Render-of-Thought: Rendering Textual Chain-of-Thought as Images for Visual Latent Reasoning

Anonymous ACL submission

Abstract

Chain-of-Thought (CoT) prompting has achieved remarkable success in unlocking the reasoning capabilities of Large Language Models (LLMs). Although CoT prompting enhances reasoning, its verbosity imposes substantial computational overhead. Recent works often focus exclusively on outcome alignment and lack supervision on the intermediate reasoning process. These deficiencies obscure the analyzability of the latent reasoning chain. To address these challenges, we introduce **Render-of-Thought (RoT)**, the first framework to reify the reasoning chain by rendering textual steps into images, making the latent rationale explicit and traceable. Specifically, we leverage the vision encoders of existing Vision Language Models (VLMs) as semantic anchors to align the vision embeddings with the textual space. This design ensures **plug-and-play** implementation without incurring additional pre-training overhead. Extensive experiments on mathematical and logical reasoning benchmarks demonstrate that our method achieves 3-4× token compression and substantial inference acceleration compared to explicit CoT. Furthermore, it maintains competitive performance against other methods, validating the feasibility of this paradigm. Our code is available at https://anonymous.4open.science/r/RoT-1606

1 Introduction

As Large Language Models (LLMs) continue to scale, Chain-of-Thought (CoT) prompting (Wei et al., 2022; Xiang et al., 2025) has become a fundamental paradigm for unlocking complex reasoning capabilities. However, the inherent verbosity of CoT leads to prolonged inference latency and excessive memory consumption, hindering efficiency and scalability. Recent approaches address this challenge by explicitly compressing the CoT. Strategies range from token-level selection (Xia et al., 2025; Zhang et al., 2025; Han et al., 2025) to reinforcement learning methods that incentivize shorter inference paths via rewards (Aggarwal and Welleck, 2025; Luo et al., 2025; Wang et al., 2025). While valuable, these methods remain bound to sparse token representations. A more promising avenue involves reasoning within dense latent spaces. Early works such as Coconut (Hao et al., 2024) and CODI (Shen et al., 2025b) established the foundation for this paradigm, while CoLaR (Tan et al., 2025) further enhanced performance through dynamic latent compression mechanisms. However, recent efforts (Yue et al., 2025; Shen et al., 2025a; Liu et al., 2025) often employ complex architectures that compromise training stability. More critically, these methods typically focus exclusively on outcome alignment and lack supervision on the intermediate reasoning process. By compressing thoughts into opaque vectors without explicit constraints, they obscure the analyzability of the latent reasoning chain, making it difficult to trace the

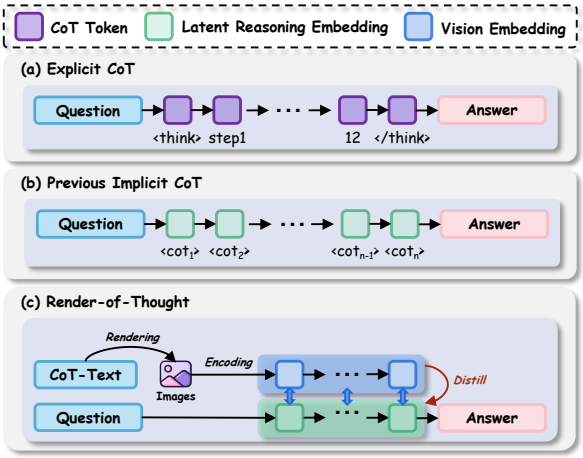


Figure 1: **Comparison of Reasoning Paradigms and Efficiency Analysis.** (a) Explicit CoT relies on verbose textual generation. (b) Implicit CoT compresses reasoning into latent space. (c) **Render-of-Thought** utilizes visual rendering as semantic anchors to structure the latent reasoning process.

model's rationale or diagnose logical errors.

To address these challenges, we propose **Render-of-Thought (RoT)** (Fig. 1), a framework that renders textual reasoning steps into images. This approach leverages the high information density of the visual modality to compress the reasoning process while keeping the rationale explicit. Crucially, unlike prior latent frameworks that require learning reasoning token from scratch, we utilize the frozen vision encoders of existing VLMs as semantic anchors. By grounding the LLM's latent states in the structured visual embeddings of rendered text, we provide a robust guide for the reasoning process.

Our pipeline implements a two-stage training strategy. Initially, we align the latent representations of the LLM with visual embeddings derived from rendered CoT. Subsequently, we enable the model to autoregressively generate the visual reasoning trajectory without requiring explicit text decoding. This design yields two key advantages: *1) Analyzability via Visualization,* which addresses the "black box" issue by making intermediate steps observable; and *2) Plug-and-Play Efficiency,* allowing standard VLMs to be upgraded via self-distillation without extra pre-training. Experiments on Qwen3-VL-4B-Instruct (Bai et al., 2025) show that Render-of-Thought achieves a 3-4× token compression rate and marked inference acceleration compared to explicit CoT, while maintaining competitive performance. Our contributions are summarized as follows:

- We introduce **Render-of-Thought**, the first framework to reify the reasoning chain by rendering textual steps into images, making latent reasoning explicit and traceable.
- We propose a mechanism using pre-trained vision encoders as **semantic anchors** to align vision embeddings with the textual space, ensuring a **plug-and-play** implementation without additional pre-training.
- Extensive experiments demonstrate that our method achieves 3-4× token compression and significant inference acceleration compared to explicit CoT, validating the feasibility and efficiency of the visual latent space as a reasoning carrier.

2 Related Work

Explicit Chain-of-Thought Reasoning. Chain-of-Thought (Wei et al., 2022) prompting has significantly enhanced the reasoning capabilities of LLMs. However, lengthy CoT chains raise generation costs, prompting methods to compress them. Methodologies such as (Xia et al., 2025; Zhang et al., 2025; Han et al., 2025) employ heuristic or learning-based strategies to select pivotal tokens while eliminating redundant content. Similarly, R1-Compress (Wang et al., 2025) introduces a chunk-based compression mechanism. Other approaches (Aggarwal and Welleck, 2025; Luo et al., 2025) leverage reinforcement learning to dynamically regulate reasoning length. C3oT (Kang et al., 2025) fine-tunes models using concise CoT datasets, while VeriThinker (Chen et al., 2025) enables the model to autonomously determine the necessity of continued reasoning.

Implicit Chain-of-Thought Reasoning. Implicit Chain-of-Thought techniques accelerate inference by encoding reasoning paths in a compact latent space. Pioneering works such as Coconut (Hao et al., 2024) and CODI (Shen et al., 2025b) established the foundation for continuous latent space compression. Building on this, SoftCoT (Xu et al., 2025) explores the use of Soft Tokens to represent intermediate reasoning, while CoLaR (Tan et al., 2025) investigates strategies for compressing reasoning chains within the latent space. Furthermore, recent frameworks like (Yue et al., 2025; Shen et al., 2025a; Liu et al., 2025) have proposed various architectural designs to support and enhance these implicit reasoning processes. Diverging from these approaches that primarily focus on linguistic or purely latent representations, we introduce a novel paradigm by reformulating reasoning steps as autoregressive visual embedding generation.

Text as Image for LLM. The paradigm of presenting textual information to LLMs via visual modalities has garnered increasing attention. Early investigations such as PixelWorld (Lyu et al., 2025) and From text to pixel (Lu et al., 2024). have demonstrated that Vision Language Models (VLMs) possess the capability to comprehend and reason over textual content embedded within images. More recent contributions including (Li et al., 2025b; Cheng et al., 2025) have established that rendering extensive textual inputs into visual formats can significantly scale context window capacities. However, existing "Text-as-Image" literature is mainly confined to input compression. To our knowledge, our framework is the first to apply visual rendering to compress the reasoning steps of VLMs.

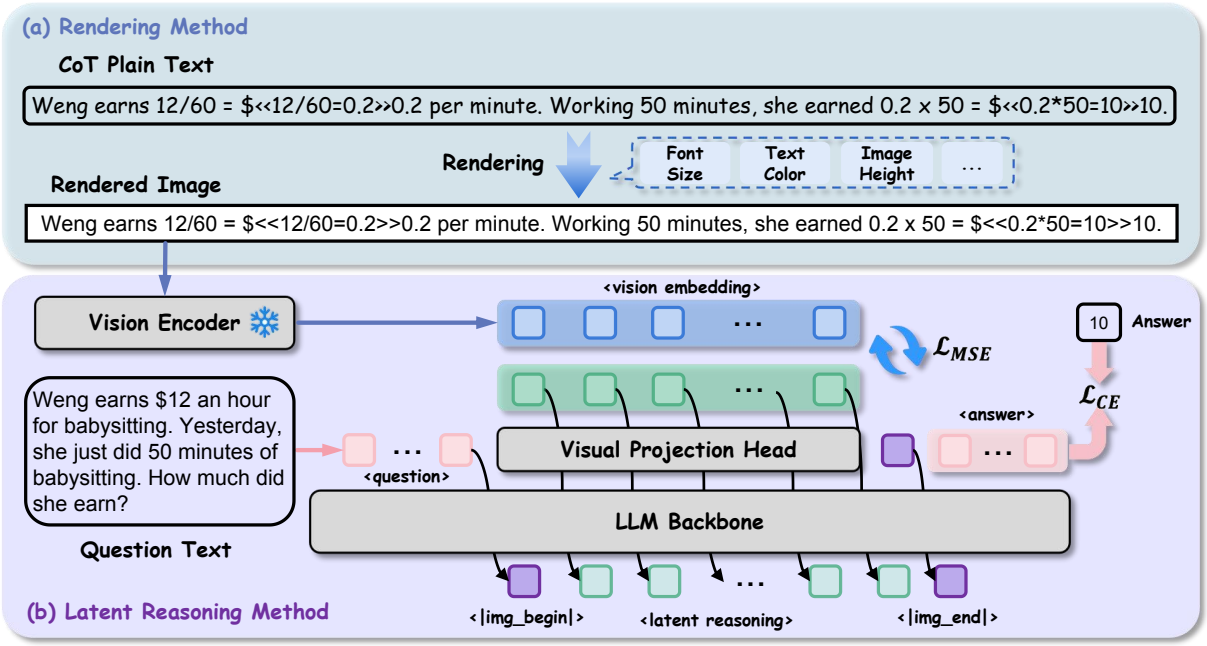


Figure 2: **Overview of the Render-of-Thought.** (a) Rendering Method transforms textual reasoning steps into compact single-line images. (b) Latent Reasoning Method aligns LLM-generated hidden states with visual features via a projection head, enabling the model to perform continuous reasoning within the visual latent space.

3 Method

3.1 Overview

Render-of-Thought introduces a novel paradigm for compressing textual CoT via optical rendering and visual knowledge distillation. Rather than processing verbose textual steps, this approach transforms intermediate reasoning paths into compact visual representations using a pre-trained vision encoder. As illustrated in Fig. 2, the framework comprises two primary components. First, textual CoT is converted into an image format using configurable rendering parameters, after which a visual encoder extracts features to serve as supervision targets. Second, the LLM backbone generates continuous latent reasoning tokens via the projection head, which are aligned with the visual features using Mean Squared Error (MSE) loss. The projection head is implemented as a two-layer MLP with SwiGLU (Shazeer, 2020) activation. During inference, rendering and visual encoding are eliminated, requiring only a forward pass through the trained LLM Backbone and Visual Projection Head.

3.2 CoT Rendering

The CoT rendering module transforms text into **single-line** images characterized by dynamic width and fixed height. This layout ensures that image patches are extracted in a strictly left-to-right manner, naturally aligning the visual sequence with the text order and eliminating spatial ambiguity. To accommodate varying text lengths while maintaining visual consistency, the image width is dynamically computed based on font size. In our experiments, the default configuration employs a 32 px `Image Height`, 4 px `Padding`, and 20 px `Font Size`. Additionally, images are rendered with black text on a white background. Visualization examples are provided in Appendix Sec. D.

3.3 Two-Stage Training Framework

To effectively translate the discrete reasoning capabilities of LLMs into a continuous visual latent space, we propose a progressive **two-stage** training paradigm. As shown in Fig. 3, This framework is designed to first align the semantic representations between the latent hidden states and visual modalities and subsequently enable the model to perform autoregressive latent reasoning.

3.3.1 Stage I: Visual Alignment

The first stage aligns the LLM's linguistic representations with the Vision Encoder's visual embeddings. While this alignment strategy mirrors the standard paradigm of Multimodal LLMs (MLLMs), it operates in the inverse direction. Unlike typical MLLMs that project visual features into the LLM's input space for understanding, we map the LLM's

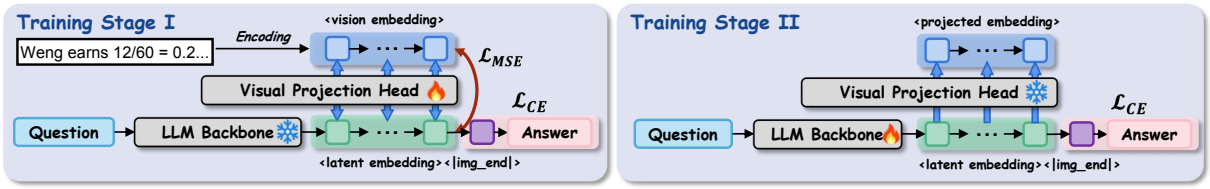


Figure 3: **Two-Stage Training Framework.** Stage I optimizes the projection head to map linguistic states to visual embeddings while freezing the backbone. Stage II fine-tunes the LLM to autoregressively generate the latent reasoning chain followed by the final answer.

hidden states to the visual embedding space at the output side. In this phase, we freeze the parameters of both the pre-trained LLM Backbone $\mathcal{M}$ and the Vision Encoder $\mathcal{V}$ to preserve their inherent semantic capabilities, exclusively optimizing a lightweight Visual Projection Head ϕ to perform this text-to-vision mapping.

Given an input question x and its corresponding CoT y_{cot}, we first render y_{cot} into an image using the rendering method described in Sec. 3.2. The Vision Encoder processes this image to extract target visual embeddings $\mathbf{V} = \{v_1, v_2, \ldots, v_K\}$, where $v_i \in \mathbb{R}^{d_v}$. The $<|\texttt{img\_begin}|>$ token is appended to the question to trigger visual reasoning. At reasoning step t, the latent reasoning embedding is derived as $\hat{v}_t = \phi(\mathcal{V}(\mathcal{M}(x, \text{<|img\_begin|>})))$. The alignment loss between $\hat{v}_t$ and the vision embeddings is defined as:

$$\mathcal{L}_{align} = \frac{1}{K} \sum_{t=1}^{K} \|\hat{v}_t - v_t\|_2^2. \tag{1}$$

Furthermore, to align the model with the proposed reasoning paradigm during Stage I, we simultaneously model the cross-entropy loss for both the $<|\texttt{img\_end}|>$ special token and the answer:

$$\mathcal{L}_{pred} = -\mathbb{E}_{(x,\hat{\mathbf{V}},y)\sim\mathcal{D}}[\log P(y_{\text{<|img\_end|>}} \mid x, \hat{\mathbf{V}}) + \sum_{j=1}^{T} \log P(y_j \mid x, \hat{\mathbf{V}}, y_{<j})], \tag{2}$$

where $\hat{\mathbf{V}}$ denotes the generated latent visual tokens, y represents the ground-truth of question x. The overall loss for Stage I can be formulated as:

$$\mathcal{L}_{\text{I}} = \mathcal{L}_{pred} + \lambda \mathcal{L}_{align}. \tag{3}$$

3.3.2 Stage II: Latent Supervised Fine-Tuning

Upon establishing the alignment between modalities, the Stage II focuses on empowering the LLM to autonomously generate the visual reasoning trajectory and the subsequent final answer. In this stage, we freeze the Vision Encoder and the now-aligned projection head ϕ. We fine-tune the LLM backbone parameters using LoRA (Hu et al., 2022) to adapt the model to the latent reasoning task.

The model generates a sequence of latent visual tokens $\hat{\mathbf{V}}$ followed by the special token $<|\texttt{img\_end}|>$ and final textual answer y_{ans}. Since the projection head is frozen, the LLM is implicitly constrained to generate hidden states that map to meaningful visual representations. The training objective is to maximize the likelihood of the correct answer and the special token, conditioned on the generated latent reasoning path. The training loss $\mathcal{L}_{\text{II}}$ in Stage II follows the same formulation as Eqn. 2.

Unlike the multi-task learning scheme in Sec. 3.3.1, we do not enforce an explicit visual regression loss in this stage. This allows the model to refine its internal reasoning process within the constraints of the aligned latent space, optimizing purely for the accuracy of the answer generation.

3.4 Inference and Decoding Strategies

The inference process requires the model to autonomously navigate the transition from the continuous latent reasoning space to the discrete textual solution space. We investigate two distinct decoding strategies to manage this modal shift.

Dynamic Termination via Special Tokens. This decoding mechanism relies on the intrinsic capability of the model to self-regulate the duration of its reasoning process. The reasoning phase concludes at the first time step T_{end} where the termination token achieves the highest probability:

$$T_{end} = \min\{t \mid \arg\max_{w\in\mathcal{T}} P(w|h_t) = w_{<|\texttt{img\_end}|>}\}, \tag{4}$$

where $\mathcal{T}$ denotes the token set and h_t represents the hidden state at time step t. The model initiates the decoding of the textual answer starting from the subsequent state $h_{T_{end}+1}$.

	GSM8k-Aug		GSM-Hard		SVAMP		MultiArith		Average		
	Pass@1	# L	Pass@1	# L	Pass@1	# L	Pass@1	# L	Pass@1	# L	Pass@1/# L
Qwen3-VL-2B-Instruct											
SFT-w/o CoT	$15.6_{\pm.31}$	$0.00_{\pm.00}$	$4.70_{\pm.22}$	$0.00_{\pm.00}$	$52.3_{\pm.34}$	$0.00_{\pm.00}$	$41.7_{\pm.28}$	$0.00_{\pm.00}$	28.6	0.00	-
SFT-CoT	$59.7_{\pm.35}$	$131.4_{\pm1.6}$	$33.1_{\pm.30}$	$207.2_{\pm1.7}$	$67.3_{\pm.27}$	$63.4_{\pm.83}$	$95.0_{\pm.36}$	$68.0_{\pm.73}$	63.8	117.5	0.54
RoT (Ours)	$23.3_{\pm.33}$	$32.0_{\pm.00}$	$8.64_{\pm.22}$	$32.0_{\pm.00}$	$53.7_{\pm.36}$	$32.0_{\pm.00}$	$62.2_{\pm.35}$	$32.0_{\pm.00}$	37.0	32.0	**1.16**
Qwen3-VL-4B-Instruct											
SFT-w/o CoT	$26.2_{\pm.25}$	$0.00_{\pm.00}$	$9.48_{\pm.13}$	$0.00_{\pm.00}$	$70.0_{\pm.31}$	$0.00_{\pm.00}$	$85.6_{\pm.39}$	$0.00_{\pm.00}$	47.8	0.00	-
SFT-CoT	$81.2_{\pm.35}$	$127.3_{\pm.96}$	$53.4_{\pm.34}$	$191.1_{\pm1.6}$	$84.3_{\pm.34}$	$55.9_{\pm.62}$	$98.3_{\pm.35}$	$59.1_{\pm.63}$	79.3	108.4	0.73
RoT (Ours)	$37.8_{\pm.30}$	$32.0_{\pm.00}$	$14.1_{\pm.15}$	$32.0_{\pm.00}$	$72.7_{\pm.44}$	$32.0_{\pm.00}$	$97.2_{\pm.43}$	$32.0_{\pm.00}$	55.4	32.0	**1.73**
LLaVa-V1.6-Mistral-7B											
SFT-w/o CoT	$10.8_{\pm.20}$	$0.00_{\pm.00}$	$2.27_{\pm.11}$	$0.00_{\pm.00}$	$40.7_{\pm.32}$	$0.00_{\pm.00}$	$52.8_{\pm.45}$	$0.00_{\pm.00}$	26.6	0.00	-
SFT-CoT	$38.6_{\pm.26}$	$151.3_{\pm1.6}$	$12.2_{\pm.11}$	$209.7_{\pm1.8}$	$56.3_{\pm.45}$	$79.9_{\pm.85}$	$86.1_{\pm.52}$	$82.3_{\pm.75}$	48.3	130.8	0.37
RoT (Ours)	$16.3_{\pm.22}$	$32.0_{\pm.00}$	$3.64_{\pm.12}$	$32.0_{\pm.00}$	$49.0_{\pm.38}$	$32.0_{\pm.00}$	$68.3_{\pm.48}$	$32.0_{\pm.00}$	34.3	32.0	**1.07**

Table 1: Experimental results on four grade-school reasoning datasets across three VLM architectures. Render-of-Thought achieves significant token compression compared to explicit CoT while maintaining competitive accuracy. The Pass@1/# L ratio measures efficiency, with higher values indicating better accuracy-to-token trade-offs.

Static Termination via Fixed Token Budgets. Despite the theoretical appeal of dynamic termination, empirical evidence suggests that self-regulated stopping can exhibit instability during the inference of continuous latent representations (Li et al., 2025a). To mitigate this, we constrain the latent chain of thought length to a fixed hyperparameter. Upon reaching this threshold, the $<|\texttt{img\_end}|>$ token is manually appended to trigger the transition from latent reasoning to text generation. We observe that decoding strategy selection significantly influences reasoning stability, as detailed in Sec. 4.3.

4 Experiments

4.1 Experiment Settings

Datasets and Tasks. Our method is primarily trained and evaluated on GSM8k-Aug-NL (Deng et al., 2023), an augmented version of the GSM8k (Cobbe et al., 2021) dataset containing approximately 385k training samples and over 1k test samples. We also assess the robustness of our model on three Out-of-Distribution (OOD) datasets: (1) GSM-Hard (Gao et al., 2023), which is a difficult variant of GSM8k with more than 1k test samples, as well as (2) SVAMP (Patel et al., 2021) and (3) MultiArith (Roy and Roth, 2015), which are simpler reasoning datasets. Additionally, we extend our experiments to the challenging MATH (Hendrycks et al., 2024) dataset, which encompasses diverse disciplines including algebra, calculus, statistics, geometry, linear algebra, and number theory, utilizing 7.5k training and 0.5k test samples. Our evaluation framework simultaneously measures accuracy (Pass@1) and computational efficiency (# L, denoting the average token length of the reasoning chain). All experiments are performed across five distinct random seeds, and we report the mean values for Pass@1 and # L alongside their 95% confidence intervals (CI).

Implementation Details. (1) Base Model: Unless otherwise specified, we utilize the pre-trained and frozen Qwen3-VL-2B/4B-Instruct (Bai et al., 2025) and LLaVa-V1.6-Mistral-7B (Liu et al., 2023) as our base models, incorporating LoRA modules (Hu et al., 2022) for efficient fine-tuning. The Visual Projection Head consists of a two-layer MLP based on the SwiGLU (Shazeer, 2020) activation function. For the Vision Encoder, we directly employ the native module from Qwen3-VL and keep it frozen. This strategy ensures alignment between the vision embeddings and the LLM backbone without the need for re-pre-training. (2) Training Epoch: All models undergo training for 3 epochs to ensure fair comparison. (3) Hyperparameter: Throughout Stage I and Stage II, we use the AdamW (Loshchilov and Hutter, 2017) optimizer with a fixed learning rate of 2e-5, a weight decay of 1e-2, and a batch size of 16. Specifically, Stage I involves 1 epoch of training, and Stage II involves 2 epochs. For inference, the temperature is set to 1.0 and top-p to 0.9. Please refer to Appendix Sec. A for additional implementation details.

4.2 Main Results

Performance on Low-Difficulty Tasks. Tab. 1 presents comprehensive results on four grade-school reasoning datasets across three VLM architectures. On Qwen3-VL-4B-Instruct, our method

	GSM8k-Aug		GSM-Hard		SVAMP		MultiArith		Average	
	Pass@1	# L	Pass@1	# L	Pass@1	# L	Pass@1	# L	Pass@1	# L
LLM Based: Qwen3-4B-Instruct										
iCoT	$13.5_{\pm.21}$	$0.00_{\pm.00}$	$4.09_{\pm.18}$	$0.00_{\pm.00}$	$36.9_{\pm.23}$	$0.00_{\pm.00}$	$49.2_{\pm.67}$	$0.00_{\pm.00}$	25.9	0.00
Coconut	$16.9_{\pm.26}$	$6.00_{\pm.00}$	$5.42_{\pm.28}$	$6.00_{\pm.00}$	$43.6_{\pm.53}$	$6.00_{\pm.00}$	$60.3_{\pm.65}$	$6.00_{\pm.00}$	31.6	6.00
CODI	$7.28_{\pm.46}$	$6.00_{\pm.00}$	$2.20_{\pm.22}$	$6.00_{\pm.00}$	$11.0_{\pm.63}$	$6.00_{\pm.00}$	$18.3_{\pm.75}$	$6.00_{\pm.00}$	9.70	6.00
CoLaR-2	**40.0**$_{\pm.19}$	$39.6_{\pm.12}$	$9.17_{\pm.05}$	$47.4_{\pm.15}$	$57.7_{\pm.23}$	$19.2_{\pm.06}$	$82.2_{\pm.11}$	$21.1_{\pm.08}$	47.3	31.8
CoLaR-5	$18.6_{\pm.13}$	$16.4_{\pm.08}$	$5.69_{\pm.05}$	$22.8_{\pm.10}$	$48.0_{\pm.42}$	$7.46_{\pm.03}$	$63.3_{\pm.35}$	$7.73_{\pm.01}$	33.9	13.6
VLM Based: Qwen3-VL-4B-Instruct										
RoT (Ours)	$37.8_{\pm.30}$	$32.0_{\pm.00}$	**14.1**$_{\pm.15}$	$32.0_{\pm.00}$	**72.7**$_{\pm.44}$	$32.0_{\pm.00}$	**97.2**$_{\pm.43}$	$32.0_{\pm.00}$	**55.4**	32.0
- w/o Stage I	$24.8_{\pm.28}$	$32.0_{\pm.00}$	$7.20_{\pm.12}$	$32.0_{\pm.00}$	$58.3_{\pm.38}$	$32.0_{\pm.00}$	$78.5_{\pm.41}$	$32.0_{\pm.00}$	42.2	32.0
- w/o Stage II	$29.9_{\pm.28}$	$32.0_{\pm.00}$	$9.48_{\pm.12}$	$32.0_{\pm.00}$	$63.7_{\pm.41}$	$32.0_{\pm.00}$	$87.8_{\pm.39}$	$32.0_{\pm.00}$	47.7	32.0

Table 2: Comparison with LLM based latent reasoning methods on four grade-school reasoning datasets. All LLM based baselines use Qwen3-4B-Instruct as the base model. Render-of-Thought employs Qwen3-VL-4B-Instruct. Best and second-best results are highlighted with "" and "", respectively.

achieves 55.4% average accuracy with only 32 latent tokens, compared to 79.3% accuracy with 108.4 tokens for explicit CoT. Notably, on simpler tasks such as MultiArith, Render-of-Thought achieves near-parity performance with a 1.8× reduction in token consumption. The consistent improvements across all three model architectures validate the generalizability of our approach.

Compared with LLM based Latent Reasoning. Tab. 2 compares Render-of-Thought against LLM-based baselines across four grade-school level reasoning datasets. To ensure fair comparison, all baselines are reproduced using Qwen3-4B-Instruct (Yang et al., 2025) as the base model. Render-of-Thought achieves an average accuracy of 55.4%, outperforming the best LLM based method, CoLaR-2, by 8.1%. While CoLaR-2 yields slightly higher accuracy on GSM8k-Aug, our approach demonstrates superior robustness in out-of-domain generalization. We attribute this to the rich semantic representations from the pre-trained visual encoder, which provide more informative supervision signals than the latent spaces learned from scratch in LLM based methods.

Performance on High-Difficulty Tasks. To assess scalability on more challenging reasoning tasks, we evaluate our method on the MATH dataset. As detailed in Tab. 3, we employ three distinct model architectures to demonstrate robustness. On Qwen3-VL-4B-Instruct, explicit CoT method achieves 55.8% accuracy but requires an average of 291.5 tokens for the reasoning chain. In contrast, Render-of-Thought achieves 33.2% Pass@1 using only 64 latent tokens, surpassing the w/o CoT baseline of 29.4%. Furthermore, meaningful improvements

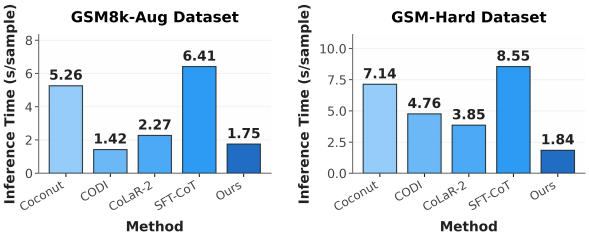


Figure 4: **Inference Time Comparison.** We evaluate the average inference time (seconds per sample) on GSM8k-Aug and GSM-Hard datasets using Qwen3-4B-Instruct/Qwen3-VL-4B-Instruct.

over the w/o CoT baseline on the smaller Qwen3-VL-2B-Instruct validate the generalizability of our approach across model scales.

Inference Time Comparison. We further analyze computational efficiency by comparing the average per-sample inference time on the GSM8k-Aug and GSM-Hard datasets. To ensure fair comparison, all experiments are conducted on a single NVIDIA H20 GPU with a batch size of 1. As shown in Fig. 4, Render-of-Thought demonstrates significant efficiency gains over explicit CoT. On the challenging GSM-Hard dataset, inference time decreases notably from 8.55s to 1.84s. This substantial reduction in latency stems from compressing lengthy textual thoughts into compact sequences of visual latent embeddings. Moreover, our method surpasses several latent reasoning baselines in speed, validating the efficiency of the visual latent space.

4.3 Ablation Study & Analysis

Effectiveness of Two-Stage Training. To assess the contribution of our progressive training strategy, we ablate each stage independently. Results in Tab. 2 and Tab. 3 confirm that both stages are

	Qwen3-VL-2B-Instruct		Qwen3-VL-4B-Instruct		LLaVa-V1.6-Mistral-7B	
	Pass@1	#L	Pass@1	#L	Pass@1	#L
SFT-w/o CoT	$20.8_{\pm.21}$	$0.00_{\pm.00}$	$29.4_{\pm.34}$	$0.00_{\pm.00}$	$11.2_{\pm.30}$	$0.00_{\pm.00}$
SFT-CoT	$29.2_{\pm.29}$	$324.5_{\pm2.6}$	$55.8_{\pm.36}$	$291.5_{\pm1.9}$	$13.8_{\pm.33}$	$200.9_{\pm2.1}$
RoT (Ours)	$24.0_{\pm.22}$	$64.0_{\pm.00}$	$33.2_{\pm.37}$	$64.0_{\pm.00}$	$12.4_{\pm.25}$	$64.0_{\pm.00}$
- w/o Stage I	$15.8_{\pm.19}$	$64.0_{\pm.00}$	$22.2_{\pm.34}$	$64.0_{\pm.00}$	$9.40_{\pm.26}$	$64.0_{\pm.00}$
- w/o Stage II	$19.2_{\pm.21}$	$64.0_{\pm.00}$	$26.2_{\pm.38}$	$64.0_{\pm.00}$	$10.8_{\pm.28}$	$64.0_{\pm.00}$

Table 3: Experimental results on the challenging MATH dataset across three VLM architectures. Render-of-Thought achieves significant token compression compared to explicit CoT while maintaining competitive accuracy.

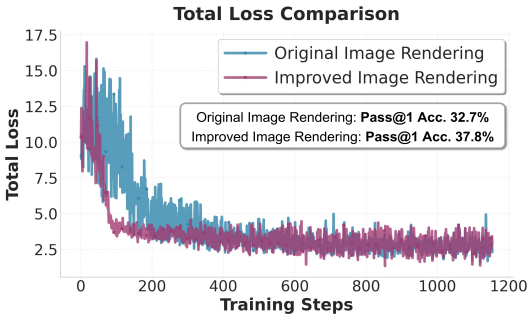


Figure 5: **Impact of Rendering Strategies on Training Convergence.** The improved **single-line rendering** demonstrates superior stability and speed compared to the **fixed-size square** approach.

essential for optimal reasoning performance. Removing Stage I causes accuracy on GSM8k-Aug to drop from 37.8% to 24.8%, indicating that visual alignment is vital for structuring the latent space and preventing representation collapse during complex tasks. Similarly, excluding Stage II leads to a significant performance decline because the model struggles to navigate the continuous latent space toward the final answer. This necessity is further evidenced on the MATH benchmark where performance falls from 33.2% to 26.2% in the absence of Stage II. These findings demonstrate that our two-stage framework establishes a robust foundation for compressing verbose textual chains into efficient visual latent representations.

Impact of Visual Rendering Configurations. The design of visual rendering configurations significantly influences the effectiveness of latent reasoning. We compare two rendering paradigms on GSM8k-Aug dataset: the original approach using fixed-size square images (1024 px × 1024 px) with multi-line text wrapping, and our improved single-line rendering with dynamic width and fixed 32 px height. As illustrated in Fig. 5, the single-line configuration demonstrates superior convergence. Quantitatively, the single-line rendering achieves 37.8% Pass@1 accuracy on GSM8k-Aug, outperforming the fixed-size square baseline. This improvement stems from several key design choices. First, dynamic width eliminates large blank regions that would otherwise produce meaningless embeddings after visual encoding, preventing the model from learning spurious patterns. Second, preserving complete text content without truncation ensures no information loss during the rendering process. Finally, the single-line format is more compatible with sequential modeling paradigms, as it naturally represents reasoning steps as a continuous visual sequence rather than discrete multi-line blocks. More ablation study results regarding the visual rendering configuration are provided in Appendix Sec. C.

Decoding Strategies	GSM8k-Aug Pass@1	MATH Pass@1
Special Tokens	3.87	2.20
Fixed Token Budgets	-	-
- 8 tokens	1.89	0.80
- 16 tokens	11.4	8.40
- 32 tokens	**37.8**	30.4
- 64 tokens	36.2	**33.2**
- 128 tokens	34.6	33.0
- 256 tokens	32.1	31.2

Table 4: Comparison of decoding strategies on GSM8k-Aug and MATH datasets using Qwen3-VL-4B-Instruct. Fixed token budgets consistently outperform dynamic termination via special tokens.

Comparison of Inference Decoding Strategies. We evaluate two decoding strategies on Qwen3-VL-4B-Instruct across GSM8k-Aug and MATH datasets, as detailed in Tab. 4. The dynamic termination strategy using $<|\texttt{img\_end}|>$ yields significantly lower performance compared to fixed token budgets. This performance gap stems from the inherent instability of self-regulated stopping in continuous latent spaces. When generating latent

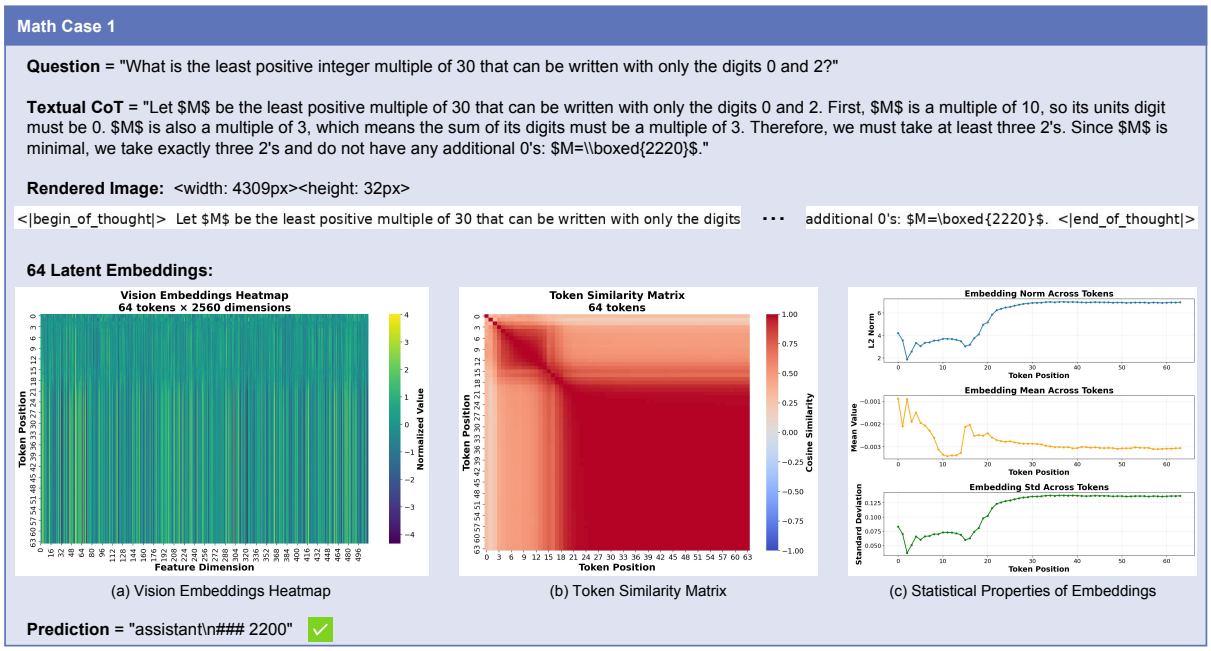


Figure 6: **Characterizations of Latent Visual Tokens**. We present a case from the MATH dataset. The generated latent embeddings are analyzed via (a) Vision Embeddings Heatmap, (b) Token Similarity Matrix, and (c) Statistical Properties, demonstrating the structured semantic encoding within the continuous visual latent space.

reasoning embeddings, the hidden states may not consistently produce high-confidence predictions for the termination token, leading to premature or delayed transitions that disrupt the reasoning flow. In contrast, fixed token budgets provide stable, predictable termination points that align better with the sequential nature of visual latent reasoning.

The optimal token budget varies across datasets, reflecting differences in task complexity and reasoning depth. On GSM8k-Aug, 32 tokens achieve the best performance, while MATH requires 64 tokens to reach peak accuracy. We speculate that this discrepancy arises because the MATH dataset is more challenging and necessitates longer reasoning chains. An insufficient token budget severely constrains the model's ability to encode complex reasoning trajectories, resulting in significant performance degradation. Conversely, an excessive budget introduces redundancy and potential noise. These findings demonstrate that task-specific token budget calibration is crucial for balancing reasoning completeness with computational efficiency.

4.4 Discussion of Latent Visual Tokens.

We observed a phenomenon in the latent tokens generated by Render-of-Thought. As illustrated in Fig. 6, the output tokens tend to become increasingly homogeneous after a certain position in the sequence. Specifically, the values in the token similarity matrix approach 1.0, the feature activation heatmaps become nearly identical, and the statistical properties of the embeddings tend to stabilize. This suggests that the model effectively encodes the core reasoning logic in the initial phase, after which the latent states enter a saturation plateau. These subsequent high-similarity tokens likely serve to maintain the semantic context required for decoding the final answer, rather than introducing new reasoning steps or feature shifts. More visualization results are available in Appendix Sec. D.

5 Conclusion

We introduce Render-of-Thought, the first framework to compress Chain-of-Thought reasoning by rendering textual steps into visual latent representations. By leveraging pre-trained vision encoders as semantic anchors, our method aims to address the analyzability issues of prior latent reasoning approaches. Our two-stage training strategy effectively bridges the modality gap, enabling plug-and-play implementation within standard VLM architectures. Extensive experiments demonstrate 3-4× token compression and significant inference acceleration compared to explicit CoT, while maintaining competitive accuracy across mathematical and logical benchmarks. This work establishes visual rendering as a viable paradigm for efficient and analyzable latent reasoning.

Limitations

While Render-of-Thought demonstrates promising results, several limitations warrant future investigation. First, our evaluation is currently limited to English-language mathematical and logical reasoning tasks. The method's effectiveness on other reasoning domains, such as commonsense reasoning or causal inference, as well as its applicability to non-English languages, remains unexplored. Future work could extend the evaluation to diverse reasoning benchmarks and multilingual settings to assess broader generalizability.

Second, the optimal latent token budget requires task-specific calibration, as evidenced by the different optimal values for GSM8k-Aug (32 tokens) and MATH (64 tokens). This manual tuning process may not be feasible for novel applications where task complexity is unknown a priori. Potential solutions include developing adaptive token budget mechanisms that dynamically adjust based on problem difficulty or learning task-specific budget predictors from problem characteristics. Furthermore, Render-of-Thought encounters a phenomenon similar to that described in (Li et al., 2025a), where dynamic termination via special tokens exhibits instability in continuous latent spaces. We also intend to address this issue in future work.

Finally, the training process incurs more computational overhead from rendering textual CoT into images and processing them through the vision encoder, though this cost is eliminated during inference. Future work could investigate more efficient rendering strategies or explore caching mechanisms to reduce training time for large-scale deployments.

Ethics Statement

This work utilizes publicly available datasets for mathematical and logical reasoning, including GSM8k-Aug (Deng et al., 2023), GSM8k (Cobbe et al., 2021), GSM-Hard (Gao et al., 2023), SVAMP (Patel et al., 2021), MultiArith (Roy and Roth, 2015), and MATH (Hendrycks et al., 2024). These datasets contain grade-school and challenging mathematical problems that do not involve personal information, sensitive content, or potentially harmful material. All datasets are used in accordance with their original licenses and intended research purposes. Regarding data privacy and protection, we conducted a specific assessment to verify whether the data contains information that names or uniquely identifies individual people. Given that the datasets (e.g., GSM8k, MATH) consist of standard mathematical word problems where names are generic and fictional, we determined that the data does not refer to real-world individuals. Consequently, no additional anonymization or de-identification steps were required beyond the standard usage of these public benchmarks.

We employ open-source vision-language models including Qwen3-VL-2B/4B-Instruct (Bai et al., 2025) and LLaVa-V1.6-Mistral-7B (Liu et al., 2023), accessed through standard model repositories such as Hugging Face Hub (Wolf et al., 2020). All models are used under their respective licenses, which permit research use. We have reviewed and complied with all terms of use for these models and their associated training data.

Our training pipeline involves rendering textual Chain-of-Thought annotations into images, which are then processed through pre-trained vision encoders. The rendered images contain only mathematical reasoning steps and problem solutions, without any personal data or offensive content. We have manually inspected a sample of rendered images to ensure they do not contain inappropriate material, and our observations confirm that all rendered content consists solely of mathematical expressions and reasoning steps.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Zigeng Chen, Xinyin Ma, Gongfan Fang, Ruonan Yu, and Xinchao Wang. 2025. Verithinker: Learning to verify makes reasoning model efficient. *arXiv preprint arXiv:2505.17941*.

Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, and 1 others. 2025. Glyph: Scaling context windows via visual-text compression. *arXiv preprint arXiv:2510.17800*.

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*.

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*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

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

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*.

D Hendrycks. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2024. Measuring mathematical problem solving with the math dataset, 2021. *URL https://arxiv. org/abs/2103.03874*, 2.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. 2025. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320.

Bangzheng Li, Ximeng Sun, Jiang Liu, Ze Wang, Jialian Wu, Xiaodong Yu, Hao Chen, Emad Barsoum, Muhao Chen, and Zicheng Liu. 2025a. Latent visual reasoning. *arXiv preprint arXiv:2509.24251*.

Yanhong Li, Zixuan Lan, and Jiawei Zhou. 2025b. Text or pixels? evaluating efficiency and understanding of llms with visual text inputs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10564–10578.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Jiayu Liu, Zhenya Huang, Anya Sims, Enhong Chen, Yee Whye Teh, and Ning Miao. 2025. Marcos: Deep thinking by markov chain of continuous thoughts. *arXiv preprint arXiv:2509.25020*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yujie Lu, Xiujun Li, Tsu-Jui Fu, Miguel Eckstein, and William Yang Wang. 2024. From text to pixel: Advancing long-context understanding in mllms. *arXiv preprint arXiv:2405.14213*.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*.

Zhiheng Lyu, Xueguang Ma, and Wenhu Chen. 2025. Pixelworld: Towards perceiving everything as pixels. *arXiv preprint arXiv:2501.19339*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1743–1752.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Xuan Shen, Yizhou Wang, Xiangxi Shi, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. 2025a. Efficient reasoning with hidden thinking. *arXiv preprint arXiv:2501.19201*.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025b. Codi: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*.

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Jian Luan, and Ruihua Song. 2025. Think silently, think fast: Dynamic latent compression of llm reasoning chains. *arXiv preprint arXiv:2505.16552*.

Yibo Wang, Haotian Luo, Huanjin Yao, Tiansheng Huang, Haiying He, Rui Liu, Naiqiang Tan, Jiaxing Huang, Xiaochun Cao, Dacheng Tao, and 1 others. 2025. R1-compress: Long chain-of-thought compression via chunk compression and search. *arXiv preprint arXiv:2505.16838*.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*.

Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, and 1 others. 2025. Towards system 2 reasoning in llms: Learning how to think with meta chain-of-thought. *arXiv preprint arXiv:2501.04682*.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025. Softcot: Soft chain-of-thought for efficient reasoning with llms. *arXiv preprint arXiv:2502.12134*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Zhenrui Yue, Bowen Jin, Huimin Zeng, Honglei Zhuang, Zhen Qin, Jinsung Yoon, Lanyu Shang, Jiawei Han, and Dong Wang. 2025. Hybrid latent reasoning via reinforcement learning. *arXiv preprint arXiv:2505.18454*.

Jintian Zhang, Yuqi Zhu, Mengshu Sun, Yujie Luo, Shuofei Qiao, Lun Du, Da Zheng, Huajun Chen, and Ningyu Zhang. 2025. Lightthinker: Thinking step-by-step compression. *arXiv preprint arXiv:2502.15589*.

Render-of-Thought: Rendering Textual Chain-of-Thought as Images for Visual Latent Reasoning

Appendix

Content

This Appendix contains the following parts:

- **More Implementation Details**. We provide detailed implementation specifications including model hyperparameters, training hyperparameters, and dataset information.
- **Ablation Study on Visual Projection Head**. We conduct ablation studies on the activation function and hidden dimension of the Visual Projection Head, demonstrating the optimal architectural choices.
- **Ablation Study on Visual Rendering Configurations**. We investigate the influence of rendering configurations, identifying the most effective visual configuration.
- **Case Study**. We visualize the representations of the latent reasoning embeddings, including heatmaps, similarity matrices and statistical properties, to qualitatively analyze the model's reasoning process across different benchmarks.

A More Implementation Details

Model hyperparameters. In our experiments, we employ frozen Qwen3-VL-2B/4B-Instruct, LLaVa-V1.6-Mistral-7B, and Qwen3-4B-Instruct as the LLM backbones, utilizing LoRA modules for fine-tuning. Across all experiments, the LoRA modules are configured with $\alpha = 32$, $r = 16$, and a dropout rate of 0.05. Our method introduces a Visual Projection Head, implemented as a two-layer MLP based on SwiGLU, with the hidden layer dimension set to $d = 4096$.

Training hyperparameters. We utilize the AdamW optimizer with a weight decay of 1e-2 for all experiments. The learning rate is set to 2e-5 for both training stages. For each training stage, we conduct experiments on two NVIDIA H20 GPUs with DeepSpeed (Rasley et al., 2020) configured to Stage 2, using a total batch size of 16. During Stage I training, the weight λ of the alignment loss $\mathcal{L}_{align}$ is set to 10.0. To ensure reproducibility, we fix the random seeds for all libraries (Python, CUDA, PyTorch, and NumPy) to 0 during the training process.

Additionally, Render-of-Thought involves the special tokens $<|\texttt{img\_begin}|>$ and $<|\texttt{img\_end}|>$ during training. For each special token, we first generate a random vector, normalize it to a unit vector, scale it to a norm of $\sqrt{h_d}$, and finally write it into the corresponding position in the embedding table, where h_d represents the hidden dimension of the LLM Backbone. The use of the $\sqrt{h_d}$ norm is intended to match the typical norm of pre-trained embeddings, ensuring numerical compatibility and training stability. The random initialization draws reference from LVR (Li et al., 2025a), which facilitates the model in distinguishing between the latent reasoning state and the normal semantic state.

Dataset information. Render-of-Thought is evaluated on five datasets: GSM8K-Aug, GSM8K-Hard, SVAMP, MultiArith, and MATH. Since the original MATH dataset does not provide an official validation set, we follow the protocol of CoLaR (Tan et al., 2025) by randomly shuffling the training set and allocating 10% of the samples for validation.

B Ablation Study on Visual Projection Head

The Visual Projection Head bridges linguistic and visual modalities by mapping LLM hidden states to the visual embedding space. To optimize its architecture, we conduct ablation studies on the activation function and hidden dimension. Regarding activation functions, we compare ReLU, GELU(Hendrycks, 2016), and SwiGLU(Shazeer, 2020). As shown in Tab. 5, SwiGLU consistently outperforms the others. We attribute this to its gated

Configuration	GSM8k-Aug Pass@1	MATH Pass@1
Activation Function (Hidden Dim = 4096)		
ReLU	33.2	28.6
GELU	35.1	30.8
SwiGLU	**37.8**	**33.2**
Hidden Dimension (Activation = SwiGLU)		
2048	34.5	30.1
4096	**37.8**	**33.2**

Table 5: Ablation study on Visual Projection Head configurations using Qwen3-VL-4B-Instruct.

mechanism, which enhances feature expressiveness and gradient flow during alignment.

For the hidden dimension, we find that the default setting of 4096 offers an optimal balance between capacity and efficiency. Reducing the dimension to 2048 leads to noticeable performance degradation, particularly on the challenging MATH dataset, underscoring the necessity of sufficient capacity to capture complex reasoning patterns.

C Ablation Study on Visual Rendering Configurations

To investigate how rendering hyperparameters influence the semantic encoding capability of the vision encoder, we conduct ablation studies on image height, font size, and padding. As detailed in Tab. 6, the configuration of 32 px height, 20 px font size, and 4 px padding achieves optimal accuracy. We observe that image height is a critical factor, reducing it to 16 px results in significant performance degradation. This is likely because insufficient vertical resolution blurs character details, impairing the vision encoder's ability to extract precise textual semantics. Increasing the height to 64 px does not consistently improve performance, suggesting that 32 px generally provide adequate resolution for character legibility without introducing excessive background noise. Regarding font size, 20 px offers the best performance. Deviating from this optimal size negatively impacts the results, potentially by distorting character stroke features or altering the spatial density to which the pre-trained encoder is adapted. Finally, appropriate padding (4 px) proves necessary to avoid boundary artifacts, ensuring that character features are fully preserved within the visual receptive field.

Configuration	GSM8k-Aug Pass@1	MATH Pass@1
Image Height (Font Size = 20, Padding = 4)		
16 px	34.2	29.8
32 px	**37.8**	**33.2**
64 px	37.1	33.5
Font Size (Height = 32, Padding = 4)		
16 px	35.6	31.4
20 px	**37.8**	**33.2**
24 px	36.9	32.7
Padding (Height = 32, Font Size = 20)		
0 px	37.2	32.9
4 px	**37.8**	**33.2**
8 px	37.5	33.0

Table 6: Ablation study on visual rendering configurations using Qwen3-VL-4B-Instruct.

D Case Study

In this section, we present a qualitative analysis of the Render-of-Thought framework by visualizing the latent reasoning process across various benchmarks. To provide deeper insights into the internal representations, we visualize three key metrics for the generated latent tokens including vision embeddings heatmaps, token similarity matrices, and statistical properties of the embeddings. These visualizations allow us to trace the semantic progression of the model within the continuous latent space.

We first examine successful reasoning examples on the GSM8k-Aug dataset as illustrated in Fig. 7. In these instances, the model compresses the reasoning path into a fixed budget of 32 latent embeddings. The token similarity matrices exhibit a distinct diagonal pattern with local coherence, suggesting that the model maintains a sequential train of thought where adjacent tokens are semantically related but distinct enough to carry new information. Furthermore, the heatmaps display sparse and structured activation patterns, indicating that the model effectively encodes specific semantic features from the visual supervision into the latent space. The successful decoding of the final answers demonstrates that 32 latent embeddings are sufficient to capture the reasoning logic for standard grade-school math problems.

Fig. 8 extends our analysis to the more challenging MATH dataset which involves complex symbols and longer reasoning chains requiring a 64-token budget. As seen in the first two cases,

the rendered images contain complex LaTeX expressions that the model successfully aligns with its latent states. The similarity matrices here show a block-diagonal structure that potentially corresponds to different stages of solving the problem, such as understanding the problem and formulating equations.

Finally, Fig. 9 highlights failure cases across Out-of-Distribution datasets including GSM-Hard, SVAMP, and MultiArith. A common pattern observed in these failure cases is the presence of large and high-similarity blocks in the similarity matrices. A common pattern observed in these failure cases is the presence of large and highly similar blocks within the similarity matrices. Unlike successful cases, failure cases typically display relatively disordered similarity patterns, implying that the model generates repetitive or indistinguishable latent tokens that fail to effectively advance the reasoning process. Additionally, we observe that failure cases tend to exhibit relatively larger variance. We attribute this to the model's inability to maintain high-confidence representations when encountering unfamiliar problem structures, ultimately leading to incorrect decoding.

GSM8k-Aug Case 1

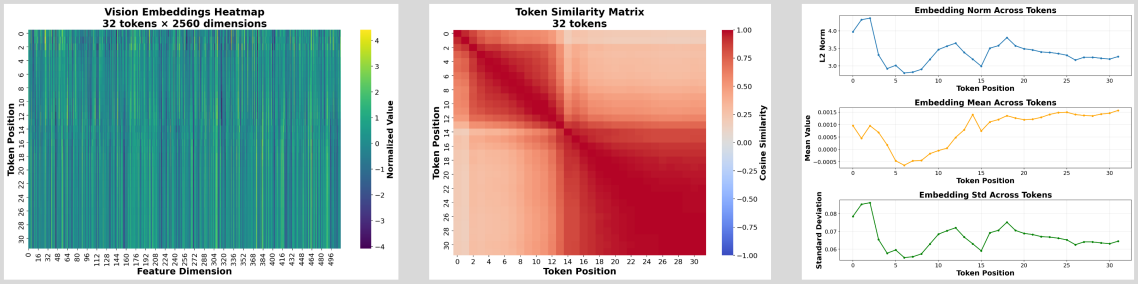
Question = "Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?"

Textual CoT = "Janet sells 16 - 3 - 4 = <<16-3-4=9>>9 duck eggs a day.\nShe makes 9 * 2 = $<<9*2=18>>18 every day at the farmer's market."

Rendered Image: <width: 1762px><height: 32px>

<|begin_of_thought|> Janet sells 16 - 3 - 4 = <<16-3-4=9>>9 duck eggs a day. She makes 9 * 2 = $<<9*2=18>>18 every day at the farmer's market. <|end_of_thought|>

32 Latent Embeddings:



(a) Vision Embeddings Heatmap (b) Token Similarity Matrix (c) Statistical Properties of Embeddings

Prediction = "assistant\n### 18" ✅

GSM8k-Aug Case 2

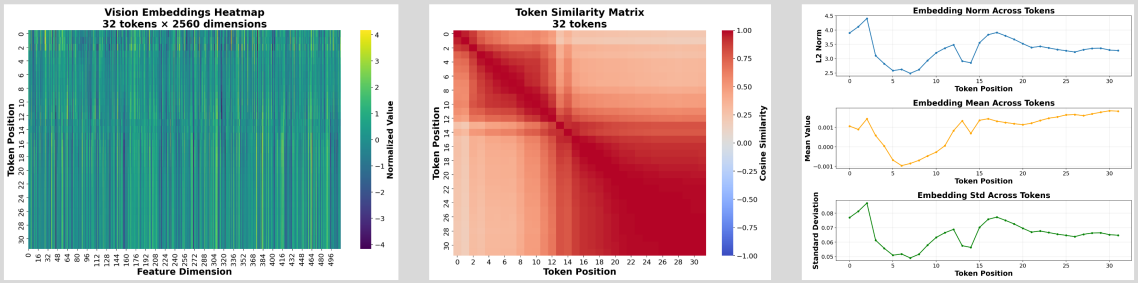
Question = "Claire makes a 3 egg omelet every morning for breakfast. How many dozens of eggs will she eat in 4 weeks?"

Textual CoT = "He sprints 3*3=<<3*3=9>>9 times\nSo he runs 9*60=<<9*60=540>>540 meters."

Rendered Image: <width: 1300px><height: 32px>

<|begin_of_thought|> He sprints 3*3=<<3*3=9>>9 times So he runs 9*60=<<9*60=540>>540 meters. <|end_of_thought|>

32 Latent Embeddings:



(a) Vision Embeddings Heatmap (b) Token Similarity Matrix (c) Statistical Properties of Embeddings

Prediction = "assistant\n### 540" ✅

GSM8k-Aug Case 3

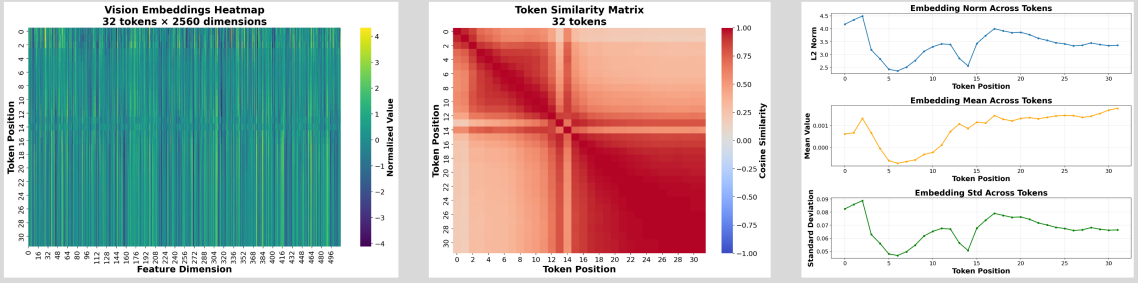
Question = "James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week?"

Textual CoT = "She eats 3 eggs every day and there are 7 days in a week so she eats 3*7 = <<3*7=21>>21 eggs a week\nAfter 4 weeks she will have eaten 4*21 = <<4*21=84>>84 eggs\nThere are 12 eggs in 1 dozen and she'll eat 84 eggs so that's 84/12 = <<84/12=7>>7 dozen eggs"

Rendered Image: <width: 3198px><height: 32px>

<|begin_of_thought|> She eats 3 eggs every day and there are 7 days in a week so she eats 3*7 ··· so that's 84/12 = <<84/12=7>>7 dozen eggs <|end_of_thought|>

32 Latent Embeddings:



(a) Vision Embeddings Heatmap (b) Token Similarity Matrix (c) Statistical Properties of Embeddings

Prediction = "assistant\n### 7" ✅

Figure 7: Visualization of reasoning on GSM8k-Aug dataset

Math Case 1

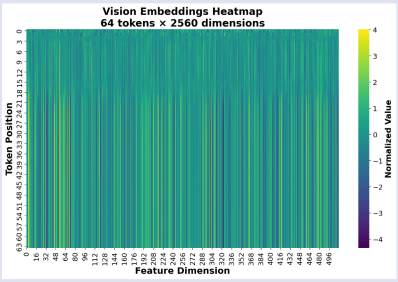
Question = "What is the least positive integer multiple of 30 that can be written with only the digits 0 and 2?"

Textual CoT = "Let M be the least positive multiple of 30 that can be written with only the digits 0 and 2. First, M is a multiple of 10, so its units digit must be 0. M is also a multiple of 3, which means the sum of its digits must be a multiple of 3. Therefore, we must take at least three 2's. Since M is minimal, we take exactly three 2's and do not have any additional 0's: $M=\\boxed{2220}$."

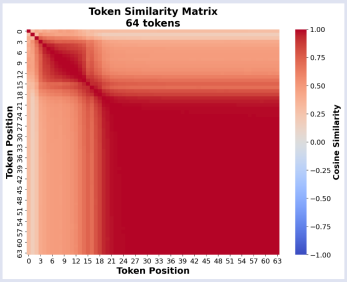
Rendered Image: <width: 4309px><height: 32px>

<|begin_of_thought|> Let M be the least positive multiple of 30 that can be written with only the digits ··· additional 0's: $M=\boxed{2220}$. <|end_of_thought|>

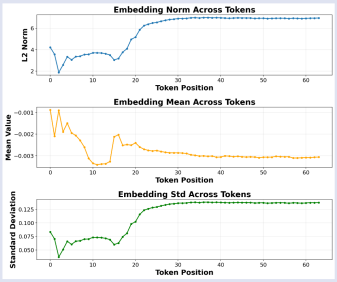
64 Latent Embeddings:



(a) Vision Embeddings Heatmap



(b) Token Similarity Matrix



(c) Statistical Properties of Embeddings

Prediction = "assistant\n### 2200" ✓

Math Case 2

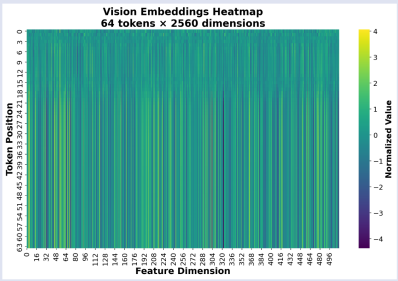
Question = "Two-thirds of the students at Baker Middle School take music. There are 834 students who take music. How many students are there at Baker Middle School?"

Textual CoT = "If there are S students at Baker Middle School, then $\\frac{2}{3}S$ students take music. Setting $\\frac{2}{3}S$ equal to 834 and multiplying both sides by $\\frac{3}{2}$, we find that there are $\\frac{3}{2}\\times 834=\\boxed{1251}$ students at the school."

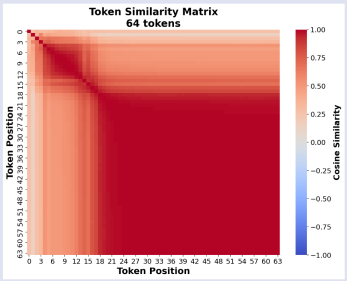
Rendered Image: <width: 3107px><height: 32px>

<|begin_of_thought|> If there are S students at Baker Middle School, then ··· $\frac{3}{2}\times 834=\boxed{1251}$ students at the school. <|end_of_thought|>

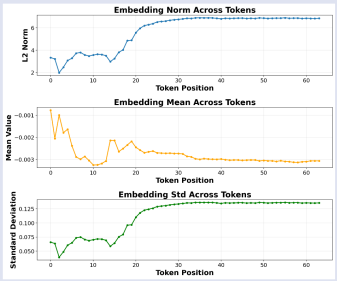
64 Latent Embeddings:



(a) Vision Embeddings Heatmap



(b) Token Similarity Matrix



(c) Statistical Properties of Embeddings

Prediction = "assistant\n### 1251" ✓

Math Case 3

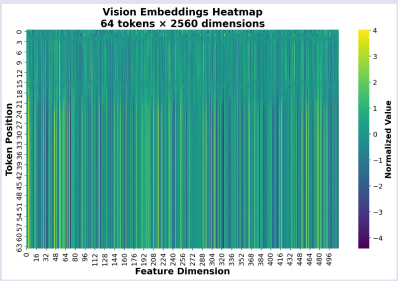
Question = "There is a total of 70 squares of three sizes whose vertices are points on this rectangular $3\\times n$ grid of points. What is the value of n?"

Textual CoT = "The number of $1\\times1$ squares in the diagram is $2(n-1)$, the number of $2\\times 2$ squares is $n-2$, and the number of $\\sqrt{2}\\times\\sqrt{2}$ squares is also $n-2$ (see diagram). Solving \\[\n2(n-1)+n-2+n-2=70\n\\] we find $n=\\boxed{19}$."

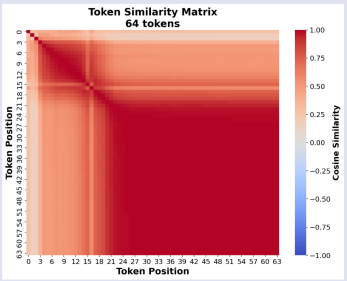
Rendered Image: <width: 2977px><height: 32px>

<|begin_of_thought|> The number of 1×1 squares in the diagram is $2(n-1)$, the number of 2×2 squares is $n-2$, ··· \boxed{19}$. <|end_of_thought|>

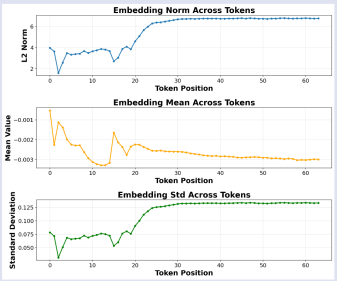
64 Latent Embeddings:



(a) Vision Embeddings Heatmap



(b) Token Similarity Matrix



(c) Statistical Properties of Embeddings

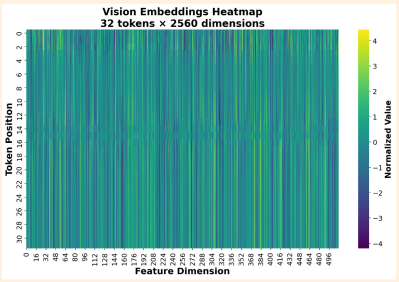
Prediction = "assistant\n### 10" ✗

Figure 8: Visualization of reasoning on the challenging MATH dataset.

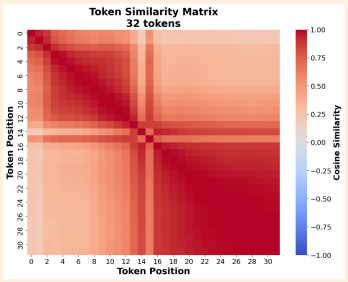
GSM-Hard Failure Case 1

Question = "Claire makes a 6022727 egg omelet every morning for breakfast. How many dozens of eggs will she eat in 4 weeks?"

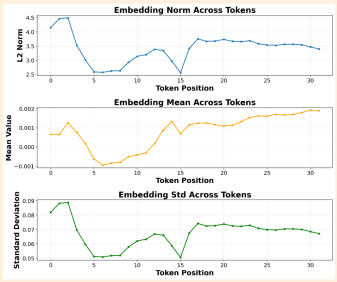
32 Latent Embeddings:



(a) Vision Embeddings Heatmap



(b) Token Similarity Matrix



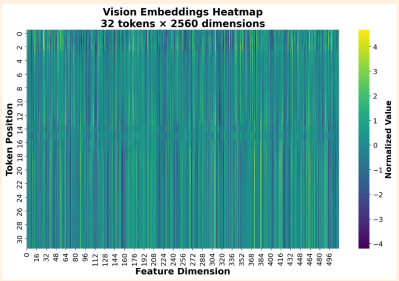
(c) Statistical Properties of Embeddings

Prediction = "assistant\n### 140568" ✗

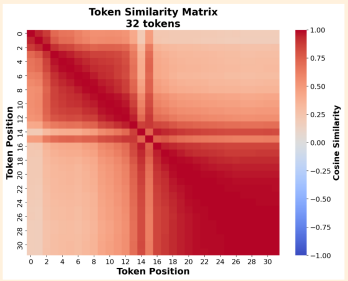
SVAMP Failure Case 1

Question = "Baker made 144 cakes. He sold 71 of them. Then he made 111 more cakes. How many more cakes did baker make than those he sold?"

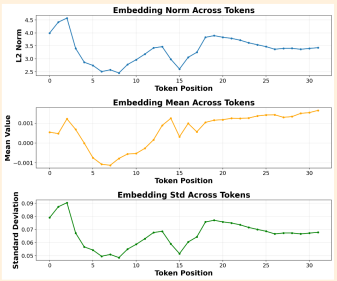
32 Latent Embeddings:



(a) Vision Embeddings Heatmap



(b) Token Similarity Matrix



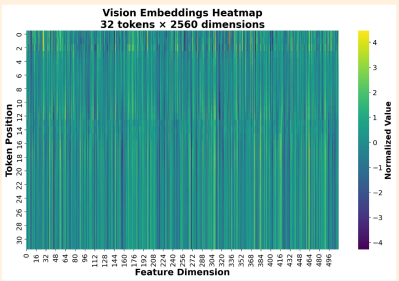
(c) Statistical Properties of Embeddings

Prediction = "assistant\n### 116" ✗

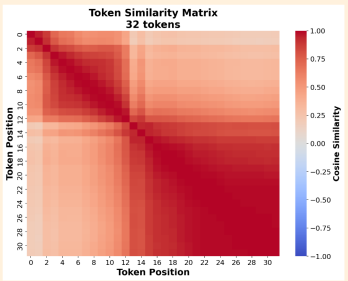
MultiArith Failure Case 1

Question = "Tom had 5 books. If he sold 4 of them and used the money he earned to buy 38 new books, how many books would Tom have?"

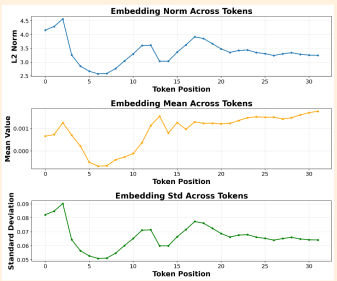
32 Latent Embeddings:



(a) Vision Embeddings Heatmap



(b) Token Similarity Matrix



(c) Statistical Properties of Embeddings

Prediction = "assistant\n### 41" ✗

Figure 9: Failure case analysis across out-of-distribution datasets.