
- 5.3 Gradient Structure as a Diagnostic Signal of Training Influence . . . 8

6 Conclusion 9

Appendix 13

A Theoretical foundations and methodological clarifications for GSD 14
- A.1 Notation and standing assumptions 14
- A.2 Signal–noise model for span aggregation 14
- A.3 Cross-sample alignment and SNR improvement 15
- A.4 Exact identities and relation to Jacobian geometry 15

B Generalization to Broader Reasoning Tasks 16
- B.1 Logical Deduction (LogiQA) . . . 16
  - B.1.1 Construction of the Distilled Logical Deduction Dataset (LogiQA) 16
  - B.1.2 Structural Stress Patterns in Logical Deduction Tasks 16
  - B.1.3 Qualitative Analysis of Clustered Stress Patterns . 17
  - B.1.4 Per-Sample Token Trace Examples 18
- B.2 Commonsense Reasoning (CSQA) 20
  - B.2.1 Construction of the Distilled Commonsense Reasoning Dataset (CSQA) . . 20
  - B.2.2 Structural Stress Patterns in Commonsense Reasoning Tasks 20
  - B.2.3 Qualitative Analysis of Clustered Stress Patterns . 20

C Quantitative Evaluation of Gradient-Based Alignment 22
- C.1 Clustering Quality of GSD Patterns 22
- C.2 Distributional Statistics of GFA Scores 22
- C.3 Robustness to LoRA Rank 24

D Gradient Force Aggregation and Fine-Tuning Behavior 25
- D.1 Construction of the Target Force and Candidate Pool 25
- D.2 Subset Aggregation and Alignment Measurement 25
- D.3 Fine-Tuning Setup and Evaluation 25
- D.4 Interpretation and Scope 26

E Algorithmic Details 26
- E.1 Unsupervised Clustering of Sequential Stresses 26
- E.2 Greedy Force-Based Data Selection Algorithm 26

F Implementation Details 26
- F.1 Training Hyperparameters and Infrastructure 26
- F.2 Token-Level Gradient Tracing Protocol 27
- F.3 LoRA Configuration and Optimization Settings 27

G Additional Results 28
- G.1 Expanded Visualizations of Stress Clusters 28
  - G.1.1 Strongly Separated Clusters 28
  - G.1.2 Moderately Cohesive Clusters 32
  - G.1.3 Low-Cohesion or Noisy Clusters 35
- G.2 Per-Sample Token Trace Examples 36

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 $Y^{(n)} = (y_1^{(n)}, \ldots, y_{L^{(n)}}^{(n)})$. When unambiguous, we omit the superscript.

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

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

and the per-sample loss decomposes as

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

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

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

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

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

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

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

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

$$f_l = J_l(\theta)^\top g_l, \tag{16}$$

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

$$f_l^{(n)} = s_{r_l^{(n)}} + \varepsilon_l^{(n)}, \qquad n = 1, \ldots, N, \tag{17}$$

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 $m = b - a + 1$,

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

   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 $(r_a, \ldots, r_b)$ the aggregated stress decomposes as

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

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

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

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 $(r_a, \ldots, r_b)$. Recall

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

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

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

Define the deviation

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

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

$$\mathbb{E}\|E^{(k)}_{[a:b]}\|^2 \le \sigma^2 m, \tag{25}$$

and therefore

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

$$\mathbb{E}[B^2] \le \sigma^4 m^2. \tag{27}$$

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

$$\mathrm{Var}(\Delta) \le 2\|S_{[a:b]}\|^2 \sigma^2 m + \sigma^4 m^2. \tag{28}$$

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

$$\mathbb{P}(|\Delta| \ge t) \le \frac{\mathrm{Var}(\Delta)}{t^2}, \tag{29}$$

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

Strong (exponential) concentration under sub-Gaussian noise. If additionally each scalar projection $\langle v, \varepsilon^{(n)}_l \rangle$ 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 $t > 0$

$$\mathbb{P}(|\Delta| \ge t) \le 2\exp\Big(-c \min\Big\{\frac{t^2}{\sigma^2 m \|S_{[a:b]}\|^2 + \sigma^4 m^2}, \frac{t}{\sigma^2 m}\Big\}\Big). \tag{30}$$

(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 $\mathrm{SNR} \triangleq \|S_{[a:b]}\|^2/\sqrt{\mathrm{Var}(\Delta)}$. 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 $\langle s_{r_l}, u\rangle \ge \mu_{\min}$ for every $l \in [a : b]$. Then

$$\|S_{[a:b]}\| \ge \mu_{\min} m, \tag{31}$$

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

$$\langle f_i, f_j\rangle = \langle J_i^\top g_i, J_j^\top g_j\rangle = g_i^\top \big(J_i J_j^\top\big) g_j, \tag{32}$$

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

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

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.

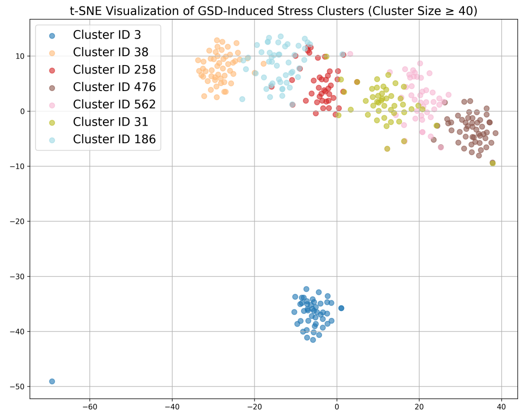


Figure 7: t-SNE visualization of GSD-induced token-level stress clusters on the LogiQA dataset. Only clusters with size ≥ 40 are shown. Each point represents a token-level stress vector.

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 3 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 3: Average Silhouette Score per Cluster (LogiQA, Cluster Size $\geq$ 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 $\geq$ 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.

but doesn’t refute the causal link
people ignore objectionable consequences of actions supporting
inherent downsides.
potential ozone layer damage from
due to the absence of
government regulation due to the
Analyzing the options:- **
highlighting the severity of water scarcity
the recovery must stem from
express unconscious thoughts but doesn

Cluster ID 562. Cluster 562 (Silhouette score: 0.2716) contains tokens involved in premises reweighting and nuanced critique. Tokens like "points to consider," "just the lack of reward," "anarchists commit violence," and "another ideology..." reflect subtle argument evaluation and hypothesis differentiation. The lower silhouette score corresponds to semantic variability, as tokens encode more heterogeneous argumentative forms.

evaluation, which is irrelevant
firms with PRP had 13% higher productivity.
points to consider:
Since *selected* construction
satellites for enabling early environmental problem detection but
overlooking potential ozone layer damage from
negative consequences of actions that benefit their cause
the conclusion (showing the issue is lack of rewards
points to consider:
just the lack of reward

Cluster ID 31. This cluster (Silhouette score: 0.1245) has the lowest internal coherence, likely due to its coverage of syntactically and semantically diverse middle-stage reasoning. Tokens include "without special exhibitions," "modernized equipment," "analogies to hold," and "counterexamples." This group appears to encode domain-specific justifications and factual grounding, functioning as semantic fillers in broader logical chains.

building supplies). Thus,
productivity boost could stem from
modernized equipment and adopted
weaken the argument by implying
conditions in Ireland must be
argument praises space-based satellites for enabling early environmental problem
a moderate scenario that could
without special exhibitions**, attendance
reasoning to hold, the
alone doesn’t explain why

Cluster ID 186. Cluster 186 (Silhouette score: 0.2805) aligns with interpretive explanation and pragmatic support statements. Examples include "critics’ ability to help," "assumes accurate identification," "citizen action," "security principle," and similar evaluative or supportive units. These tokens help to reinforce the conclusion or highlight potential flaws in assumptions, consistent with the cluster’s mid-to-late role in argumentative structure.

is flawed), we need an assumption that
critics’ ability to help
is undermined. If it does not decline, the argument is supported.
think step by step.
directly ties citizen action to
this down logically. The
Correct. The argument never
’s think step by step.
lifestyle) might explain both

B.1.4 Per-Sample Token Trace Examples

SAMPLE 1

annotation ratio = 24.5%
GFA score = 0.7597

> Let **’s think step by step. First, let’s analyze the problem. The** manager argues against offering free gift wrapping because:
> - ****If most customers use it**, it** ś expensive and time-consuming.
> - **If few **customers use it**, thereś** no advantage.
>
> The conclusion is that offering it has no benefit in either scenario.
>
> **To identify the assumption, we need a premise that bridges** the gap between the two scenarios and the conclusion. The argument implicitly assumes that **only two outcomes are possible**: either "most" or "few"

customers use the service. This excludes **a middle ground (e.g.,** a moderate number of customers **) where the costs might be** manageable *and* the service provides a net advantage.

Analysis of options:
- **Option 0**: Discusses cost of *informing* customers, which the argument never mentions. Irrelevant.
- **Option 1**: Compares costs to previous seasons, but the argument focuses on uptake (most/few), not historical costs. Irrelevant.
- **Option 2**: States that either few or most customers would want it. This matches the managerś dichotomy and is **necessary to rule out a moderate scenario that could justify offering the service. **** Required assumption ****. - **Option** 3**: Links slowing down shoppers to costs. While time-consuming is mentioned, the argument doesnt́ assume *all* slowdowns cost money only those tied to high uptake. Not required.

2

============================

SAMPLE 2
annotation ratio = 38.9%
GFA score = 0.7802

Let's break this down **logically. The environmentalist's** argument hinges on the necessity of **government regulation due to the industry's inaction.** For this **reasoning to hold, the environmental** ist must assume that no other avenues exist to **enforce stricter safety standards beyond the** industry itself **or the government. If alternative effective sources (e.g., international bodies, local governments) existed** , the claim that "national government regulation" is required would be weakened.

Analyzing the options: - **Option 0** discusses negotiation processes, which **isnt́ directly tied to the** necessity of government action.
- **Option 1** introduces a comparison between environmental and financial priorities, but the argument focuses on *who* should act, not *why* one priority outweigh **s another. - **Option 2** claims cost-saving benefits of double hulls, but the** environmentalistś stance doesnt́ depend on long-term savings only on preventing spills.
- **Option 3** addresses the exclusivity of enforcement **sources, directly supporting the environmental** istś premise that government regulation is the only viable alternative to industry action.

Thus, the environmentalist assumes no other effective sources exist besides the industry or government.

3

============================

SAMPLE 3
annotation ratio = 39.3%
GFA score = 0.8752

Let **'s think step by step. First, let's analyze the problem. The** argument begins by **highlighting the severity of water scarcity and** pollution, predicting future conflicts. However **, it shifts focus to solutions,** emphasizing citizens' potential to address the crisis through activism (lobbying governments, recycling, reducing biological footprint). The conclusion **directly ties citizen action to mitig** ating the crisis.

Key points to consider:
1. The argument identifies a problem (water crisis) but prioritizes actionable solutions.
2. Specific steps for **citizens (lobbying, recycling) are central. 3. The conclusion explicitly states that active** citizen response can mitigate the crisis.

Options analysis:
- **Option 0** (crit **icize government) is incorrect** because the focus is on citizen agency, **not government failure.**
- **Option 1 (spur activism) aligns** perfectly with the call for citizens to lobby, recycle, and act.

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

The argumentś 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.

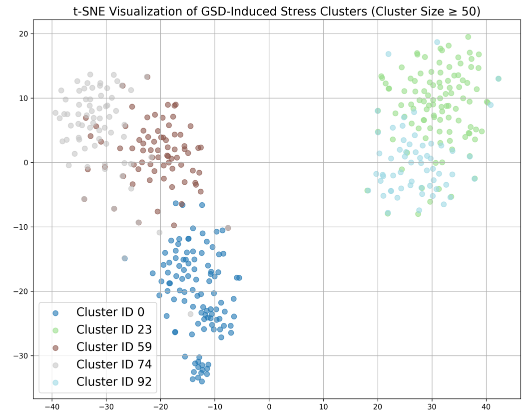


Figure 8: t-SNE visualization of GSD-induced token-level stress clusters on the CSQA dataset. Only clusters with size ≥ 50 are shown. Each point corresponds to a token stress vector.

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 4 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 4: 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

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.

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.

```
Let's think step by step. Sammy wants to be
is to find a choker
The remaining logical option is a
Let's think step by step
Let's think step by step;
Let's think step by
Between 'shop' (generic
Let's think step by
's think step by step
refers to the head of
```

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 stepwise explanations. The silhouette score (0.2183) indicates weaker coherence, likely due to semantic drift in intermediate steps of reasoning.

```
step by step. First, the goal
goal is to find a choker not
step. First, let
let's analyze the situation
step. First, let
let's analyze the situation:
, let's analyze the
it's implausible. Windowsill
this, we need to apply some everyday
let's analyze the situation
```

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.

```
by not acknowledging them. "
aligns with the punitive
typically worn around the neck
with this reasoning, as
it the most strategic location
are possible, bitterness specifically captures resentment
pet shops are the logical
of a national government, not a business or
the president is the governing
strain. To recover, the body and mind need downtime
```

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.

```
  or public spaces.  This aligns with
common
  gregate, like cities or public
  by not acknowledging them.
  Let's think step by step.
  specifically designed to soak up
  possible,   bitterness   specifically
captures resentment
  knowledge. What typically happens
  place. The solar system
  time, it likely means he
  skills but doesn't immediately
```

Cluster ID 92. Cluster 92 aggregates tokens contributing to nuanced or emotionally grounded interpretations, often near the end of explanations. Tokens such as "grapes are typically...", "resentment due to...", "relaxed (C)...", or "barbeque meat..." represent affective, environmental, or cultural context elements that influence human-centered reasoning. These often involve multi-modal associations and pragmatic knowledge, making the cluster lexically varied but conceptually rich. The silhouette score (0.2939) reflects this balance between diversity and thematic focus.

```
  string bag, which implies
  arises from resentment due to
  meat on a grill aligns with "bar
  involves direct heat, but
  Painting and movie hall are human-made
  step by step. A thirsty
  people hurry when they're
  step by step: The
  physical     exercise,     typically
associated   with   workouts   or   group
drills
  stration is a common reaction
```

C Quantitative Evaluation of Gradient-Based Alignment

C.1 Clustering Quality of GSD Patterns

In addition to distributional and performance-based analyses, we quantitatively assess the structural coherence of token clusters revealed by the Gradient-based Structural Developer (GSD). We apply three widely used unsupervised clustering evaluation metrics to the t-SNE projections of stress-aligned token representations:

- **Silhouette Score** (Rousseeuw, 1987): Evaluates how similar each token is to its own cluster relative to other clusters, ranging from -1 to 1, where higher values indicate better-defined boundaries.
- **Calinski–Harabasz Index** (Caliński and Harabasz, 1974): Measures the ratio of between-cluster dispersion to within-cluster cohesion; higher values reflect more compact and well-separated clusters.
- **Davies–Bouldin Index** (Davies and Bouldin, 2009): Computes intra-cluster similarity versus inter-cluster dissimilarity, where lower values indicate higher-quality clustering.

These metrics confirm that the token-level clusters induced by GSD are not only visually distinct but also statistically well-formed in latent space, offering robust evidence of reasoning-related alignment patterns.

To evaluate the structural coherence of stress clusters discovered by GSD, we apply a cluster size threshold filter to retain only clusters containing more than a given number of samples. Table 6 through Table 9 report the mean Silhouette Score of each retained cluster under thresholds $\geq$50, 55, 60, and 65, respectively. A comparative summary of global clustering metrics is provided in Table 5.

We observe that as the minimum cluster size increases, the **average Silhouette Score and Calinski-Harabasz Index both improve**, while the **Davies-Bouldin Index consistently decreases**, indicating more compact and better-separated clusters. Notably, Cluster ID 282 and 556 consistently achieve high internal consistency across all thresholds.

To complement the quantitative clustering metrics, we visualize the high-dimensional stress vectors using t-SNE for each threshold configuration. Figures 9–12 display the clustering results for minimum cluster sizes $\geq$50, 55, 60, and 65, respectively.

As the threshold increases, less coherent clusters are progressively filtered out, leading to improved separation and tighter cohesion. This observation is consistent with the rising silhouette scores and declining Davies-Bouldin indices, as discussed earlier.

C.2 Distributional Statistics of GFA Scores

To examine whether full supervision is necessary for effective gradient alignment, we analyze the GFA distribution under a pruning strategy where

Table 5: 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 6: 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 7: 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 8: 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 9: 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				

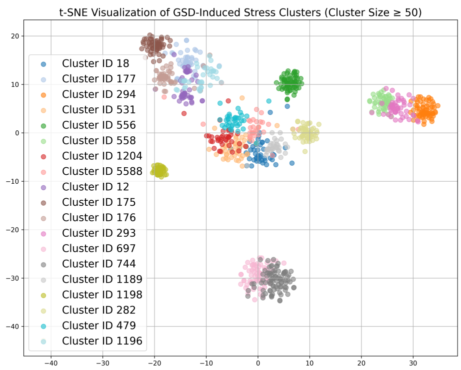


Figure 9: t-SNE visualization of stress clusters discovered by GSD with minimum cluster size ≥ 50. A total of 19 clusters are retained, showing a wide variation in spatial separation and density. Some clusters, such as ID 556 and ID 1198, form tight and well-separated groups, while others like ID 18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

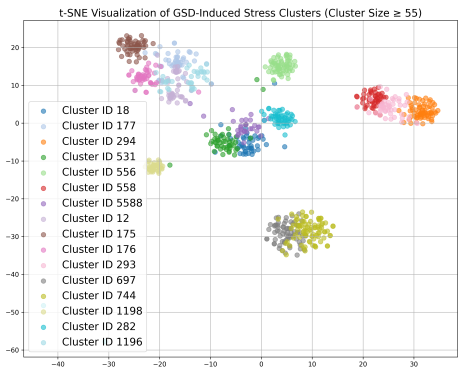


Figure 10: t-SNE visualization of stress clusters with size threshold $\geq$ 55. After filtering, 16 clusters are retained. Spatial coherence improves compared to the previous setting, with visibly tighter clusters such as ID 294 and ID 1198.

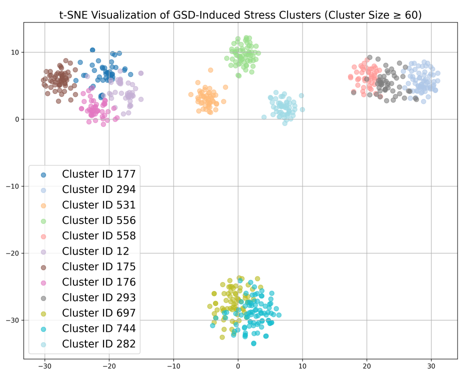


Figure 11: t-SNE visualization for clusters of size $\geq$ 60. Only 12 clusters remain, demonstrating further consolidation of structure. Notably, clusters such as ID 282 and ID 556 are clearly separated, suggesting strong internal consistency.

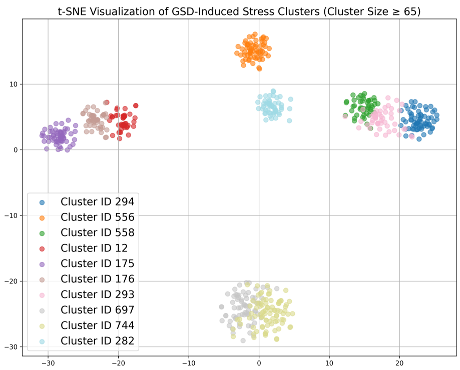


Figure 12: t-SNE plot with cluster size threshold $\geq$ 65. Only 10 dense and well-separated clusters are retained. Most clusters exhibit strong internal cohesion and clear boundaries, highlighting the robustness of GSD clustering under stricter size constraints.

only tokens with importance ratio $\geq$ 0.1 are retained. Table 10 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 11, 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 12 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.

C.3 Robustness to LoRA Rank

We verify that the observed gradient structure is not an artifact of the rank-16 LoRA configuration by ablating $r \in \{8, 16, 64\}$ while holding the data and pipeline fixed. Table 13 summarizes the cluster distribution statistics.

As rank increases, cluster concentration monotonically decreases (Top-5 share: $0.389 \rightarrow 0.326 \rightarrow 0.137$; HHI: $0.074 \rightarrow 0.029 \rightarrow 0.006$) while diversity increases (eff.K: $13.6 \rightarrow 34.3 \rightarrow 155.5$). This indicates that rank primarily con-

Table 10: 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 11: 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 12: 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

trols the *resolution granularity* of clustering—how finely recurring reasoning fragments are partitioned—rather than creating the structure itself. The persistence of interpretable procedural roles across ranks (scaffold → transform → answer) confirms that gradient alignment reflects data-driven patterns rather than low-rank constraints.

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.

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,

Table 13: Cluster distribution statistics across LoRA ranks. Lower Top-5 share and HHI indicate decreased concentration; higher entropy and eff.K indicate increased diversity.

Rank	#Clusters	Top-5 Share	HHI	Norm. Entropy	eff.K
8	398	0.389	0.074	0.767	13.6
16	294	0.326	0.029	0.815	34.3
64	595	0.137	0.006	0.937	155.5

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.

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

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

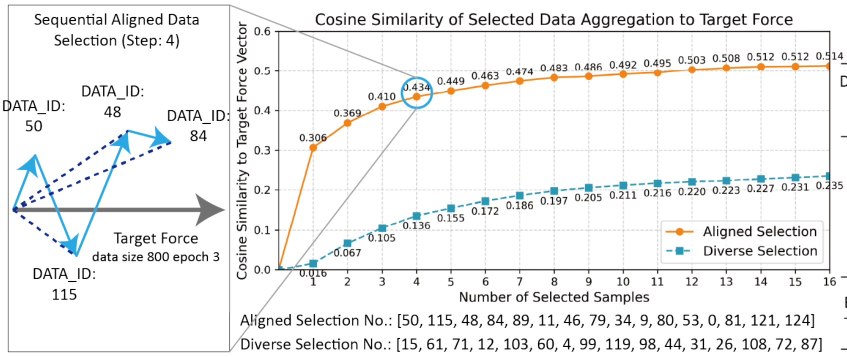
α, β are weighting parameters. The algorithm greedily picks the sample with the highest F_i into S, updates any necessary statistics, and repeats until the desired subset size or a force threshold is reached.

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.



Effect of Aligned Force Aggregation on Fine-Tuning Performance (Acc on GSM8K)

Data size with train epoch	Aligned		Diverse	
	Cos Sim	Acc(%)	Cos Sim	Acc(%)
8 (ep.28)	0.483	***49.0***	0.197	42.0
12 (ep.24)	0.503	***54.7***	0.220	48.7
16 (ep.20)	0.514	***55.0***	0.235	51.9

Base (llama3-8b) Acc on GSM8K: 28.1%
Target (size. 800 ep.3) Acc on GSM8K: 71.4%

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.

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

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.

F.2 Token-Level Gradient Tracing Protocol

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:

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

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

   where A and B are the low-rank adaptation matrices, and α/r is the LoRA scaling factor.
5. Compare the directional consistency of the computed ΔW against a reference stress alignment map, verifying whether the gradient supports or contradicts the expected optimization signal.

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

F.3 LoRA Configuration and Optimization Settings

All LoRA adapters are inserted into the MLP projection layers of the transformer, specifically targeting the modules `gate_proj`, `down_proj`, and `up_proj`. The adapters are configured with rank $r = 16$, scaling factor $\alpha = 16$, and dropout rate 0.05.

To prevent numerical instability and preserve interpretability of adapter gradients, LoRA parameters are excluded from 8-bit quantization. Instead, they are trained in full precision while other modules are quantized using standard 8-bit techniques. Gradient checkpointing is enabled to reduce memory usage and to support long-sequence backpropagation.

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

Input: Sequence set $\mathcal{S} = \{\mathbf{s}_i\}_{i=1}^N$, 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 $\mathbf{s} \in \mathcal{S}$ **do**
  for $w = L_{min}$ **to** L_{max} **do**
    for $t = 1$ **to** $|\mathbf{s}| - w + 1$ **do**
      $\mathbf{f} \leftarrow \frac{1}{w} \sum_{j=t}^{t+w-1} \mathbf{s}_j$; `// treat 2D as 3D for consistency`
      Append $(\mathbf{s}, t, t+w, \mathbf{f})$ to $\mathcal{W}$;
    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 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
`// 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 $\mathrm{IoU}((s,e),(s',e')) < \tau_{iou}$ 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
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**
    $A_i \leftarrow \mathrm{sim}(f_i, c_{\text{global}})$;
    $R_i \leftarrow |S| > 0 \; ? \; \frac{1}{|S|} \sum_{f_j \in S} \mathrm{sim}(f_i, f_j) \; : \; 0$;
    $F_i \leftarrow \alpha\, A_i - \beta\, R_i$;
  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

G Additional Results

G.1 Expanded Visualizations of Stress Clusters

Cluster Analysis Overview. We analyze the internal structure and cohesion of 21 discovered clusters, identified by their respective cluster IDs: 18, 177, 294, 531, 556, 558, 1204, 1611, 5588, 12, 175, 176, 293, 697, 744, 1189, 1198, 282, 479, and 1196. Each cluster exhibits distinct gradient-level and structural characteristics based on token-level alignment patterns. Clusters such as 556, 1198, 531, and 282 show strong silhouette scores, indicating well-separated reasoning patterns, while others like 1611 and 12 demonstrate lower cohesion or potential misalignment. These clusters serve as the basis for our fine-grained analysis of reasoning stress localization across diverse samples.

G.1.1 Strongly Separated Clusters

Clusters exhibiting strong structural separation include **556**, **1198**, **531**, **175**, **282**. These clusters present high silhouette scores (typically above 0.6), suggesting coherent and distinct internal reasoning structures. The following are representative sequences sampled from each cluster:

Cluster ID 556. `student is:120 seconds - 105 seconds = 15 seconds####`

vehicles) + $175 (registration) = $1825####
60:6 hours x 60 = 360 minutes####
they require $49 more to buy the rare baseball card.####
days of the week.####
24 = $64####
42 = 28.####
= 120,000 trees####
24 = 2####
74 questions right.####
$150 - $140 = $10####
$500 - $74 = $426####
6 megabytes/picture = 4,000 pictures####
to spend 50 hours to complete the trip to the hotel.####
+ 80 + 60 = 180 days####
57 pistachios with opened shells####
were 15 individuals left at the zoo.####
160 + 180 + 200= 540 lbs####
5.00####
of juice left.####
24 + 10 = 34 trash cans in Veteran's Park.####
pages with text####
purple flowers) + 7 (green flowers) = 35####
11 hours each day.####
12 (birds) + 18 (fish) = 39####
ÃĹ 2 fillets/fish = 120 fillets####
x 2 = 12 leaves####
ow) + 10 (Cat Cafe Paw) = 40####
$110= $310####
24 + 8 + 8 = 40 cases####
apples left to eat raw.####
Colton has 42 stickers left.####
worth of organisms traveling together for this walk.####
18 years####
Ã· 20 ounces/steak = 4 steaks####
1423 more pages than Beatrix.####
of chocolates left.####
1.00.####
flower costs $15,
divide both sides by $15
is:20 points / 4 players = 5 points/player####
Ela's firewood = 13####
= 14####
2 years is $132.####
25 = 10So, Purple Valley has 10 skirts.####
= 180 pages.####
is:120 fruits/tree ÃĹ 8 trees = 960 fruits####
ane picked up 62 pounds of garbage.####
, Juan picked 61 oranges.####
540 - 104 = 436.####
360 + 90 = 450####
have 7 fish.####
5 = $3####
22.5 - 10.5 = 12 centimeters####
Area = 196 square inches####
6 + 20 + 1 = 27 slices.####
150 (part) = $300####
960,000####
800 bricks to complete his fence.####
:360 drums / 6 drums per day = 60 days####
7 minutes (for marinara stain) = 19 minutes####
one amoeba to divide into 16 amoebae.####
"TOP" books:$104 - $92 = $12####
the band has played 5 gigs.####
end of 2019.####

Cluster ID 1198. by step:1.
by step:1. Layla
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1. Jaime saves $50
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.

by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
by step:1.
:1. Bucky
by step:1. The
by step:1.
by step:1.
by step:1.

Cluster ID 531. last three students is 105 seconds
4 runs =
much more money they require
.He then brings the remaining
has 10 -
keep 2 pieces of fruit
pay for the month is
6 large pieces,
of Oreos Jordan has
the trip to the hotel
lbs less than Jamison
12 bottles in 1
times as old as Aaron
is 15 years old
long doing accounting as calling clients
, and 11 students
9, 10,
the average number of goals
has 400g of
so the amount of flour
2. Each fish gives
many cats are there in
30 cats in Cat Cafe
- 2000 pounds = 500 pounds
{Mike wants to buy
8 customers bought 3 cases
his friend Mandy
4 stickers each to
14 - 10 = 4 stickers
total weight of plates per
organisms traveling together for this
game is $900 per
take his son to see
Jane gave to the boys
cookies for 20% more than it costs to make them
as many skirts as Seafoam Valley.
67 = 40
x 0.25 =
she reads 20 pages
picked up 4 times as many pounds of
by step:1. Johny traveled South 40
Del and Juan is
.3. She then
by step:1.
Each person's share x
third year, which is
how much more money Mary has
add it to the height

Cluster ID 175. Let's break this problem down step by step!
Let's break this problem
Let's break down the
Let's break this problem down step by step!
Let's break this problem
Let's break this problem down step by step.
Let's break this problem down step by step!
Let's break this problem down step by step!
hours2. They
's break this problem down step
Let's break down the problem step by step
Let's break this problem down step by step!1
Let's break this problem down step by
Let's break down the
Let's break this down step
Let's break down the problem step by step:
Let's break this problem down step
Let's break this problem down step by step
Let's break this problem down step by step!
Let's break this problem down step by step!
Let's break this problem down step by step!
Let's break this down step by step!
Let's break this problem down step by step!

, we are asked to
Let's break this problem down step by step
Let's break down the problem
Let's break down the
's break this down step by step!
Let's break down the problem
Let's break it down step by step!
Let's break this down step by step!
Let's break it down step
Let's break down the problem step by step
Let's break down the
's break this problem down step
shells - Number of shells
Let's break this problem down step by step!
Let's break this problem down step by step
Let's break this problem down step by step:
Let's break this problem down step
's the step-by-step
Let's break this problem down step by step!
Let's break this problem down step by step:1
Let's break this problem down step by step.
Let's break down the problem step by step:
Let's break it down step by
Let's break this problem down step by step!
Let's break this problem down step by step!
Let's break it down step
Let's break down the
Let's break down the problem step by step
Pablo's mother agrees to pay him one cent for every page he
Let's break down the problem step by step:
Let's break down the problem step by step:
Let's break it down step
Let's break this problem down step
Let's break this problem down step by step!Let

Cluster ID 282. - 10 = 14
analysis:Mark had 10 pieces
20% of the
!1. Mary weighs
10 trash cans were
Week 2) +
9 cakes x 100
2. There is a 20-minute advertisement before the start of each
6 (dogs) +
original number by 2:
x 2 = 12 leaves
. Mr. Ray has 10 tuna, each weighing 200 pounds
customers x 25 pounds
= 2500 pounds
customers#### Correct!
$3000Total
Colton had 72
silverware) + 24 ounces (
. Robie bought
= 4 bags
So, Robie
. Jayden received
1.2 = 1.2
x 1.2 =
by 60:(cost
.Since Joe sold
(cost of making each cookie) = 1
6 lilies in
. Kimberley collects 10 pounds of firewood
, we calculate:$120
. Jim then adds another $28 to his portfolio
10%. To find
= $132So,
. Daliah picked up 17.
15.5 x
= 120 miles North
. Lorenzo placed one thumbtack from each of the
. The three friends pay
3 = 129
rained 2.5 centimeters
Mondays, we subtract the
imeters, so the total
able to make 10
diaper for $5,
/pack = 192
0 packs x 160
by year:Year 1

. Each wall is
grass stains) + 7 minutes (for marinara stain) =
Tommy has 3
add 5.Tom
ae every 2 days

G.1.2 Moderately Cohesive Clusters

Clusters such as **294**, **176**, **744**, **177** exhibit moderate cohesion, with silhouette scores between 0.2 and 0.5. These clusters often encode partially aligned reasoning patterns with some diversity across examples.

Cluster ID 294. bought 14 carnations.#### 14[SPECIAL_<|END_OF_TEXT|>]
.#### 20[SPECIAL_<|END_OF_TEXT|>]
6 hours x 60 = 360 minutes#### 360[SPECIAL_<|END_OF_TEXT|>]
.#### 7[SPECIAL_<|END_OF_TEXT|>]
final balance is 100% of the starting balance.#### 100[SPECIAL_<|END_OF_TEXT|>]
points: 70 - 42 = 28.#### 28[SPECIAL_<|END_OF_TEXT|>]
120,000[SPECIAL_<|END_OF_TEXT|>]
2[SPECIAL_<|END_OF_TEXT|>]
.#### 74[SPECIAL_<|END_OF_TEXT|>]
$150 - $140 = $10#### 10[SPECIAL_<|END_OF_TEXT|>]
:100 x 20 = 2000#### 2000[SPECIAL_<|END_OF_TEXT|>]
180 days#### 180[SPECIAL_<|END_OF_TEXT|>]
, there were 15 individuals left at the zoo.#### 15[SPECIAL_<|END_OF_TEXT|>]
lbs#### 540[SPECIAL_<|END_OF_TEXT|>]
370[SPECIAL_<|END_OF_TEXT|>]
).#### 5[SPECIAL_<|END_OF_TEXT|>]
.00#### 1035[SPECIAL_<|END_OF_TEXT|>]
.#### 150[SPECIAL_<|END_OF_TEXT|>]
180 (Henry) = 240#### 240[SPECIAL_<|END_OF_TEXT|>]
.#### 70[SPECIAL_<|END_OF_TEXT|>]
.#### 10[SPECIAL_<|END_OF_TEXT|>]
10 = 34 trash cans in Veteran's Park.#### 34[SPECIAL_<|END_OF_TEXT|>]
+ 42 + 63 = 231#### 231[SPECIAL_<|END_OF_TEXT|>]
.#### 32[SPECIAL_<|END_OF_TEXT|>]
92.#### 92[SPECIAL_<|END_OF_TEXT|>]
pages with text#### 19[SPECIAL_<|END_OF_TEXT|>]
in his tank at the end of the heavy rain.#### 280[SPECIAL_<|END_OF_TEXT|>]
35#### 35[SPECIAL_<|END_OF_TEXT|>]
120 fillets#### 120[SPECIAL_<|END_OF_TEXT|>]
6 leaves x 2 = 12 leaves#### 12[SPECIAL_<|END_OF_TEXT|>]
Correct!#### 20[SPECIAL_<|END_OF_TEXT|>]
40[SPECIAL_<|END_OF_TEXT|>]
Lex will have 42 apples left to eat raw.#### 42[SPECIAL_<|END_OF_TEXT|>]
) + 4 (cat) = 22#### 22[SPECIAL_<|END_OF_TEXT|>]
.#### 42[SPECIAL_<|END_OF_TEXT|>]
5040[SPECIAL_<|END_OF_TEXT|>]
this walk.#### 12[SPECIAL_<|END_OF_TEXT|>]
1423[SPECIAL_<|END_OF_TEXT|>]
.#### 4[SPECIAL_<|END_OF_TEXT|>]
1.00[SPECIAL_<|END_OF_TEXT|>]
) = 13 apples#### 13[SPECIAL_<|END_OF_TEXT|>]
- 4 = 8 fish.#### 8[SPECIAL_<|END_OF_TEXT|>]
.#### 660[SPECIAL_<|END_OF_TEXT|>]
5[SPECIAL_<|END_OF_TEXT|>]
's firewood = 13#### 13[SPECIAL_<|END_OF_TEXT|>]
after 2 years is $132.#### 132[SPECIAL_<|END_OF_TEXT|>]
375 pounds per square inch = 600 pounds#### 600[SPECIAL_<|END_OF_TEXT|>]
= 180 pages.#### 180[SPECIAL_<|END_OF_TEXT|>]
people#### 80[SPECIAL_<|END_OF_TEXT|>]
Zane picked up 62 pounds of garbage.#### 62[SPECIAL_<|END_OF_TEXT|>]
+ 60 + 120 = 220 miles#### 220[SPECIAL_<|END_OF_TEXT|>]
now.#### 40[SPECIAL_<|END_OF_TEXT|>]
oranges.#### 61[SPECIAL_<|END_OF_TEXT|>]
2888[SPECIAL_<|END_OF_TEXT|>]
540 - 104 = 436.#### 436[SPECIAL_<|END_OF_TEXT|>]
after five years, Tony will have 7 fish.#### 7[SPECIAL_<|END_OF_TEXT|>]
centimeters#### 12[SPECIAL_<|END_OF_TEXT|>]

.#### 27[SPECIAL_<|END_OF_TEXT|>]
70 = 140 cents#### 140[SPECIAL_<|END_OF_TEXT|>]
960,000[SPECIAL_<|END_OF_TEXT|>]
800 bricks to complete his fence.#### 800[SPECIAL_<|END_OF_TEXT|>]
books.#### 12[SPECIAL_<|END_OF_TEXT|>]
.#### -2[SPECIAL_<|END_OF_TEXT|>]
+ 3 + 11 = 17 cars.#### 17[SPECIAL_<|END_OF_TEXT|>]
books:$104 - $92 = $12#### 12[SPECIAL_<|END_OF_TEXT|>]
So, the band has played 5 gigs.#### 5[SPECIAL_<|END_OF_TEXT|>]
.#### 21[SPECIAL_<|END_OF_TEXT|>]

Cluster ID 176. break this problem down step by step
break this problem down step by step!
break this problem down step by step
break this problem down step by step
break down the information we
break this problem down step
break down the problem step
break this problem down step by
break this problem down step
break this problem down step by step.
break this problem down step by step!
43 Oreos.
break this problem down step by step
break this problem down step by step
break this problem down step by
break this problem down step by step
break down the information we have:
break this down step by
break this problem down step
break this problem down step by step
break this problem down step by step!
this problem down step by
break this problem down step by step!
break this down step by step!
this problem down step by
problem down step by step
this problem down step by
break this down step by step
break this down step by
break this problem down step
break this problem down step by step!
break this problem down step by step
this problem down step by step
break this problem down step
break this problem down step by step
this problem down step by step
break this problem down step by step:1
break this problem down step by step.
break it down step by
break this problem down step
break this problem down step
for every page he reads
problem down step by step
break it down step by step:
break this problem down step by step
break this problem down step

Cluster ID 744. + $175 (registration) = $
- $51 = $49
2 = $15
x $156.25 = $31.25So, she subtract
:20% of $156.25 = 0.20 x $
$31.25 = $156.25
ed $31.25
31.25So
$31.25 to
$125) x 100 =
$156.25 - $31.25 = $
:$125 + $
the account.5.
to the account.3
balance, $125.
9600 - $7200 = $2400
the month is:72 deliveries x $100 per delivery = $
= $7200
$100 per delivery =
= $22* Senior citizen
$9 = $18
per student = $140
$100 = $150
student = $100
3. The total
$27*
20 + $15 =
$2.00 =
= $18.00
5.00 = $225.00
.00 + $225.00 =
$810.006.
$2.00
:$20.00 x
price by the discount percentage:
$200 - $50 = $150
Therefore, James has to pay $150.
.25 = $50
minus $46 = $54).3.

to save $54.
is $50.2
2 = $100, minus $
to save $54:
:$54 / $
%= $20000 x 0.15
00 + $15000
00 x 15%=
+ $5=
420= 0.05 x $420= $21
$420 - $110
= 5% of $420=
= $84 + $21 +
= 0.20 x $420= $84
900= $27,000
$0.35 =
flower = $2160
:144 flowers x $
92 + $28 = $
6 = $60.3.
x $6 = $36.
$202.
$135,000
Ã· 4 = $600,000 Ã· 4 = $
= $150,000
1 - 0.10) = $
= $600,0002. James splits the revenue with his
= $150,000 x 0.
1.2 = $
5 = $3
square foot= 56 square feet x $40 per square foot
foot= $2240
$75 per hour =
is $150.3.
:$15 + $8 = $23
4. Add the
$23 = $58
- $58 = $2
-gill x $4
-gill is:2
has $3 leftover.
.2. Pablo always
$5 = $22
$12 = $27.
$24 by 2:$24 Ã· 2 = $
" books is:4 x $23 = $92
- $92 = $12
:13 x $8 = $
4 x $20

Cluster ID 177. this problem down step by step!
this problem down step by step
down step by step!
down step by step!
problem down step by step!We
down step by step!
this problem down step by step.
down step by step!
!Let x be the number of Oreos
number of Oreos James has
down step by step!
problem down step by step
parents and 2 teachers
down step by step!
on in 10 minutes
down step by step!
, James has to pay
spend the money ($50
problem down step by step!
this problem down step by step!
problem down step by step!
down step by step!
down step by step!
down step by step!
break it down step by step!
down step by step!
Let's break it down step by step:1. Robie bought
4 bags of chocolates left.
many apples Simone ate.
down step by step!
2: 90
's journey step by step
' expenses step by step
number of kangaroos
problem down step by step!
problem down step by step!
this problem down step by step
problem down step by step.
it down step by step
down step by step!
down step by step!
break it down step by step
problem step by step!
This means that he earned
earned a total of $
Let's break it down step by step:Tommy has
down step by step!
end of 2019.

G.1.3 Low-Cohesion or Noisy Clusters

Clusters such as **12**, **293** demonstrate relatively low silhouette scores, suggesting weaker structural alignment or noisier optimization signals. These clusters may contain sequences that deviate from common reasoning patterns or suffer from semantic ambiguity.

Cluster ID 12.

```
problem step by step:
down step by step!1.
down step by step!
this problem down step by step:1. The
3/4:96
down step by step!
down the problem step by step
down step by step!1.
problem down step by step.
/picture = 24
Carla has:500 ml
down step by step:1.
down step by step!1.
down step by step:1.
down step by step:1.
problem down step by step:1.
problem down step by step
down step by step!1.
break down the problem step by step:1. The
down step by step!1.
down step by step!1.
problem down step by step!
step by step:1
's break it down step by step:1. Robie bought
break down the information given:
step by step:1. The
down step by step!1.
down step by step:
problem down step by step:1.
problem down step by step:1
step by step:1.
problem down step by step
step by step:1
problem down step by step!1.
down step by step:1.
= 5 fish. However
However, one of them
problem down step by step:1. James sells
break down the problem step by step:1. The three friends pay an
down step by step!1.
step by step:1.
break down the problem step by step:1. The paint
break down the problem step by step:1.
step by step:1.
problem down step by step
```

Cluster ID 293.

```
, Ariana bought 14 carnations.#### 14
Cove.#### 20
baseball card.#### 49[SPECIAL_<|END_OF_TEXT|>]
22 + $18 + $24 = $64#### 64[SPECIAL_<|END_OF_TEXT|>]
:$150 - $140 = $10#### 10
amount:$500 - $74 = $426#### 426[SPECIAL_<|END_OF_TEXT|>]
+ 7 = 43 Oreos.####
50 hours to complete the trip to the hotel.#### 50[SPECIAL_<|END_OF_TEXT|>]
lbs#### 540
120 + 100 + 150 = 370 bottles#### 370
5.00#### 1035
James has to pay $150.#### 150
ister) + 180 (Henry) = 240#### 240
spent 70 minutes calling clients.#### 70
126 + 42 + 63 = 231#### 231
32 coconuts left after selling 10 of them.#### 32
in is 92.#### 92[SPECIAL_<|END_OF_TEXT|>]
rainwater in his tank at the end of the heavy rain.####
each day.####
x 2 = 12 leaves#### 12
= $420 - $110= $310#### 310[SPECIAL_<|END_OF_TEXT|>]
, Lex will have 42 apples left to eat raw.#### 42
show.#### 19
left.#### 42
= 18 years#### 18[SPECIAL_<|END_OF_TEXT|>]
Beatrix.#### 1423[SPECIAL_<|END_OF_TEXT|>]
#### 1.00[SPECIAL_<|END_OF_TEXT|>]
5 apples (Lauri) = 13 apples#### 13
- 4 = 8 fish.#### 8
's firewood = 13#### 13
final portfolio value after 2 years is $132.#### 132[SPECIAL_<|END_OF_TEXT|>]
```

0.375 pounds per square inch = 600 pounds#### 600
= 180 pages.#### 180
people#### 80
fruits/tree ÃĹ 8 trees = 960 fruits#### 960[SPECIAL_<|END_OF_TEXT|>]
Zane picked up 62 pounds of garbage.####
40 + 60 + 120 = 220 miles####
the same number of kangaroos as Kameron has now.####
picked 61 oranges.#### 61
540 - 104 = 436.#### 436
7 fish.#### 7
3 = 129#### 129[SPECIAL_<|END_OF_TEXT|>]
she now has:70 + 70 = 140 cents####
960,000#### 960
, Cooper needs 800 bricks to complete his fence.#### 800
books.#### 12
Marco.#### -2
+ 3 + 11 = 17 cars.#### 17
has played 5 gigs.#### 5

G.2 Per-Sample Token Trace Examples

Let's break this problem down step by step! 1. Ari ana bought a bunch of 40 flowers.
2. 2/5 of the flowers were roses. To find the number of roses, we can multiply 40 by 2/5:

40 ÃĹ (2/5) = 40 ÃĹ 0.4 = 16

So, Ariana bought 16 roses.
3. She also bought 10 tulips.
4. The rest of the flowers were carnations. To find the number of carnations, we subtract the number of roses and tulips from the total number of flowers:

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 = $120** 0
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 Ã· **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 initially:

10 (initial pieces) - 3 (pieces eaten on Friday) = 7

So, Mark ate 7 **pieces of fruit in the first four days of the week. #### 7**

==============================

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% of $125 = 0.25 x $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% of $156.25 = 0.20 x $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 Ã· **$125) x 100 = 100** %

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

==============================

{Oula and Tona work for a delivery service company, **supplying different parts of their state with milk products. They are paid $100 for each delivery made in a day** . In a particular month, Oula made 96 deliveries while Tona made 3/4 times as many deliveries as Oula. Calculate their difference in pay that month.}

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

96 deliveries x **$100 per delivery = $** 9600

Tona made 3/4 times as many deliveries as Oula. To find Tona's number of deliveries, multiply O **ula's number of deliveries by 3/4: 96 deliveries** x 3/4 = 72 deliveries

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

$9600 - $7200 = $2400 #### 2400

==============================

Let's break **down the problem step by step:** 1. The area of the street is given by the length of one side squared, which is 100 meters squared (100^2 = 10,000 square meters).
2. The area of the forest is three times the area of the street, so it is 3 x 10,000 = 30,000 square meters.
3. The number of trees per square meter in the forest is 4, so the total number of trees in the forest is the area of the forest multiplied by the number of trees per square meter:

Total number of trees = Area of forest x Trees per square meter
= 30,000 square meters x 4 trees/square meter
= 120,000 trees #### 120,000

==============================

Let's break down the problem step by step:

1. The class raised $50 initially.
2. Each of the 20 students contributed $5 each, so the total amount contributed by the students is:
20 students x **$5 per student = $100 3. The total amount** in the class fund is the initial amount raised ($50) plus the amount contributed by the students ($100):
$50 **+ $100 = $150** 4. The cost of the trip is $7 per student, and there are 20 students. So, the total cost of the trip is:
20 students x $7 **per student = $140 5** . After paying for the trip, **the amount left in the class** fund is the total amount in the fund minus the cost of the trip **: $150 - $140 = $10 #### 10**

==============================

Let's break down the cost of each item:

* 6 kilos of potatoes for **$2 per kilo: 6 x $2 = $12 * 9 kilos of tomatoes for $3 per kilo: 9 x $3 = $27 * 5 kil** os of cucumbers for **$4 per kilo:** 5 **x $4 = $20 * 3** kilos of bananas for $5 per kilo: 3 **x $5 = $15**

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

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

==============================

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

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

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

x + (4x + 7) = 52

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 x 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 = $** 103 **5.00 #### 1035**

==============================

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 x 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 x 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 x** 4 = 180

So, Henry is 180 years old.

Now, we can find the combined **age of the siblings by adding** their ages:

15 (Aaron) + 45 (s **ister) + 180 (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 fish/day ÃĹ 30 days = 2 ÃĹ 30 = 60 fish

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

2 fillets/fish

3. Since Luke catches 60 fish, the total number of fillets he gets is:

60 fish **ÃĹ 2 fillets/fish = 120 fillets #### 120**

==============================

Let's break down the problem step by step:

1. Anderson adds 3 sprigs of mint to every batch of mud.
2. For every sprig of mint, he adds 2 green tea leaves.
3. So, the total number of green tea leaves added to a batch of mud is 3 sprigs x 2 leaves/sprig = 6 leaves.
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.

To find the new number of green tea leaves, **we can multiply the original number by 2: 6 leaves x 2 = 12 leaves #### 12**

==============================

Let's break down the problem step by step:

1. Vikki's weekly earnings are calculated by multiplying her hourly pay rate by the number of hours she worked:

Weekly earnings = Hourly pay rate x Number of hours
= $10 x 42
= $420

2. 20% of the weekly earnings is **deducted as tax: Tax =** 20 **% of $420 = 0.20 x $420 = $84 3** . 5% of the weekly earnings is deducted as insurance cover:

Insurance cover **= 5% of $420 = 0.05 x $420 = $21 4** . $5 is deducted for union dues:

Union dues = $5

5. To find the total amount deducted, add the tax, insurance cover, and union dues:

Total deductions = Tax + Insurance cover + Union dues
= $84 + $21 + $5 = $110 6 . To find the amount Vikki takes home, subtract the total deductions from her weekly earnings:

Amount taken home = Weekly earnings - Total deductions
= $420 - $110 = $310 #### 310

==============================

Let's start by counting the number of legs of each member of Javier's household:

* Javier: 2 legs
* Wife: 2 legs
* Children (3): 3 x 2 = 6 legs
* Dogs (2): 2 x 4 = 8 legs
* Cat: 4 legs

Now, **let's add up the total** number of legs:

2 (Javier) + 2 (wife) + 6 (children) + 8 (dogs **) + 4 (cat) = 22 #### 22**

==============================

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 x 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

==============================

Let's break it down step by step: 1. Robie bought 3 bags of chocolates initially.
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 times the cost of making each cookie **. Since Joe sold 50** cookies and earned $60, the total selling price is $60. We can set up an equation to represent this:

50 x (cost of making each cookie) **x 1.2** = 60

To solve for the cost of making each cookie, we can divide both sides of **the equation by 60: (cost of** making each cookie) x 1.2 = 60 / 50

(cost of making each cookie) x **1.2 = 1.2** Now, we can divide both sides of the equation by 1.2 to solve for the cost of making each cookie:

(cost of making each cookie) = 1.2 / 1.2
(cost of making each cookie) = 1 So , the cookies cost $1 to make.

==============================

Let **'s break this problem down step by step: 1. The team** scored a total of 270 points.
2. 5 players averaged 50 points each, so the total points scored by these 5 players is:

5 players x 50 points/player = 250 points

3. Since the total points scored by the team **is 270, the remaining** points scored by the 4 other players (9 - 5 = 4) is:

270 - 250 = 20 points

4. Since there are 4 players, the average points scored by each of these players **is: 20 points / 4 players = 5 points/player #### 5**

==============================

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 x (1 + 0.15) = $80 x 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 x** (1 + 0.10) = $120 x 1.10 **= $132 So, the final portfolio value after 2 years is $132. #### 132**

==============================

Let's start by finding the area of the sandbox. Since it's a **square, we can use the formula: Area = side ÃĹ side =** 40 inches ÃĹ 40 inches
= 1600 square inches

Since a 30-pound bag of sand is enough to fill 80 square inches, we can find the amount of sand needed to fill 1 square inch:

30 pounds Ã· 80 square inches = 0.375 pounds per square inch

Now, we can find the total amount of sand needed to fill the entire sandbox:

1600 square inches ÃĹ **0.375 pounds per square inch = 600 pounds #### 600**

==============================

Let's break this problem down step by step! 1. Azure Valley has 60 skirts.
2. Seafoam Valley has two-thirds as many skirts as Azure Valley. To find the number of skirts in Seafoam Valley, we can multiply the number of skirts in Azure Valley by two-thirds:

60 x (2/3) = 60 x 0. **67 = 40 So** , 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 x (1/4) = 40 **x 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 ÃĹ 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 (initial** ly) = 60 people (still)

However, the number of people **in each car changes: 3 people/car** ÃĹ 20 cars = 60 people (initially)
4 people/car ÃĹ 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 x 4 = 62 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 x 4 = 62 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 oranges/day ÃĹ 2 days = 46 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 oranges - 46 oranges = 61 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** x $23600 = $2360
3. Total earnings: Basic salary + Commission = $125 **0 + $2360 =** $3610

Now, Robert allocates 20% of his total earnings to savings:

4. Savings: 20% of $3610 = 0.20 x $3610 = $722

The rest of the money goes to his monthly expenses:

5. Monthly expenses: Total earnings - **Savings = $3610 - $722 = $2888 #### 2888**

==============================

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

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

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

This pattern continues for the next three years:

Year 3: 4 + 2 = 6 fish, 6 - 1 = 5 fish
Year 4: 5 + 2 = 7 fish, 7 - 1 = 6 fish
Year 5: 6 + 2 = 8 fish, 8 - 1 = 7 fish

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

==============================

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

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

Let's break this **problem down step by step: 1. James sells the** house for 20% over market value. To find the selling price, we multiply the market value by 1.2 (since 20% is the same as 0.20, and 1 + 0.20 = 1.20).

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

Each person's share = Total revenue Ã· 4 = Selling price **Ã· 4 = $600,000 Ã· 4 = $150,000** 3. After taxes take away 10%, each person's share is reduced by 10%.

Each person's share after taxes = **Each person's share x (** 1 - 0.10) = $150,000 x (**1 - 0.10) = $150,000 x 0.90** = **$135,000 #### 135,** 000

============================

Let's break down the problem step by step: 1. The three friends pay an equal amount of money, which is $5 each.
2. Since they are paying an equal amount, the total amount they pay is 3 x $5 = $15.
3. This total amount is for 5 bags of chips, so to find the cost of each bag, we can divide the total amount by the number of bags:

$15 Ã· 5 = $3 #### 3

============================

Let's break it down step by step: On each of 7 Mondays, it rained 1.5 centimeters, so the total amount **of rain on Mondays is:** 7 x 1.5 = 10.5 centimeters

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

9 x 2.5 = 22.5 centimeters

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) ÃĹ 5 bricks (height) ÃĹ 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) ÃĹ 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 weekend, 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:

5 x 0.6 = 3 trout

The remaining fish are blue-gill:

5 - 3 = 2 blue-gill

3 trout and 2 blue-gill means he caught a total of 5 fish.

3. Trout earn $5 each, so the total amount earned from trout is:

3 trout x $5 = $15

2 blue-gill earn $4 each, so the total amount earned from blue **-gill is: 2 blue-gill x $4 =** $8

The total amount earned this Sunday is **: $15 + $8 = $23 4. Add the amount** earned this Sunday to the amount earned last weekend:

$35 + **$23 = $58** 5. Bucky needs $60 to buy the game, so he still needs:

$60 **- $58 = $2** #### 2

==============================

Here's the correct solution:

Let's break this **problem down step by step: 1. Marco** has $24 and gives Mary half of what he has. To find out how much he gives Mary, we can divide **$24 by 2: $24 Ã· 2 = $12**

So Marco gives **Mary $12. 2.** Mary now has $15 + **$12 = $27. 3 . Mary spends $5, so** we subtract $5 from $27:

$27 - **$5 = $22** 4. Marco still has $24, and Mary now has $22. To find out **how much more money Mary has than** Marco, we subtract Marco's amount from Mary's amount:

$22 - $24 = -$2

However, the problem states that Mary has more than Marco, so we need to find the correct difference. Since Mary has $22 and Marco has $24, Mary has $2 less than **Marco. #### -2**

==============================

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:

18 drums / 3 days = 6 drums per day

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 drums / 6 drums per day = 60 days #### 60**

==============================

{Let's break down the information given in the problem.}

* 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.}

* Aubrey is 2 years younger than Luka, so Aubrey's age on Luka's 4th birthday is 4 - 2 = 2 years old.
* Since Aubrey's 8th birthday is mentioned, that means Aubrey has aged 8 - 2 = 6 years since her 2nd birthday.
* Therefore, Aubrey's age on her 8th birthday is 2 + 6 = 8 years old.

{Now, let's find the age of Max.}

* Max was born on the day Luka turned 4 years old.
* Since Aubrey is 2 years younger than Luka, Aubrey was also 2 years old on the day Max was born.
* On Aubrey's 8th birthday, Max's age is the same as the number of years that have passed since his birth.
* The number of years that have passed since Max's birth is Aubrey's age on her 8th birthday minus her age on the day Max was born, which is 8 - 2 = 6 years.
* Therefore, Max is 6 years old on Aubrey's 8th birthday.

6

==============================

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

Tommy's neighbor, Jessie's older brother, has 5 more cars than Tommy and Jessie combined. To find out how many cars he has, we need to add Tommy's and Jessie's cars together and then **add 5. Tommy** and Jessie have 3 + 3 = 6 cars in total.
Adding 5 more cars to that, Jessie's older brother has 6 + 5 = 11 cars.

Now, let's count the total number of cars the three of them have:
Tommy has 3 cars.
Jessie has 3 cars.
Jessie's older brother has 11 cars.

In total, they have 3 **+ 3 + 11 = 17 cars. #### 17**

==============================

Let's break down the problem step by step:

1. The cost of the best-selling book "TOP" is $8, and 13 books were sold. So, the total revenue from "TOP" books is **: 13 x $8 = $104**

2. The cost of the least-selling book "ABC" is $23, and 4 books were sold. So, the total revenue from "ABC **" books is: 4 x $23 = $92** 3. To find the difference in the bookstore's earnings on these two books last week, we subtract the revenue from "ABC" books from the revenue from **"TOP" books: $104 - $92 = $12 #### 12**

==============================

Let's break this problem down step by step! Let the number of gigs played be x.

Each member earns $20 per gig, so the total earnings per gig is **4 x $20 =**

$80.

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

To solve for x, we can divide both sides of **the equation by 80:** x = 400 Ã· 80
x = 5

So, the band has played 5 gigs. #### 5