CALR: CONTINUOUS ANCHORED LATENT REASONING VIA RENDER-OF-THOUGHT COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual latent reasoning compresses rendered derivations into compact intermediate states, reducing textual reasoning overhead. Existing approaches differ in how they represent these states: continuous methods avoid vocabulary constraints, whereas discrete methods improve accuracy through quantization into a finite codebook. Our analysis of representative continuous and discrete systems identifies two functional requirements: answers must rely on latent states, and those states must carry valid, problem-specific reasoning. Continuous latents influence answers despite collapsed reasoning content, whereas discrete latents retain recoverable intermediate reasoning that answer prediction largely bypasses. To address these challenges, we propose Continuous Anchored Latent Reasoning (CALR), which connects latent formation with answer use through functional anchoring. With reference latents from information-balanced compression, CALR couples latent-mediated answer supervision with derivation-level semantic anchoring: the former routes answer supervision through intermediate states, while the latter grounds their decoded content in problem-specific derivations. A parallel-to-autoregressive curriculum develops sequential reasoning by conditioning subsequent latent blocks on generated prefixes. Under matched budgets, CALR gains 26 points over a comparable continuous latent reasoning method. Further analyses show that its latents support answer prediction and carry problem-specific intermediate reasoning.

1 INTRODUCTION

Chain-of-thought (CoT) enables language models to develop intermediate reasoning before producing an answer (Wei et al., 2022). Verbalizing that reasoning, however, requires a long sequence of decoding steps. Latent reasoning moves the intermediate solution process into compact hidden states, reducing the need to express every step in words (Hao et al., 2025; Shen et al., 2025; Tan et al., 2025). Optical compression offers a concrete connection between these states and explicit derivations: rendering text as an image allows long passages to be encoded in substantially fewer visual tokens (Wei et al., 2025). Applied to reasoning traces, this motivates *visual latent reasoning*, in which a model generates optical latent states to carry its solution process before answering.

Previous approaches learn continuous visual latents from rendered reasoning traces, either aligning generated states with visual embeddings or regularizing latent distributions with visual-semantic features (Wang et al., 2026a;b). Motivated by collapse in continuous representations, subsequent work quantizes optical features into a discrete vocabulary and models the resulting latent codes autoregressively, achieving higher accuracy than earlier latent methods (Chang et al., 2026). Both paradigms advance compact representations of reasoning, but retain distinct representational limitations: existing continuous latents are susceptible to feature collapse, whereas the gains of discrete methods come with a restricted range of latent representations determined by a finite vocabulary.

To understand how these compact representations support problem solving, we analyze representative continuous and discrete systems through interventions and derivation decoding. Our analysis identifies two requirements for effective latent reasoning (Figure 1): *latent reliance* (C1), whereby answer prediction depends on the generated latents, and *reasoning capability* (C2), whereby those latents carry a valid derivation for the current problem. In the discrete system, intermediate information can be recovered from the latents, yet answer prediction largely bypasses them. In the continuous system, the latent interface is load-bearing, yet replacing states across problems barely changes accuracy, and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

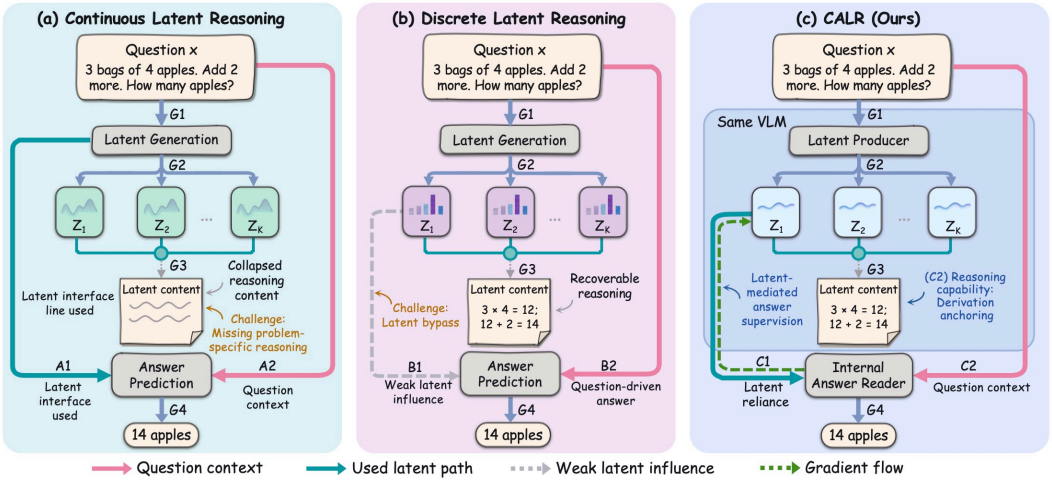

Figure 1: **The latent handoff challenge.** (a) Continuous latents can support answer prediction without carrying problem-specific reasoning. (b) Discrete latents can encode recoverable reasoning yet be bypassed by the answer model. (c) CALR couples latent reliance (C1) with derivation anchoring for reasoning capability (C2).

decoding recovers little problem-specific reasoning. These complementary failures reveal a *missing handoff* between latent formation and answer prediction: representation supervision shapes the latent states, while question-visible answer supervision can be satisfied without using the reasoning those states are intended to carry.

To address these challenges, we propose **Continuous Anchored Latent Reasoning (CALR)**, which couples answer prediction and derivation supervision through the same continuous latent trajectory. A question-blind external reader directs answer supervision through the latents, while a frozen DeepSeek-OCR2 decoder (Wei et al., 2026a) supervises their problem-specific reasoning content. Stage I establishes these supervisory paths by learning an information-balanced resampler of rendered derivations and training the external reader on the resulting reference latents. Stage II learns problem-conditioned latent generation under both frozen supervisors: parallel drafting warm-starts autoregressive refinement, in which later latent blocks build on generated prefixes. CALR supports a decoupled realization with an external answer model and a unified realization that generates latents and answers within the same VLM.

We evaluate CALR across five reasoning benchmarks and multiple model families. On GSM8K-Aug (Qwen3-VL-4B-Instruct), unified autoregressive CALR achieves 63.8% versus RoT's 37.8% accuracy with 32 continuous latents each, a 26.0-point gain. Interventions and decoding show that its latents support answers and carry problem-specific reasoning. Our contributions are threefold:

- We establish a unified functional framework for visual latent reasoning based on latent reliance and reasoning capability, revealing complementary failures in existing approaches.
- We propose CALR, a unified training framework that couples latent-mediated answer supervision with derivation anchoring, jointly shaping the use and reasoning content of continuous latent trajectories.
- We demonstrate that continuous latents can deliver substantial accuracy gains while carrying load-bearing, problem-specific reasoning, supported by cross-model evaluations, interventions, and derivation decoding.

2 Preliminary Analysis: The Missing Handoff in Latent Reasoning

Visual latent reasoning requires a latent trajectory that the answer model uses and that carries the reasoning needed to solve the problem. We formalize these requirements as *latent reliance* (C1) and *reasoning capability* (C2), then examine them in continuous and discrete visual latent systems.

2.1 Problem Formulation: Two Requirements for Visual Latent Reasoning

Let $(x, y) \sim \mathcal{D}$ denote a question–answer pair, and let $\mathcal{D}_X$ be its question marginal. A visual latent reasoner consists of a latent generator G_θ and an answer model p_ϕ,

$$G_\theta : \mathcal{X} \to \mathcal{Z}^K, \qquad p_\phi : (\mathcal{X} \cup \{\varnothing_x\}) \times (\mathcal{Z}^K \cup \{\varnothing_z\}) \to \Delta(\mathcal{Y}), \tag{1}$$

where $\mathcal{Z}$ may be a continuous visual feature space or a discrete optical code space, while $\varnothing_x$ and $\varnothing_z$ denote the absence of the question and latent inputs, respectively. The two maps specify functional roles and may share parameters. For a sequence of K latent states, inference is

$$z_{1:K} = G_\theta(x), \qquad p_{\theta,\phi}(y \mid x) = p_\phi(y \mid x, z_{1:K}), \qquad \hat{y} = \arg\max_{y' \in \mathcal{Y}} p_\phi(y' \mid x, z_{1:K}). \tag{2}$$

The intended computation therefore has two functional stages:

$$x \xrightarrow[\text{latent formation}]{G_\theta} z_{1:K}, \qquad (x, z_{1:K}) \xrightarrow[\text{answer generation}]{p_\phi} \hat{y}. \tag{3}$$

The first stage should develop the problem's intermediate reasoning within the latent trajectory; the second should draw on that trajectory to produce the answer. These roles define two requirements.

C1: Latent Reliance. In the common question-visible architecture of Equation 2, the answer model conditions on both x and $z_{1:K}$. The contextual route $x \to \hat{y}$ may therefore coexist with the latent-mediated route $x \to z_{1:K} \to \hat{y}$. Latent reliance does not require the answer model to discard x or to predict from $z_{1:K}$ alone. It requires the latent sequence to remain an active determinant of the answer distribution when the question is held fixed.

For a question x, C1 requires that disabling the latent input changes the answer distribution:

$$C_1(x) := \mathbf{1}[p_\phi(\cdot \mid x, G_\theta(x)) \neq p_\phi(\cdot \mid x, \varnothing_z)] = 1. \tag{C1}$$

Thus, with x unchanged, C1 makes the latent branch $z_{1:K} \to \hat{y}$ load-bearing within the full predictive computation. It asks whether the answer model uses the latent input; C2 concerns what it represents.

C2: Reasoning Capability. The latent-construction stage must itself support the reasoning that transforms the question into the answer. Let $\mathcal{T}(x, y) \subseteq \mathcal{T}$ be the set of valid derivations from x to y, where each derivation contains the non-trivial intermediate inferences required by the problem. Let $\widehat{c}(z)$ denote the textual derivation recovered from a latent trajectory by its prescribed decoder, held fixed during evaluation and given only z. Define

$$V_{\text{reason}}(x, z, y) = \mathbf{1}[\widehat{c}(z) \in \mathcal{T}(x, y)]\,. \tag{4}$$

For a question–answer pair (x, y), C2 requires

$$C_2(x, y) := V_{\text{reason}}(x, G_\theta(x), y) = 1. \tag{C2}$$

The trajectory $G_\theta(x)$ is produced for the current problem, and its decoded text is judged against the corresponding set $\mathcal{T}(x, y)$. This condition concerns the represented derivation rather than the final answer alone, and applies equally to continuous visual states and discrete optical codes.

Together, C1 and C2 align the two stages of Equation 3: the answer must rely on the latent trajectory, and the trajectory must carry a valid reasoning process. We use their conjunction as the criterion for visual latent reasoning:

$$\text{VLR}(x, y) := \underbrace{C_1(x)}_{\text{latent reliance}} \wedge \underbrace{C_2(x, y)}_{\text{reasoning capability}}. \tag{5}$$

2.2 Interventional Analysis of Continuous and Discrete Latent Reasoners

We analyze representative *continuous* and *discrete* optical-character latent reasoners under a shared interventional framework, detailed with supporting measurements in Appendix D.

Latent interventions. With question and answer model fixed, we intervene on *content* (cross-problem and constant replacement) or *interface* (norm-matched randomization and deletion). For native accuracy A_{base} and intervened accuracy A_ι, define $\Lambda_\iota = A_{\text{base}} - A_\iota$; Λ_{content} and $\Lambda_{\text{interface}}$ average each pair. Deletion tests C1, content replacement tests sensitivity to problem-specific latents, and derivation decoding assesses C2. Removing only the answer-side question, retaining native latents, gives $\Lambda_x = A_{\text{base}} - A_{-x}$. Figure 2 shows results, Appendix D.1 defines the measures.

Continuous latents: a load-bearing interface without problem-specific reasoning. Removing or randomizing RoT's latent interface reduces accuracy from 0.43 to 0.01, whereas cross-example and constant replacement retain 0.42 and 0.43. Answer prediction thus relies on the interface (C1; $\Lambda_{\text{interface}} = 0.42$) but is insensitive to its problem-specific content; latent attention remains substantial despite higher question attention (Appendix D). Yet decoding recovers no chain of thought from native states: they are interchangeable control signals, not problem-specific reasoning, failing C2.

Discrete latents: supervised codes bypassed by the answer model. DLR-SFT stays near its 0.53 baseline under all four latent interventions: cross-example and constant replacement preserve accuracy; randomization and deletion yield 0.52 and 0.51. Removing the answer-side question reduces accuracy to 0.01. Thus, the answer model bypasses supervised latents and relies on the question, failing C1, consistent with question-dominated attention and weak latent gradients.

These failures are complementary: the continuous system satisfies C1 but fails C2, the discrete system fails C1. Effective visual latent reasoning requires both.

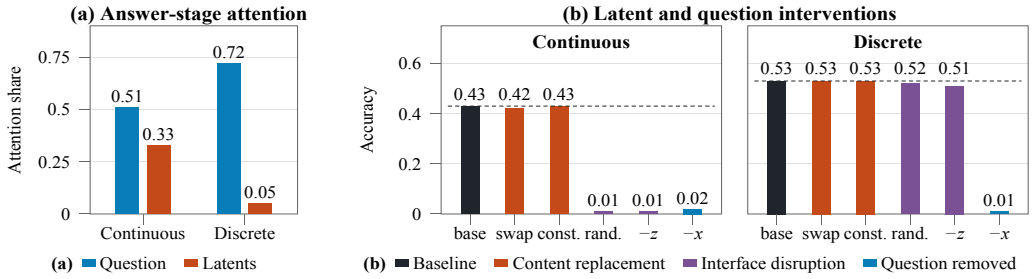

Figure 2: **Complementary handoff failures in continuous and discrete latents.** (a) Both favor the question, with higher latent attention in the continuous system. (b) Content replacement has little effect ($\Lambda_{\text{content}} \approx 0$); interface disruption yields accuracy drops of 0.42 and 0.015, respectively.

2.3 Question-Visible Training Decouples Latent Reasoning from Answering

The complementary failures point beyond representation format to the objective shared by both paradigms. Let $\mathcal{D}_{\text{tr}}$ be a training distribution over triples (x, c, y), where $c \in \mathcal{T}(x, y)$ is a reference derivation. A question-visible objective has the general form

$$\mathcal{L}_{\text{vis}}(\theta, \phi) = \mathbb{E}_{(x,c,y)\sim\mathcal{D}_{\text{tr}}} \left[-\log p_\phi(y \mid x, G_\theta(x)) + \lambda_{\text{repr}} \mathcal{L}_{\text{repr}}(G_\theta(x), c) \right], \tag{6}$$

where $\lambda_{\text{repr}} \geq 0$, and $\mathcal{L}_{\text{repr}}$ may align continuous states with rendered features or make discrete codes reconstruct a reference derivation. The representation term shapes what z represents; the answer term rewards the correct output while exposing both x and z. Neither requires problem-specific reasoning to be both formed in z and used for the answer.

This ambiguity is captured by the latent-content-invariant family

$$\mathcal{P}_{\text{inv}}(\theta) = \{ p_\phi : p_\phi(\cdot \mid x, z) = p_\phi(\cdot \mid x, z'),\ \forall x \in \text{supp}(\mathcal{D}_X),\ \forall z, z' \in \mathcal{Z}_\theta \}, \tag{7}$$

where $\mathcal{Z}_\theta = \{G_\theta(x) : x \in \text{supp}(\mathcal{D}_X)\}$ is the set of reachable trajectories. A question-only answer distribution belongs to this family. The representation term supervises G_θ without requiring the answer model to use its output, so representation supervision and latent-content-invariant answer prediction can coexist. Within this family, an answer model may still react to the presence of the latent interface, as in RoT, or bypass it entirely, as in DLR-SFT.

The objective therefore leaves latent formation and answer prediction uncoupled. Addressing C1 and C2 jointly calls for supervision that makes the same trajectory support answer prediction and encode a valid, problem-specific derivation.

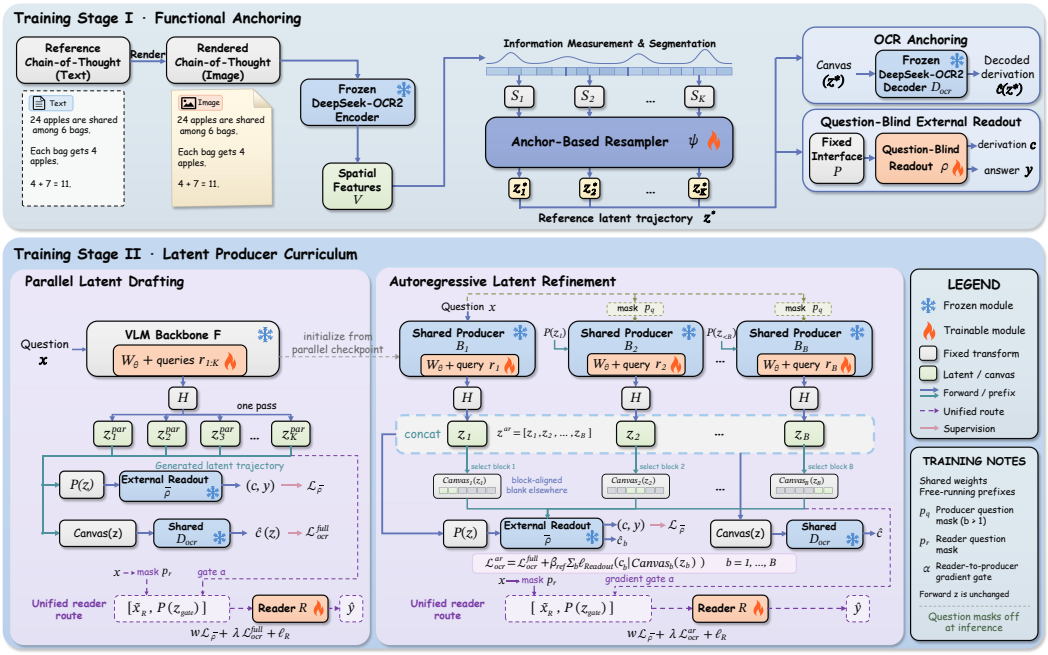

Figure 3: CALR training framework: functional anchoring followed by parallel drafting and autoregressive refinement. Purple dashed paths denote the unified internal-reader route.

3 CALR: Functional Anchoring of Latent Reasoning

3.1 Functional Formulation

For $(x, c, y) \sim \mathcal{D}_{\text{tr}}$ with $c \in \mathcal{T}(x, y)$, let $z = G_\theta(x)$. We formulate the supervision shared by both realizations through decoupled CALR. During producer learning, its question-blind reader $\bar{\rho}$, optical decoder D_{ocr}, and input mappings P and Canvas remain fixed. The answer model and prescribed decoder of Section 2.1 are instantiated as

$$p_\phi(y \mid x, z) := p_{\bar{\rho}}(y \mid P(z), \tau_{\text{ans}}), \qquad \widehat{c}(z) := \text{Decode}_{D_{\text{ocr}}}(\text{Canvas}(z)). \tag{8}$$

The reader's mode tokens τ_{ans} and τ_{cot} select answer and derivation recovery, respectively. Both readouts receive the same trajectory without direct question input.

Joint supervision of answer use and reasoning content. Writing $\ell_M(t \mid u) = -\log p_M(t \mid u)$, the full-trajectory producer objective is

$$\min_\theta \ \mathbb{E}_{(x,c,y)\sim\mathcal{D}_{\text{tr}}} \Big[\underbrace{\ell_{\bar{\rho}}(y \mid P(z), \tau_{\text{ans}})}_{\text{answer supervision}} + \underbrace{\eta \ell_{\bar{\rho}}(c \mid P(z), \tau_{\text{cot}}) + \lambda \ell_{D_{\text{ocr}}}(c \mid \text{Canvas}(z))}_{\text{derivation supervision}} \Big]. \tag{9}$$

Here $\eta, \lambda \geq 0$, $\mathcal{L}_{\bar{\rho}}(z)$ combines the reader's answer and η-weighted derivation losses, and $\mathcal{L}_{\text{ocr}}^{\text{full}}(z) := \ell_{D_{\text{ocr}}}(c \mid \text{Canvas}(z))$, giving $\mathcal{L}_{\bar{\rho}}(z) + \lambda \mathcal{L}_{\text{ocr}}^{\text{full}}(z)$ per example.

Coupling formation to use. Unlike Equation 6, the answer term receives problem-specific information only through $G_\theta(x)$. A readout invariant across reachable trajectories would be question-independent, unable to exploit the direct question-conditioned route admitted by $\mathcal{P}_{\text{inv}}(\theta)$. Latent-mediated answer supervision thus addresses the use requirement in C1. For C2, the OCR term supervises $c \in \mathcal{T}(x, y)$ through the prescribed evaluation decoder; the reader's derivation mode additionally supervises recovery through $P(z)$.

During producer learning, gradients pass through the fixed readouts and input mappings to update G_θ via z, adapting the trajectory under stable answer and derivation readout rules.

3.2 Constructing Functional Anchors

Stage I establishes the reference trajectories and fixed readouts required by the objective before learning question-conditioned generation in Stage II (Figure 3).

Information-balanced reference trajectories. To preserve derivation content within a fixed latent budget, we balance information across states. The frozen DeepSeek-OCR2 encoder E maps a rendered derivation c to spatial features $V \in \mathbb{R}^{N\times d}$. Corpus statistics (μ_j, σ_j), fixed before training, define an unnormalized diagonal-Gaussian feature-surprisal score a_i. We form K contiguous segments of approximately equal cumulative score and resample each with global page context:

$$V = E(\mathrm{render}(c)), \qquad a_i = \frac{1}{2}\sum_{j=1}^{d}\left(\frac{V_{ij}-\mu_j}{\sigma_j}\right)^2, \qquad \sum_{i\in S_g} a_i \approx \frac{1}{K}\sum_{i=1}^{N} a_i, \tag{10}$$
$$z_g^\star = \mathrm{Attn}_\psi\big(q_{\mathrm{pool}}, \mathrm{Ctx}_\psi(V)_{S_g}\big), \qquad g = 1,\dots,K.$$

Here ψ comprises the shared learned query q_{pool}, context, and resampling parameters, and $z^\star = (z_1^\star,\dots,z_K^\star)$ with $z_g^\star \in \mathbb{R}^{d_z}$. Information-dense regions receive narrower spatial segments, while sparse regions receive wider ones. Reference text is separately segmented at paragraph and punctuation boundaries; spans are assigned in reading order, with c_g denoting the target for latent g.

Anchoring derivation content. The frozen DeepSeek-OCR2 decoder D_{ocr} supervises both the complete reference trajectory and its individual segments. The full canvas places all K latents on a fixed visual grid; $\mathrm{Canvas}_g(z_g)$ retains only position g. Unused positions contain blank-page features. We learn the resampler by minimizing

$$\begin{aligned}\mathcal{L}_{\mathrm{anchor}} &= \ell_{D_{\mathrm{ocr}}}(c \mid \mathrm{Canvas}(z^\star)) \\ &\quad + \beta\sum_{g=1}^{K}\ell_{D_{\mathrm{ocr}}}\big(c_g \mid \mathrm{Canvas}_g(z_g^\star)\big) + \gamma\Omega(z^\star), \quad \Omega(z^\star) = \frac{1}{K}\sum_{g=1}^{K}\big(\|z_g^\star\|_2 - \nu_0\big)^2,\end{aligned} \tag{11}$$

where $\beta, \gamma \geq 0$ and ν_0 is the target feature norm. Global supervision preserves the complete derivation, while segment supervision makes its ordered parts individually decodable. These reference trajectories establish the representation on which the answer reader is trained. Appendix C details spatial segmentation, text targets, and canvas construction.

Anchoring answer recovery. The reader ρ uses a trainable low-rank readout adapter and mode-token embeddings on a frozen base VLM F. It receives reference latents through a fixed scaled-isometric interface. For $d_z \leq d_{\mathrm{emb}}$, define

$$P = sU, \qquad H = s^{-1}U^\top, \qquad U \in \mathbb{R}^{d_{\mathrm{emb}}\times d_z}, \qquad U^\top U = I_{d_z}, \qquad HP = I_{d_z}, \tag{12}$$

where $s > 0$. Both maps act position-wise: P embeds optical latents in the VLM input space, and H maps producer states back to optical coordinates. Keeping the resampler and reference latents fixed, we train ρ without the question:

$$\mathcal{L}_{\mathrm{contract}} = \ell_\rho(y \mid P(z^\star), \tau_{\mathrm{ans}}) + \eta\ell_\rho(c \mid P(z^\star), \tau_{\mathrm{cot}}). \tag{13}$$

The answer and derivation targets make both functions accessible from the anchored representation. We then freeze ρ as $\bar{\rho}$. Stage II learns to generate trajectories that support these established readouts, with P and H also fixed throughout.

3.3 Learning Anchored Latent Trajectories

Stage II replaces reference trajectories from c with trajectories generated from x. We train a parallel producer under the joint objective, then refine it autoregressively through prefix conditioning and block-aligned derivation recovery. Both phases support decoupled and unified CALR.

Parallel drafting. The producer uses the frozen base VLM F with a low-rank route W_θ and learned queries $r_{1:K}$. These parameters and the special-token embeddings form θ. Selecting query states with Π_r, we generate all K latents in one pass:

$$z^{\mathrm{par}} = G_\theta^{\mathrm{par}}(x) = H\big(\Pi_r F^{W_\theta}([x; r_{1:K}])\big). \tag{14}$$

Training minimizes $\mathcal{L}_{\bar{\rho}}(z^{\mathrm{par}}) + \lambda\mathcal{L}_{\mathrm{ocr}}^{\mathrm{full}}(z^{\mathrm{par}})$. This phase learns question-conditioned trajectories under the fixed readout functions before introducing inter-block feedback.

Autoregressive refinement. Warm-starting from the parallel-trained producer, we make later blocks condition on the generated prefix. Let $(\mathcal{B}_1, \ldots, \mathcal{B}_B)$ partition the K latent positions into ordered contiguous blocks, with $\sum_b |\mathcal{B}_b| = K$. Write $\mathbf{z}_b := z_{\mathcal{B}_b}$, $\mathbf{r}_b := r_{\mathcal{B}_b}$, and $c_{\mathcal{B}_b} := \mathrm{concat}_{g \in \mathcal{B}_b} c_g$ for latent blocks, query blocks, and their derivation targets. The recurrence is

$$\mathbf{z}_b = H\big(\Pi_{\mathbf{r}_b} F^{W_\theta}([\widetilde{x}_b; P(\mathbf{z}_{<b}); \mathbf{r}_b])\big), \quad b = 1, \ldots, B; \;\; z^{\mathrm{ar}} = \mathrm{concat}(\mathbf{z}_1, \ldots, \mathbf{z}_B). \tag{15}$$

Here $\mathbf{z}_{<b}$ concatenates preceding blocks and $\widetilde{x}_1 = x$. For $b \geq 2$, $\widetilde{x}_b$ is replaced by $\langle \mathrm{blind} \rangle$ with probability p_q; on these steps, problem-specific information must reach the next block through the prefix. Training and inference use free-running prefixes, with gradients connected across blocks during training.

Prefix conditioning governs inter-block information flow, while the frozen external reader supervises each block's reasoning content. We add block-aligned recovery to full-trajectory OCR supervision:

$$\mathcal{L}^{\mathrm{ar}}_{\mathrm{ocr}}(z^{\mathrm{ar}}) = \mathcal{L}^{\mathrm{full}}_{\mathrm{ocr}}(z^{\mathrm{ar}}) + \beta_{\mathrm{ref}} \sum_{b=1}^{B} \ell_{\mathrm{Read}_{\mathrm{out}}}(c_{\mathcal{B}_b} \mid \mathrm{Canvas}_{\mathcal{B}_b}(\mathbf{z}_b)). \tag{16}$$

Here $\mathrm{Read}_{\mathrm{out}}$ denotes the canvas-based derivation readout of $\bar{\rho}$, with its fixed input interface and τ_{cot} implicit. Block canvases retain absolute positions with blank-page padding. We minimize $\mathcal{L}_{\bar{\rho}}(z^{\mathrm{ar}}) + \lambda \mathcal{L}^{\mathrm{ar}}_{\mathrm{ocr}}(z^{\mathrm{ar}})$, retaining full-trajectory reader losses for answers and derivations.

Unified CALR. Unified CALR learns answer use inside the *same VLM* F while retaining the fixed external supervision. An internal reader adapter R, initialized from the Stage I readout adapter, defines the answer model $p_R := p_{F^R}$ and remains trainable alongside the separate producer adapter. It receives $P(z)$ and $\widetilde{x}_R$, which equals $\langle \mathrm{blind} \rangle$ with probability p_r and x otherwise. This reader mask trains answer recovery from latents even when question context is absent. We separate learning to read the trajectory from influencing its generation through a gradient gate:

$$\begin{aligned} z^m_{\mathrm{gate}} &= \alpha z^m + (1-\alpha)\,\mathrm{sg}[z^m], \qquad \alpha \in [0,1], \\ \mathcal{L}^m_{\mathrm{uni}} &= w\mathcal{L}_{\bar{\rho}}(z^m) + \lambda \mathcal{L}^m_{\mathrm{ocr}}(z^m) + \mathbb{E}_{\widetilde{x}_R} \ell_R\big(y \mid \widetilde{x}_R, P(z^m_{\mathrm{gate}}), \tau_{\mathrm{ans}}\big). \end{aligned} \tag{17}$$

Here $m \in \{\mathrm{par}, \mathrm{ar}\}$, $\mathcal{L}^{\mathrm{par}}_{\mathrm{ocr}} := \mathcal{L}^{\mathrm{full}}_{\mathrm{ocr}}$, $w \geq w_{\min} > 0$, and sg stops gradients. The gate preserves $z^m_{\mathrm{gate}} = z^m$ in the forward pass and scales only the internal reader's gradient into the producer; R continues learning at $\alpha = 0$. The fixed external losses remain active throughout. At inference, both masks are disabled: decoupled CALR uses the frozen external reader on $P(z^{\mathrm{ar}})$, while unified CALR uses the internal reader on $[x; P(z^{\mathrm{ar}})]$. Appendix B.1 details gradient admission and inference.

4 Experiments

4.1 Experimental Setup

Training and base model. We train the reader and producer on GSM8K-Aug. Resampler training additionally uses rendered reference derivations from MATH and MathX. The frozen DeepSeek-OCR2 encoder and decoder supply optical features and supervision, separately from the base VLM. Our main model is Qwen3-VL-4B-Instruct, with evaluations also covering Qwen3-VL at 2B and 8B and LLaVA-V1.6-Mistral-7B. CALR uses 32 continuous latents, arranged as four blocks of eight during autoregressive refinement. Appendix B details data allocation and training configurations.

Evaluation and metrics. We evaluate on GSM8K-Aug, GSM-Hard, SVAMP, MultiArith, and MATH-500, with disjoint training and evaluation data. Models receive only the question at inference. We report greedy-decoding final-answer accuracy (Pass@1 or Acc.) and intermediate reasoning length (#L), using unweighted benchmark averages. Appendix D details diagnostics and ablations.

4.2 Main Results

Supervised latent reasoning. Table 1 compares CALR with supervised latent-reasoning baselines on Qwen3-VL-4B-Instruct. Unified autoregressive CALR achieves the highest accuracy among the compared latent methods on all four benchmarks, averaging 64.4% versus 57.3% for DLR (SFT).

Table 1: Results on four grade-school reasoning benchmarks with Qwen3-VL-4B-Instruct. Par. and AR denote parallel drafting and autoregressive refinement, respectively.

	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average	
Method	Acc.	#L	Acc.	#L	Acc.	#L	Acc.	#L	Acc.	#L
Qwen3-VL-4B-Instruct (LMM)										
Direct-SFT	$26.2_{\pm.3}$	$0.0_{\pm.0}$	$9.5_{\pm.1}$	$0.0_{\pm.0}$	$70.0_{\pm.3}$	$0.0_{\pm.0}$	$85.6_{\pm.4}$	$0.0_{\pm.0}$	47.8	0.0
CoT-SFT	$81.2_{\pm.4}$	$127.3_{\pm1.0}$	$53.4_{\pm.3}$	$191.1_{\pm1.6}$	$84.3_{\pm.3}$	$55.9_{\pm.6}$	$98.3_{\pm.4}$	$59.1_{\pm.6}$	79.3	108.4
Coconut (CoLM 2025)	$16.1_{\pm.4}$	$6.0_{\pm.0}$	$5.2_{\pm.3}$	$6.0_{\pm.0}$	$42.0_{\pm.3}$	$6.0_{\pm.0}$	$58.9_{\pm.2}$	$6.0_{\pm.0}$	30.6	6.0
CODI (EMNLP 2025)	$7.1_{\pm.3}$	$6.0_{\pm.0}$	$2.0_{\pm.4}$	$6.0_{\pm.0}$	$10.7_{\pm.1}$	$6.0_{\pm.0}$	$17.8_{\pm.2}$	$6.0_{\pm.0}$	9.4	6.0
CoLaR (NeurIPS 2025)	$39.2_{\pm.2}$	$38.8_{\pm.5}$	$8.2_{\pm.4}$	$47.8_{\pm.7}$	$52.0_{\pm.4}$	$18.4_{\pm.6}$	$81.0_{\pm.3}$	$22.1_{\pm.3}$	45.1	31.8
RoT (ACL 2026)	$37.8_{\pm.3}$	$32.0_{\pm.0}$	$14.1_{\pm.2}$	$32.0_{\pm.0}$	$72.7_{\pm.4}$	$32.0_{\pm.0}$	$97.2_{\pm.4}$	$32.0_{\pm.0}$	55.5	32.0
DLR (SFT)	$48.0_{\pm.4}$	$6.3_{\pm.4}$	$14.7_{\pm.3}$	$6.8_{\pm.5}$	$68.7_{\pm.2}$	$3.6_{\pm.1}$	$97.8_{\pm.3}$	$4.2_{\pm.2}$	57.3	5.2
Ours, decoupled (Par.)	$51.2_{\pm.2}$	$32.0_{\pm.0}$	$13.8_{\pm.3}$	$32.0_{\pm.0}$	$71.3_{\pm.2}$	$32.0_{\pm.0}$	$97.6_{\pm.1}$	$32.0_{\pm.0}$	58.5	32.0
Ours, unified (Par.)	$52.4_{\pm.3}$	$32.0_{\pm.0}$	$14.3_{\pm.2}$	$32.0_{\pm.0}$	$72.4_{\pm.1}$	$32.0_{\pm.0}$	$98.1_{\pm.1}$	$32.0_{\pm.0}$	59.3	32.0
Ours, decoupled (AR)	$62.1_{\pm.2}$	$32.0_{\pm.0}$	$16.1_{\pm.2}$	$32.0_{\pm.0}$	$76.6_{\pm.3}$	$32.0_{\pm.0}$	$98.8_{\pm.1}$	$32.0_{\pm.0}$	63.4	32.0
Ours, unified (AR)	$\mathbf{63.8}_{\pm.2}$	$32.0_{\pm.0}$	$\mathbf{16.8}_{\pm.3}$	$32.0_{\pm.0}$	$\mathbf{77.4}_{\pm.1}$	$32.0_{\pm.0}$	$\mathbf{99.7}_{\pm.1}$	$32.0_{\pm.0}$	**64.4**	32.0

Autoregressive refinement benefits both realizations: average accuracy rises from 58.5% to 63.4% for decoupled CALR and from 59.3% to 64.4% for unified CALR, with all configurations retaining 32 latents. Unified autoregressive CALR further improves on its decoupled counterpart on every benchmark. These results show that continuous latent reasoning can outperform discrete latent coding under supervised training, with refinement improving both external and internal readout realizations. The complete comparison appears in Appendix E.1.

Table 2: Comparison with RoT across three vision-language models on four mathematical reasoning benchmarks. Pass@1 is reported in percent, and #L denotes reasoning length.

	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average		
Method	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	$15.6_{\pm.3}$	$0.0_{\pm.0}$	$4.7_{\pm.2}$	$0.0_{\pm.0}$	$52.3_{\pm.3}$	$0.0_{\pm.0}$	$41.7_{\pm.3}$	$0.0_{\pm.0}$	28.6	0.0	–
SFT-CoT	$59.7_{\pm.4}$	$131.4_{\pm1.6}$	$33.1_{\pm.3}$	$207.2_{\pm1.7}$	$67.3_{\pm.3}$	$63.4_{\pm.8}$	$95.0_{\pm.4}$	$68.0_{\pm.7}$	63.8	117.5	0.54
RoT	$23.3_{\pm.3}$	$32.0_{\pm.0}$	$8.6_{\pm.2}$	$32.0_{\pm.0}$	$53.7_{\pm.4}$	$32.0_{\pm.0}$	$62.2_{\pm.4}$	$32.0_{\pm.0}$	37.0	32.0	1.16
Ours (CALR)	$42.8_{\pm.3}$	$32.0_{\pm.0}$	$10.6_{\pm.2}$	$32.0_{\pm.0}$	$59.5_{\pm.3}$	$32.0_{\pm.0}$	$85.5_{\pm.2}$	$32.0_{\pm.0}$	49.6	32.0	**1.55**
Qwen3-VL-4B-Instruct											
SFT-w/o	$26.2_{\pm.3}$	$0.0_{\pm.0}$	$9.5_{\pm.1}$	$0.0_{\pm.0}$	$70.0_{\pm.3}$	$0.0_{\pm.0}$	$85.6_{\pm.4}$	$0.0_{\pm.0}$	47.8	0.0	–
SFT-CoT	$81.2_{\pm.4}$	$127.3_{\pm1.0}$	$53.4_{\pm.3}$	$191.1_{\pm1.6}$	$84.3_{\pm.3}$	$55.9_{\pm.6}$	$98.3_{\pm.4}$	$59.1_{\pm.6}$	79.3	108.4	0.73
RoT	$37.8_{\pm.3}$	$32.0_{\pm.0}$	$14.1_{\pm.2}$	$32.0_{\pm.0}$	$72.7_{\pm.4}$	$32.0_{\pm.0}$	$97.2_{\pm.4}$	$32.0_{\pm.0}$	55.5	32.0	1.73
Ours (CALR)	$63.8_{\pm.2}$	$32.0_{\pm.0}$	$16.8_{\pm.3}$	$32.0_{\pm.0}$	$77.4_{\pm.1}$	$32.0_{\pm.0}$	$99.7_{\pm.1}$	$32.0_{\pm.0}$	64.4	32.0	**2.01**
LLaVA-V1.6-Mistral-7B											
SFT-w/o	$10.8_{\pm.2}$	$0.0_{\pm.0}$	$2.3_{\pm.1}$	$0.0_{\pm.0}$	$40.7_{\pm.3}$	$0.0_{\pm.0}$	$52.8_{\pm.5}$	$0.0_{\pm.0}$	26.7	0.0	–
SFT-CoT	$38.6_{\pm.3}$	$151.3_{\pm1.6}$	$12.2_{\pm.1}$	$209.7_{\pm1.8}$	$56.3_{\pm.5}$	$79.9_{\pm.9}$	$86.1_{\pm.5}$	$82.3_{\pm.8}$	48.3	130.8	0.37
RoT	$16.3_{\pm.2}$	$32.0_{\pm.0}$	$3.6_{\pm.1}$	$32.0_{\pm.0}$	$49.0_{\pm.4}$	$32.0_{\pm.0}$	$68.3_{\pm.5}$	$32.0_{\pm.0}$	34.3	32.0	1.07
Ours (CALR)	$28.4_{\pm.3}$	$32.0_{\pm.0}$	$5.2_{\pm.2}$	$32.0_{\pm.0}$	$52.7_{\pm.2}$	$32.0_{\pm.0}$	$81.3_{\pm.3}$	$32.0_{\pm.0}$	41.9	32.0	**1.31**

Consistency across vision-language models. Table 2 compares CALR with RoT at the same length of 32 continuous latents. CALR improves Pass@1 in all twelve model–benchmark combinations: average accuracy rises from 37.0% to 49.6% on Qwen3-VL-2B, from 55.5% to 64.4% on Qwen3-VL-4B, and from 34.3% to 41.9% on LLaVA-Mistral-7B. The GSM8K-Aug gain on Qwen3-VL-4B reaches 26.0 percentage points. CALR also attains the highest reported Pass@1/#L in each model group. These gains span model families and scales without increasing latent sequence length.

Mathematical reasoning across scales. On MATH-500 (Table 3), CALR outperforms RoT at all three Qwen3-VL scales while using 32 rather than 64 latents. Its advantage over DLR (SFT) increases from 5.2 points at 2B to 7.9 at 4B and 15.9 at 8B, showing that the continuous formulation benefits from larger models in this setting. Without reinforcement learning, CALR also attains the

Table 3: Qwen3-VL on MATH-500.

	2B		4B		8B	
Method	Acc.	#L	Acc.	#L	Acc.	#L
Direct-SFT	$20.8_{\pm.2}$	$0.0_{\pm.0}$	$29.4_{\pm.3}$	$0.0_{\pm.0}$	$33.5_{\pm.5}$	$0.0_{\pm.0}$
CoT-SFT	$29.2_{\pm.3}$	$324.5_{\pm2.6}$	$55.8_{\pm.4}$	$291.5_{\pm1.9}$	$69.4_{\pm.3}$	$310.6_{\pm1.7}$
RoT	$24.0_{\pm.2}$	$64.0_{\pm.0}$	$33.2_{\pm.4}$	$64.0_{\pm.0}$	$37.8_{\pm.3}$	$64.0_{\pm.0}$
DLR (SFT)	$29.4_{\pm.3}$	$26.7_{\pm.8}$	$35.4_{\pm.3}$	$27.9_{\pm.7}$	$39.8_{\pm.4}$	$27.2_{\pm.5}$
DLR (RL)	$34.0_{\pm.3}$	$28.9_{\pm.6}$	$\mathbf{44.6}_{\pm.4}$	$28.4_{\pm.5}$	$54.0_{\pm.3}$	$29.5_{\pm.6}$
CALR	$\mathbf{34.6}_{\pm.2}$	$32.0_{\pm.0}$	$43.3_{\pm.3}$	$32.0_{\pm.0}$	$\mathbf{55.7}_{\pm.2}$	$32.0_{\pm.0}$

Table 4: Unified training ablations.

Training	Training components			GSM8K-Aug	
	Ext. (Par.)	Mask (p_r)	Ext. (AR)	Pass@1	#L
Par. only	×	×	—	39.7	32.0
	✓	×	—	46.8	32.0
	✓	✓	—	52.2	32.0
Par. → AR	✓	×	×	53.7	32.0
	✓	✓	×	56.3	32.0
	✓	✓	✓	**63.8**	32.0

highest latent-method accuracy at 2B and 8B; at 4B, it reaches 43.3%, compared with 44.6% for DLR (RL). Together, these results extend CALR's accuracy gains beyond the grade-school benchmarks.

4.3 Component Ablations

Training curriculum and supervision. External supervision contributes at both stages of the curriculum (Table 4). In parallel training, it improves Pass@1 from 39.7% to 46.8%, and reader question masking further raises it to 52.2%. During autoregressive refinement with masking, removing external supervision yields 56.3%, compared with 63.8% when retained. This 7.5-point gap demonstrates the value of continued external supervision after the parallel warm start. With both components, refinement improves over parallel training by 11.6 points while retaining 32 latents. These results support autoregressive refinement with latent-mediated supervision and reader question masking.

Information-balanced resampling. Table 5 links reconstruction fidelity to the resampling strategy. Fixed anchors improve on global cross-attention, but ignore variation in information density. Adapting segment widths to information content raises character similarity from 95.3% to 99.1% and reduces decoder cross-entropy from 0.158 to 0.107. Balancing information per latent thus preserves reference derivations more faithfully than assigning equal feature counts. Settings and metrics are detailed in Appendix D.6. Complementary diagnostics (Appendix E.2) show that cross-problem latent swaps reduce CALR's accuracy from 64.3% to 1.4% with the question retained, while decoded trajectories recover 74% of intermediate values for the current problem versus 7% for others. Feedback interventions further support inter-block dependence.

Table 5: Resampler ablations.

Resampling	GSM8K-Aug	
	Sim. (%)↑	CE↓
Global cross-attn.	92.8	0.183
Fixed anchors	95.3	0.158
Dynamic anchors	**99.1**	**0.107**

5 Conclusion

In this paper, we propose CALR to bridge the gap between latent formation and answer prediction in render-of-thought compression. Guided by latent reliance and reasoning capability, CALR couples latent-mediated answer supervision with derivation anchoring. A parallel-to-autoregressive curriculum develops continuous trajectories whose successive blocks build on generated prefixes. Experiments across five mathematical reasoning benchmarks and multiple model families demonstrate improved accuracy with compact latent states. Interventions and derivation decoding show that these states both support answer prediction and carry problem-specific intermediate reasoning. Together, these findings establish functional anchoring as a means of jointly shaping what continuous latents represent and how they support reasoning.

6 Related Work

Latent reasoning. Continuous methods internalize reasoning through hidden-state recurrence in Coconut (Hao et al., 2025), self-distillation in CODI (Shen et al., 2025), and dynamic trace compression in CoLaR (Tan et al., 2025). SoftCoT (Xu et al., 2025a) supplies assistant-generated soft thoughts. SIM-CoT (Wei et al., 2026b) adds step-aligned supervision through a training-only auxiliary decoder. CALR couples derivation anchoring with question-blind answer supervision, jointly shaping what latents represent and how they support prediction. See Appendix A for more.

AI USE STATEMENT

Large Language Models (LLMs) were used solely to assist with the writing and language polishing of this manuscript. Specifically, LLMs were used for sentence rephrasing, grammar checking, improving readability, and enhancing the overall clarity and flow of the text. LLMs were not used for research ideation, methodology development, experimental design, data analysis, or interpretation of experimental results. All scientific ideas, methods, experiments, analyses, and conclusions were developed and conducted by the authors. The authors take full responsibility for all content in this manuscript, including any text generated or edited with the assistance of LLMs.

ETHICS STATEMENT

Our research is conducted in accordance with established ethical practices, with particular attention to the responsible use of data. All datasets used in this study are publicly available and have been widely adopted in the computer vision and natural language processing communities. We do not collect any private or personally identifiable information, and our experiments do not involve human subjects or sensitive personal data.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of the research presented in this paper. Comprehensive implementation details for our models and experiments are provided in Appendix B, including the training procedures, experimental settings, and hyperparameters. Furthermore, upon acceptance, we will publicly release the source code, datasets, and trained model checkpoints to facilitate reproduction and further research.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Shuochen Chang, Qingyang Liu, Shaobo Wang, Bingjie Gao, Qianli Ma, Haonan Zhao, Yibo Miao, Yulin Sun, Zelin Peng, Jiangtong Li, and Li Niu. Why struggle with continuous latents? interpretable discrete latent reasoning via rendered compression. *arXiv preprint arXiv:2606.29712*, 2026. URL https://arxiv.org/abs/2606.29712.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025. URL https://arxiv.org/abs/2505.05410.

Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, Yushi Bai, Jie Tang, Hongning Wang, and Minlie Huang. Glyph: Scaling context windows via visual-text compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37145–37158. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1722. URL https://aclanthology.org/2026.acl-long.1722/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023. URL https://arxiv.org/abs/2311.01460.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit CoT to implicit CoT: Learning to internalize CoT step by step. *arXiv preprint arXiv:2405.14838*, 2024. URL https://arxiv.org/abs/2405.14838.

Connor Dilgren and Sarah Wiegreffe. Are latent reasoning models easily interpretable? In *Conference on Language Modeling*, 2026. URL https://arxiv.org/abs/2604.04902.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021. doi: 10.1162/tacl_a_00359. URL https://aclanthology.org/2021.tacl-1.10/.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10764–10799. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

Halil Alperen Gozeten, M. Emrullah Ildiz, Xuechen Zhang, Hrayr Harutyunyan, Ankit Singh Rawat, and Samet Oymak. Continuous chain of thought enables parallel exploration and reasoning. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2505.23648.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Itxz7S4Ip3.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://arxiv.org/abs/2103.03874.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. 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)*, pp. 3543–3556. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357/.

Houcheng Jiang, Jiajun Fu, Junfeng Fang, Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. UniVLR: Unifying text and vision in visual latent reasoning for multimodal LLMs. *arXiv preprint arXiv:2605.11856*, 2026. URL https://arxiv.org/abs/2605.11856.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023. URL https://arxiv.org/abs/2307.13702.

Yiqi Li, Xu Chen, Chen Ju, Jiangchao Yao, Zhaoyang Li, Jinsong Lan, Xiaoyong Zhu, Bo Zheng, and Yu Wang. Structural process supervision for latent chain-of-thought reasoning. *arXiv preprint arXiv:2609.09928*, 2026. URL https://arxiv.org/abs/2609.09928.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods, 2022. URL https://github.com/huggingface/peft.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Parvesh Rawal. MathX: Large-scale mathematical reasoning dataset. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/Modotte/MathX-5M. Released by Modotte.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1743–1752. Association for Computational Linguistics, 2015. doi: 10.18653/v1/D15-1202. URL https://aclanthology.org/D15-1202/.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. Language modelling with pixels. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2207.06991.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Wenhui Tan, Jiaze Li, Jianzhong Ju, Zhenbo Luo, Ruihua Song, and Jian Luan. Think silently, think fast: Dynamic latent compression of LLM reasoning chains. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0164.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, pp. 74952–74965, 2023. URL https://arxiv.org/abs/2305.04388.

Fanmeng Wang, Haotian Liu, Guojiang Zhao, Hongteng Xu, and Zhifeng Gao. ReGuLaR: Variational latent reasoning guided by rendered chain-of-thought. *arXiv preprint arXiv:2601.23184*, 2026a. URL https://arxiv.org/abs/2601.23184.

Yifan Wang, Shiyu Li, Peiming Li, Xiaochen Yang, Zheng Wei, and Yang Tang. Render-of-Thought: Rendering textual chain-of-thought as images for visual latent reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 45236–45253. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-long.2097. URL https://aclanthology.org/2026.acl-long.2097/.

Haoran Wei, Yaofeng Sun, and Yukun Li. DeepSeek-OCR: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025. URL https://arxiv.org/abs/2510.18234.

Haoran Wei, Yaofeng Sun, and Yukun Li. DeepSeek-OCR 2: Visual causal flow. *arXiv preprint arXiv:2601.20552*, 2026a. URL https://arxiv.org/abs/2601.20552.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1800. URL https://arxiv.org/abs/2201.11903.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. SIM-CoT: Supervised implicit chain-of-thought. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2509.20317.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Haoyi Wu, Zhihao Teng, and Kewei Tu. Parallel continuous chain-of-thought with jacobi iteration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 914–926. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.47. URL https://aclanthology.org/2025.emnlp-main.47/.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23336–23351. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.acl-long.1137. URL https://aclanthology.org/2025.acl-long.1137/.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. SoftCoT++: Test-time scaling with soft chain-of-thought reasoning. *arXiv preprint arXiv:2505.11484*, 2025b. URL https://arxiv.org/abs/2505.11484.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

A Extended Related Work

Latent reasoning and process supervision. Early approaches internalize explicit derivations through hidden-state distillation and progressive removal of verbalized steps (Deng et al., 2023; 2024). Subsequent work develops complementary ways to construct and learn latent trajectories. Coconut (Hao et al., 2025) feeds hidden states back as continuous thoughts, CODI (Shen et al., 2025) transfers explicit reasoning through self-distillation, and CoLaR (Tan et al., 2025) dynamically compresses reasoning traces. SoftCoT (Xu et al., 2025a) supplies assistant-generated soft thoughts, while SoftCoT++ (Xu et al., 2025b) encourages diverse latent paths for test-time scaling. Parallelism takes different forms: PCCoT (Wu et al., 2025) updates latent positions through Jacobi iterations, whereas CoT2 (Gozeten et al., 2026) represents multiple reasoning paths through mixtures of vocabulary embeddings. Beyond learning compact states, process supervision specifies the reasoning they should preserve. SIM-CoT (Wei et al., 2026b) applies step-aligned supervision through a training-only auxiliary decoder; concurrent prototype-mediated process supervision (PMPS) (Li et al., 2026) aligns latent and explicit trajectories through shared prototypes and many-to-many assignments. CALR connects this representational objective to answer prediction: derivation anchoring supervises latent reasoning content, while a question-blind external reader directs answer supervision through the generated trajectory.

Optical compression and rendered reasoning. Rendering provides a visual interface for language, exemplified by PIXEL (Rust et al., 2023), which learns from image patches of text. DeepSeek-OCR (Wei et al., 2025), DeepSeek-OCR2 (Wei et al., 2026a), and Glyph (Cheng et al., 2026) develop optical representations for compactly encoding textual contexts. Visual latent reasoning extends this interface from context encoding to supervision of problem-conditioned reasoning. RoT (Wang et al., 2026b) aligns generated latents with rendered-CoT features, while DLR (Chang et al., 2026) constructs a discrete optical vocabulary for autoregressive modeling. UniVLR (Jiang et al., 2026) combines textual derivations and auxiliary visual evidence within a shared rendered workspace. These approaches connect visual encoding with compact reasoning representations. CALR establishes an information-balanced continuous interface by anchoring reference trajectories and their segments with a frozen optical decoder, then couples derivation recovery with latent-mediated answer supervision through a frozen external reader.

Faithfulness of latent reasoning. Faithfulness concerns the relationship between represented reasoning and the computations governing a prediction. Studies of explicit CoT examine this relationship through input-bias probes (Turpin et al., 2023), interventions on reasoning traces (Lanham et al., 2023), and tests of whether reasoning models disclose answer-influencing hints (Chen et al., 2025). Representation analysis makes a related distinction between information availability and use. Attention interventions (Jain & Wallace, 2019) assess prediction sensitivity to attention distributions, while Amnesic Probing (Elazar et al., 2021) removes decodable properties to test their contribution to task behavior. Concurrent work on latent reasoning examines token necessity and recovers textual traces through vocabulary projections (Dilgren & Wiegreffe, 2026). CALR studies latent reliance and reasoning capability jointly across continuous and discrete visual reasoners. Answer-input interventions test the contribution of generated latents, optical decoding assesses their problem-specific reasoning content, and feedback-prefix interventions examine dependencies between successive latent blocks. This combination connects what latent trajectories encode with their roles in answer prediction and subsequent reasoning.

B CALR Realizations and Experimental Setup

B.1 Decoupled and Unified CALR: Training and Inference

This section brings together CALR's two realizations, their training objectives, the prefix interventions used to assess autoregressive dependence, and the resulting gradient-admission and inference procedures.

Model roles and initialization. For a question–derivation–answer triple (x, c, y), the producer G_θ generates K optical latents $z = (z_1, \ldots, z_K)$. The fixed interface P embeds these latents into the base VLM's input space, while H maps selected producer states back to optical coordinates. Stage I first learns the resampler ψ with the optical encoder E and decoder D_{ocr} frozen. Holding the resulting reference latents $z^\star$ fixed, we train the question-blind reader ρ through its low-rank adapter and mode-token embeddings. Its answer and derivation modes receive the same $P(z^\star)$, with respective mode tokens τ_{ans} and τ_{cot}. The trained reader is then frozen as $\bar{\rho}$.

Decoupled CALR pairs G_θ with this external readout: answer prediction conditions on $P(z)$ alone. Unified CALR performs latent generation and answer generation within the *same VLM* F, using separate low-rank producer and internal-reader adapters while keeping the pretrained parameters frozen. The internal reader R is initialized from the Stage I readout adapter and remains trainable during Stage II; $p_R := p_{F^R}$ denotes its conditional distribution. It receives both the question and $P(z)$, while $\bar{\rho}$ remains a frozen, question-blind supervisor. Both realizations learn a parallel producer and warm-start autoregressive refinement from it.

Training objectives and gradient flow. Let $m \in \{\mathrm{par}, \mathrm{ar}\}$ index the producer phase. The external loss $\mathcal{L}_{\bar{\rho}}(z^m)$ supervises both answers and derivations from $P(z^m)$. The loss $\mathcal{L}_{\bar{\rho}}(z^m)$ combines full-trajectory answer and derivation supervision from the frozen external reader. In the parallel phase, $\mathcal{L}^{\mathrm{par}}_{\mathrm{ocr}} := \mathcal{L}^{\mathrm{full}}_{\mathrm{ocr}}$ uses full-trajectory OCR supervision. Autoregressive refinement additionally supervises aligned derivation spans through the external reader's block-canvas readout, as defined in Equation 16.

$$
\begin{aligned}
z^m_{\mathrm{gate}} &= \alpha z^m + (1-\alpha)\,\mathrm{sg}[z^m], \qquad \alpha \in [0,1], \\
\mathcal{L}^m_{\mathrm{uni}} &= w\mathcal{L}_{\bar{\rho}}(z^m) + \lambda \mathcal{L}^m_{\mathrm{ocr}}(z^m) + \mathbb{E}_{\widetilde{x}_R} \ell_R\big(y \mid \widetilde{x}_R, P(z^m_{\mathrm{gate}}), \tau_{\mathrm{ans}}\big)\,.
\end{aligned}
$$

Here $\ell_R(t \mid u) = -\log p_R(t \mid u)$ is the negative log-likelihood of target t given conditioning input u, and sg is stop-gradient. The internal reader's question input $\widetilde{x}_R$ equals $\langle \mathrm{blind} \rangle$ with probability p_r and x otherwise. This readout mask is distinct from the producer mask p_q, which replaces the question for autoregressive blocks $b \geq 2$; the first block always receives x.

The gate satisfies $z^m_{\mathrm{gate}} = z^m$ in the forward pass and scales only the internal-reader gradient into the producer by α. The internal reader continues learning when $\alpha = 0$. The parameters of $\bar{\rho}$ and D_{ocr} stay frozen, while differentiation through their latent inputs continues to update the producer. Autoregressive prefixes remain connected across blocks. These properties separate learning to read the trajectory from admitting the internal reader's additional influence on its generation.

Paired prefix interventions. We assess dependence on generated prefixes by intervening on the producer's feedback and regenerating the subsequent blocks. Let B denote the number of ordered latent blocks, $\mathcal{D}_{\mathrm{eval}}$ a paired question–answer evaluation set, and $\mathcal{Q}_{\mathrm{eval}}$ its producer-question-mask distribution. A realization ω specifies $(\widetilde{x}_1, \ldots, \widetilde{x}_B)$, with $\widetilde{x}_1 = x$; native and intervened rollouts share this realization. Write $z^{\mathrm{ar}}(x; \omega) = \mathrm{concat}(\mathbf{z}_1, \ldots, \mathbf{z}_B)$ for the native trajectory, where $\mathbf{z}_b$ contains the latents of block b. At a boundary $b \geq 2$, the replacement prefixes are

$$
\mathbf{u}^{\mathrm{zero}}_{<b} = \mathbf{0}_{<b}, \qquad \mathbf{u}^{\mathrm{swap}}_{<b} = \mathbf{z}'_{<b}, \tag{18}
$$

Here z' is an unmodified, shape-matched trajectory from another problem. Because P is fixed and linear, zeroing feedback embeddings is equivalent to applying P to a zero latent prefix. Let W_θ denote the producer adapter, $\mathbf{r}_j$ the learned queries for block j, and $\Pi_{\mathbf{r}_j}$ their state selector. For $\mathcal{S} \in \{\mathrm{zero}, \mathrm{swap}\}$, the remaining blocks are regenerated as

$$
\widehat{\mathbf{z}}^{\mathcal{S}}_{j|b} = H\Big(\Pi_{\mathbf{r}_j} F^{W_\theta}\Big([\widetilde{x}_j; P([\mathbf{u}^{\mathcal{S}}_{<b}; \widehat{\mathbf{z}}^{\mathcal{S}}_{b:j-1|b}]); \mathbf{r}_j]\Big)\Big), \qquad j = b, \ldots, B. \tag{19}
$$

The generated portion is empty at $j = b$ and accumulates newly generated blocks thereafter. The donor prefix remains fixed throughout regeneration. The final readout receives

$$\bar{z}_b^{\mathcal{S}} := [\mathbf{z}_{<b}; \widehat{\mathbf{z}}_{b:B|b}^{\mathcal{S}}], \qquad \widehat{\mathbf{z}}_{b:B|b}^{\mathcal{S}} := \mathrm{concat}(\widehat{\mathbf{z}}_{b|b}^{\mathcal{S}}, \ldots, \widehat{\mathbf{z}}_{B|b}^{\mathcal{S}}). \tag{20}$$

Thus, replacement acts on the feedback copy of the prefix: the readout retains its native prefix and receives the regenerated suffix. Question tokens and latent queries are unchanged within each pair. The accuracy difference therefore measures how preceding blocks shape subsequent generation.

Measuring prefix dependence. Training-time readiness checks use the frozen external readout, whose answer prediction is

$$\widehat{y}_{\bar{\rho}}(z) := \arg\max_{y'} p_{\bar{\rho}}(y' \mid P(z), \tau_{\mathrm{ans}}). \tag{21}$$

The paired native and intervened accuracies are

$$\begin{aligned} A_{\bar{\rho}}^{\mathrm{native}} &= \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{eval}},\, \omega\sim\mathcal{Q}_{\mathrm{eval}}} \mathbf{1}[\widehat{y}_{\bar{\rho}}(z^{\mathrm{ar}}(x;\omega)) = y], \\ A_{\bar{\rho},b}^{\mathcal{S}} &= \mathbb{E}_{(x,y)\sim\mathcal{D}_{\mathrm{eval}},\, \omega\sim\mathcal{Q}_{\mathrm{eval}}} \mathbf{1}[\widehat{y}_{\bar{\rho}}(\bar{z}_b^{\mathcal{S}}(x;\omega)) = y]. \end{aligned} \tag{22}$$

For $B > 1$, their difference is averaged over all intervention boundaries:

$$\Lambda_{\mathcal{S}}^{\mathrm{ar}} = \frac{1}{B-1} \sum_{b=2}^{B} \left(A_{\bar{\rho}}^{\mathrm{native}} - A_{\bar{\rho},b}^{\mathcal{S}} \right), \qquad \mathcal{S} \in \{\mathrm{zero}, \mathrm{swap}\}, \tag{23}$$

Both empirical averages use the same examples, mask realizations, and donor assignments. Zero interventions measure reliance on the feedback path; donor-prefix interventions measure reliance on its problem-specific content. For training checks, $\mathcal{D}_{\mathrm{eval}} = \mathcal{D}_{\mathrm{check}}$ and $\mathcal{Q}_{\mathrm{eval}} = \mathcal{Q}_{\mathrm{check}}$ denote the fixed training-check set and its question-mask distribution.

Readiness-guided gradient admission. The parallel phase gradually introduces the internal reader's gradient into the producer. Autoregressive refinement resets α to zero, allowing the frozen supervision paths to first develop the blockwise producer. Let $\Lambda_{\mathcal{S}}^{\mathrm{ar}}(t)$ denote the effect at check t. Readiness requires both interventions to reach a positive threshold in two consecutive checks:

$$\min_{\mathcal{S}\in\{\mathrm{zero},\mathrm{swap}\}} \min_{k\in\{t-1,t\}} \Lambda_{\mathcal{S}}^{\mathrm{ar}}(k) \geq \delta_{\mathrm{dep}}, \qquad \delta_{\mathrm{dep}} > 0. \tag{24}$$

We evaluate this criterion every 500 steps on a fixed set of 200 training-side problems, disjoint from the benchmark test sets, with $\delta_{\mathrm{dep}} = 0.03$ (three percentage points). The reopening window spans steps 6,000–8,000 of autoregressive training. The ramp begins at the first check in this window satisfying the criterion, with step 8,000 as the latest scheduled start. Over the next 3,000 steps, α increases linearly from 0 to 1 and w decreases linearly from 1 to $w_{\min} = 0.5$.

Table 6 collects the complete internal-reader schedule. During parallel training, p_r takes values 1.0, 0.5, and 0.3 in successive epochs. The gate increases linearly from 0 to 1 during the second epoch as w decreases linearly from 1 to 0.5; both remain at their endpoint values in the third epoch. At the start of autoregressive refinement, $\alpha = 0$ and $w = 1$, with $p_r = 0.3$ throughout that phase. Refinement uses one block for its first 1,000 steps, then four blocks of eight latents, with producer masking $p_q = 0.5$ after the first block. Decoupled training uses $w = 1$ throughout and has no internal-reader gradient path. Epochs and steps are local to each phase.

Table 6: Unified-reader schedules. Epochs and steps are local to each phase.

Setting	Parallel	Autoregressive
Reader question mask p_r	1.0 / 0.5 / 0.3 by epoch	0.3
Gradient gate α	$0 \to 1$ in epoch 2	$0 \to 1$ over 3,000 steps
External weight w	$1 \to 0.5$ in epoch 2	$1 \to 0.5$ over 3,000 steps
Readiness checks	–	Every 500 steps; 200 problems
Reopening window	–	Steps 6,000–8,000

Inference. At inference, both question masks are disabled, $p_q = p_r = 0$, and the producer generates z^{ar} from x using its own preceding latent blocks. Decoupled CALR predicts from $p_{\bar{\rho}}(\cdot \mid P(z^{\mathrm{ar}}), \tau_{\mathrm{ans}})$; unified CALR predicts from $p_R(\cdot \mid x, P(z^{\mathrm{ar}}), \tau_{\mathrm{ans}})$ through its internal adapter in the same VLM. Answers are decoded greedily and compared with reference final answers. Rendering, the optical encoder, and reference derivations and latents serve training and representation evaluation; question-to-answer inference uses the question-conditioned producer and the chosen readout. D_{ocr} serves training and derivation evaluation, while $\bar{\rho}$ participates in answer prediction only in the decoupled realization.

B.2 Training Data

Training separates learning an optical representation of a derivation from learning to generate that representation from a question. The resampler uses the rendered derivations in Table 7; the external reader and both producer phases use GSM8K-Aug question–derivation–answer triples. MATH (Hendrycks et al., 2021) and MathX (Rawal, 2024) contribute only to reference-latent construction. Training examples are disjoint from the benchmark evaluation examples, including MATH-500.

Table 7: Training data and their roles in CALR.

Source	Training use	Examples
GSM8K-Aug	Resampler, reader, and producer	384,618
MATH	Resampler	11K
MathX-5M subset	Resampler	818K

Resampler inputs contain the derivation text, with the separate question and answer fields omitted. GSM8K-Aug derivations are rendered in their original and natural-language forms, yielding approximately 1.6M encoded pages across the combined corpus. A held-out set of 2,701 chains supports transcription validation. The resampler ablation in Table 5 is evaluated separately on the 1,319 GSM8K-Aug benchmark examples. Rendering and feature preparation are specified in Appendix C.1.

B.3 Evaluation Data

We evaluate on five mathematical reasoning benchmarks, from grade-school arithmetic to competition-level mathematics. Evaluation is zero-shot: the model receives the question without a reference derivation or worked examples. The reader and producer are trained on GSM8K-Aug; the remaining benchmarks evaluate transfer beyond this training distribution.

- **GSM8K-Aug.** Evaluation uses the 1,319 GSM8K test problems (Cobbe et al., 2021). These grade-school word problems require multi-step arithmetic and provide the in-distribution evaluation for the reader and producer. The same test set supports the diagnostic and ablation studies, with sampling specified in Appendix D.1.
- **GSM-Hard.** This benchmark (Gao et al., 2023) contains 1,319 problems constructed by replacing numbers in the GSM8K test set with substantially larger quantities. It preserves the underlying problem structure while testing arithmetic reasoning under a numerical distribution shift.
- **SVAMP.** Elementary arithmetic word problems are constructed through structural and linguistic variations of existing problems (Patel et al., 2021). The benchmark tests whether a model follows the mathematical relationships expressed in the question as its wording and structure change.
- **MultiArith.** This benchmark (Roy & Roth, 2015) consists of word problems requiring multiple arithmetic operations. It evaluates the composition of elementary operations into a coherent solution beyond the GSM8K-Aug training distribution.
- **MATH-500.** This 500-problem subset of the MATH test set (Lightman et al., 2024) evaluates competition-level mathematical reasoning. Its evaluation examples are excluded from reference-latent training, and the reader and producer receive no MATH training problems. It therefore tests transfer from grade-school problem solving to more demanding mathematics.

B.4 Base Models and Training Configurations

Model roles and optimization. The base model F supplies the producer and readout routes. Our principal configuration uses Qwen3-VL-4B-Instruct (Bai et al., 2025), with Qwen3-VL-2B-Instruct, Qwen3-VL-8B-Instruct, and LLaVA-V1.6-Mistral-7B (Liu et al., 2024) used in the cross-model evaluations. The complete results also include the Qwen3-4B language model (Yang et al., 2025) (Appendix E.1). DeepSeek-OCR2 supplies the fixed optical encoder E and decoder D_{ocr} used to construct and supervise the latent representation. Its optical feature dimension remains fixed, while the interface P maps into the input dimension of the selected base model.

Training uses PyTorch (Paszke et al., 2019), Transformers (Wolf et al., 2020), and PEFT (Mangrulkar et al., 2022) on eight NVIDIA H200 GPUs (141 GB each), with data parallelism and bfloat16 precision. All stages use AdamW (Loshchilov & Hutter, 2019), a peak learning rate of 10^{-4}, cosine decay after linear warm-up, and gradient clipping at 1.0. Optical page features are precomputed with the frozen encoder. Table 8 gives the stage-specific settings. The LoRA (Hu et al., 2022) adapters have rank 32 for the reader and rank 16 for the producer. In unified CALR, the internal reader inherits the Stage-I reader adapter, while its producer has the same configuration as in decoupled CALR.

Table 8: Per-stage training configurations. Rows marked "unified" apply only to unified CALR; the decoupled form uses the same shared settings with $\alpha = 0$, $w = 1$, and no internal reader.

Hyperparameter	Stage I: Representation	Stage I: Readout	Stage II: Drafting	Stage II: Refinement
Training data	Rendered chains (Appendix B.2)	GSM8K-Aug 385K	GSM8K-Aug 385K	GSM8K-Aug 385K
Optimizer / LR schedule	AdamW / cosine	AdamW / cosine	AdamW / cosine	AdamW / cosine
Peak learning rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}
Warm-up steps	300	200	360	360
Steps (epochs)	12,000 (≈ 1)	6,000 (2)	18,000 (3)	18,000 (3)
Global batch size	128	128	64	64
Adapter rank	–	32	16	16
Latents K / blocks B	32	32	32 / 1	32 / 4
Block-wise masking p_q	–	–	–	0.5 ($b > 1$)
OCR weight λ	–	–	$0.25 \to 0.05$	$0.25 \to 0.05 \to 0.01$
Reader derivation weight η	–	1	$0.5 \to 0.1$	1
Resampler segment weight β	0.5	–	–	–
AR block weight β_{ref}	–	–	–	1
Reader mask p_r (unified)	–	–	1.0 / 0.5 / 0.3 by epoch	0.3
Gradient gate α (unified)	–	–	$0 \to 1$ (epoch 2)	$0 \to 1$ (after admission)
External weight w (unified)	–	–	$1 \to 0.5$ (epoch 2)	$1 \to 0.5$ (after admission)
Trainable modules	Resampler ψ	Reader adapter, mode tokens	W_θ, $r_{1:K}$, tokens ($+R$ if unified)	Same as drafting
Frozen modules	E, D_{ocr}	P, base VLM	E, D_{ocr}, $\bar{\rho}$, P, H, base VLM	Same as drafting

385K denotes 384,618 examples. Token entries refer to trainable special-token embeddings; R has rank 32. Refinement uses one block for its first 1,000 steps before switching to four. Table 6 specifies the admission timing and ramp.

Functional anchoring. The resampler is trained for approximately one pass over the rendered corpus with $\beta = 0.5$, $\gamma = 0.01$, and target norm $\nu_0 = 6.5$ in Equation 11. Relative Gaussian noise of 5% is applied to latent states during this stage. We then fix the resampler and train the external reader for two epochs with $\eta = 1$. The interface scale s is calibrated to match optical-latent and token-embedding norms; U, P, and H remain fixed during reader and producer training. The trained reader is frozen as $\bar{\rho}$ for Stage II.

Parallel-to-autoregressive curriculum. Each producer phase runs for approximately three epochs over GSM8K-Aug. In parallel training, λ decreases linearly from 0.25 to 0.05 during the first epoch and then remains fixed. The external-reader derivation weight η is 0.5 in the first epoch, decreases to

0.1 during the second, and remains at 0.1 in the third. Autoregressive training warm-starts from the parallel-trained producer. It uses one block for the first 1,000 steps, followed by four blocks of eight latents; question masking with $p_q = 0.5$ applies after the first block. Its OCR weight follows the three-epoch schedule $0.25 \to 0.05 \to 0.01$, with $\eta = 1$ for the external reader. Equation 16 combines full-trajectory supervision from D_{ocr} with block-level derivation recovery by the frozen external reader. We set $\beta_{\mathrm{ref}} = 1$, so the block-level term follows the same λ schedule as the full-trajectory OCR term; the reader's full-trajectory derivation loss retains $\eta = 1$. Gradients remain connected across the generated prefix.

B.5 General Evaluation Settings

Pass@1 and Acc. denote final-answer accuracy in percent. Reasoning length #L counts intermediate text tokens or latent states, excluding the question and final answer. CALR uses 32 such states, while direct-answer methods have zero intermediate length. Average columns are unweighted means over the listed benchmarks; the Pass@1/#L column divides the displayed mean accuracy by the displayed mean reasoning length, with a dash when the latter is zero. Character similarity and decoder cross-entropy evaluate reconstruction of a reference derivation, separately from answer accuracy (Appendix D.3).

C Implementation Details of CALR

C.1 Rendering and Information-Balanced Resampling

Reference derivations are rendered in a monospaced font, with font sizes sampled from a Gaussian distribution truncated to 15–20 points. Text wraps at 70 characters on a 1,024-pixel-wide page whose height adapts to its content. The frozen DeepSeek-OCR2 encoder produces $N = 256$ spatial features of dimension $d = 1280$; its decoder also remains fixed throughout anchoring.

Index $V_1, \ldots, V_N$ in the encoder's spatial sequence order. Let $A(t) = \sum_{i=1}^{t} a_i$ be the cumulative feature-surprisal score, with $A(0) = 0$. The segmentation in Equation 10 takes the form

$$0 = t_0 < t_1 < \cdots < t_K = N, \qquad S_g = \{t_{g-1} + 1, \ldots, t_g\}, \qquad A(t_g) \approx \frac{g}{K} A(N). \tag{25}$$

The feature statistics remain fixed across training, while boundaries are recomputed for each rendered derivation. The score is the nonconstant quadratic component of diagonal-Gaussian negative log likelihood. Equal cumulative score allocates fewer spatial positions to information-dense regions and more positions to sparse regions.

The resampler contextualizes the complete feature sequence through four self-attention layers. A shared query then cross-attends to the contextualized features within each S_g, using an intra-segment position bias. Each output therefore represents its designated region with access to global page context.

C.2 Supervision Targets and Canvas Construction

Spatial segmentation and text-target construction are separate operations. Spatial segments follow the feature-surprisal scores above; text spans are formed from paragraph and punctuation boundaries in the reference derivation. These spans retain their reading order when assigned as supervision for latent segments. We write c_g for the target associated with latent position g. For a block of positions $\mathcal{B}_b$, its target is $c_{\mathcal{B}_b} = \mathrm{concat}_{g \in \mathcal{B}_b} c_g$.

The optical latent dimension matches the decoder's visual feature dimension, $d_z = d$. Let $V^{\mathrm{blank}} \in \mathbb{R}^{N \times d_z}$ be the features of a rendered blank page, and let $\pi : \{1, \ldots, K\} \to \{1, \ldots, N\}$ denote the fixed injective placement of latent positions on the decoder's visual grid. For a subset J of latent positions, define

$$[\mathrm{Canvas}_J(z_J)]_i = \begin{cases} z_g, & i = \pi(g), \quad g \in J, \\ V_i^{\mathrm{blank}}, & i \notin \pi(J). \end{cases} \tag{26}$$

The complete, single-latent, and block canvases take $J = \{1, \ldots, K\}$, $J = \{g\}$, and $J = \mathcal{B}_b$, respectively; we write $\mathrm{Canvas}(z)$ for the complete canvas. Partial canvases retain absolute positions

under the same placement map. Unused positions contain blank-page features rather than zero vectors.

During Stage I, the frozen DeepSeek-OCR2 decoder supervises both the complete reference canvas and single-latent canvases, together with norm regularization. During Stage II, full-trajectory OCR supervision is retained in both producer phases, while autoregressive refinement additionally uses the frozen external reader to recover aligned derivation spans from block canvases. The reader's full-trajectory answer and derivation losses remain active in both phases. The optical readout $\widehat{c}(z)$ is used to evaluate C2, while $\mathrm{Read}_{\mathrm{out}}$ provides additional block-level derivation supervision during autoregressive training.

C.3 Fixed Interfaces

The maps $P : \mathbb{R}^{d_z} \to \mathbb{R}^{d_{\mathrm{emb}}}$ and $H : \mathbb{R}^{d_{\mathrm{emb}}} \to \mathbb{R}^{d_z}$ act position-wise. We construct U by applying a reduced QR decomposition to a randomly initialized matrix $M \in \mathbb{R}^{d_{\mathrm{emb}} \times d_z}$:

$$M = UT, \qquad U^\top U = I_{d_z},$$

where $T \in \mathbb{R}^{d_z \times d_z}$ is upper triangular. We retain the orthonormal factor U and fix it throughout reader and producer training. With a positive calibration scale s, the interfaces satisfy

$$\begin{aligned} P &= sU, \qquad H = s^{-1}U^\top, \\ HP &= I_{d_z}, \qquad PH = UU^\top. \end{aligned} \tag{27}$$

Thus, H recovers the original coordinates of an embedded optical latent, while PH projects a VLM state onto the interface subspace. In both producer phases, H converts selected query states into optical latents. The OCR canvas receives z directly, both readout routes and the producer feedback route receive $P(z)$. The interfaces remain fixed during training. For block-canvas supervision, $\mathrm{Read}_{\mathrm{out}}$ includes the same fixed reader-input interface and the derivation mode τ_{cot}.

D Diagnostic and Ablation Protocols

D.1 Sampling and Answer-Input Interventions

Figure 2 and Tables 11–12 use the same fixed subset of 300 problems sampled at random from the 1,319 GSM8K-Aug evaluation problems. RoT uses the released model, DLR-SFT uses our reproduction, and CALR uses the unified realization. CALR percentages average repeated evaluations on this subset, including sampled replacements for stochastic interventions. The remaining diagnostic and ablation evaluations use the complete GSM8K-Aug evaluation set.

For a dataset $\mathcal{D}$, let A_{base} denote accuracy with each problem's native latents. An intervention ι replaces $G_\theta(x)$ by a possibly stochastic state $z^\iota \sim \iota(G_\theta(x))$ while keeping the answer model fixed. We define

$$\begin{aligned} A_{\mathrm{base}} &= \Pr_{(x,y)\sim\mathcal{D}}\left[\arg\max_{y'} p_\phi(y' \mid x, G_\theta(x)) = y\right], \\ A_\iota &= \mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbb{E}_{z^\iota\sim\iota(G_\theta(x))}\mathbf{1}\left[\arg\max_{y'} p_\phi(y' \mid x, z^\iota) = y\right], \\ \Lambda_\iota(z \to \hat{y}) &= A_{\mathrm{base}} - A_\iota. \end{aligned} \tag{28}$$

We apply two intervention families. *Content interventions* preserve the latent interface but replace its native trajectory: I_{swap} supplies another problem's latents, and I_{const} supplies one fixed latent sequence to every problem. *Interface interventions* disrupt the channel itself: I_{rand} supplies norm-matched random states, and I_{del} removes the latent slots. Writing $\Lambda_{\mathrm{swap}} := \Lambda_{I_{\mathrm{swap}}}$ and analogously for the remaining interventions, their aggregate effects are

$$\begin{aligned} \mathcal{I}_{\mathrm{content}} &= \{I_{\mathrm{swap}}, I_{\mathrm{const}}\}, \qquad \mathcal{I}_{\mathrm{interface}} = \{I_{\mathrm{rand}}, I_{\mathrm{del}}\}, \\ \Lambda_{\mathrm{content}} &= \tfrac{1}{2}(\Lambda_{\mathrm{swap}} + \Lambda_{\mathrm{const}}), \qquad \Lambda_{\mathrm{interface}} = \tfrac{1}{2}(\Lambda_{\mathrm{rand}} + \Lambda_{\mathrm{del}}). \end{aligned} \tag{29}$$

The deletion effect Λ_{del} directly operationalizes C1 at the accuracy level, while $\Lambda_{\mathrm{interface}}$ summarizes the model's broader sensitivity to disruption of the latent input. Content interventions test whether an

active interface carries problem-specific information, while decoded-derivation validity evaluates C2. As a control on the competing direct route, we withhold x from the answer model while retaining the latents generated from the original question,

$$A_{-x} = \Pr_{(x,y)\sim\mathcal{D}}\left[\arg\max_{y'} p_\phi(y' \mid \varnothing_x, G_\theta(x)) = y\right], \qquad \Lambda_x = A_{\text{base}} - A_{-x}. \tag{30}$$

Each native trajectory is generated from its own question before modifying the answer model's inputs. DLR-SFT uses six pooled latent codes in these input interventions.

When accuracies are expressed as probabilities, intervention effects are reported in percentage points as 100Λ. The unchanged-answer rate compares each swapped prediction with the corresponding native prediction. For an evaluation with n examples,

$$U_{\text{swap}} = \frac{100}{n}\sum_{i=1}^{n} \mathbf{1}[\widehat{y}_i^{\text{swap}} = \widehat{y}_i^{\text{base}}]. \tag{31}$$

The reported rate averages these comparisons over evaluations, independently of whether the shared answer is correct.

D.2 Attention and Gradient Measurements

Let $a_z^{(i)}$ and $a_x^{(i)}$ denote the answer-token attention masses assigned to latent and question positions for example i. Table 12 reports the mean of the per-example normalized latent shares,

$$\overline{a}_{\text{latent}} = \frac{100}{n}\sum_{i=1}^{n} \frac{a_z^{(i)}}{a_z^{(i)} + a_x^{(i)}}. \tag{32}$$

Normalization includes the question and latent groups and precedes averaging over examples. Figure 2 instead presents group means without renormalizing over question and latent positions; Table 9 collects these values alongside the gradient measurements. Appendix E.4 provides complementary answer-step-level attention maps.

For the answer loss $\ell_{\text{ans}} = -\log p_\phi(y \mid x, z)$, let e_x denote the question-input embeddings. We compute

$$\overline{g} = \frac{1}{n}\sum_{i=1}^{n} \frac{\|\nabla_z \ell_{\text{ans}}^{(i)}\|_2}{\|\nabla_{e_x} \ell_{\text{ans}}^{(i)}\|_2}. \tag{33}$$

For discrete latents, differentiation is with respect to their continuous input embeddings. Each norm includes all positions and feature dimensions in its input group.

Table 9: Answer-stage attention and gradient measurements.

System	Question attention	Latent attention	Latent/question gradient ratio
RoT	0.51	0.33	1.12
DLR-SFT	0.72	0.05	0.13

D.3 Decoded Chains and arithmetic-step recovery

All systems in Table 13 are decoded with the frozen DeepSeek-OCR2 decoder, which receives their optical latent representations without question text. Let $\widehat{c}_i$ and c_i denote the decoded and reference chains. Character similarity uses Python's `difflib.SequenceMatcher.ratio()`, which recursively matches contiguous character blocks. If $M(\widehat{c}_i, c_i)$ counts the matched characters, the score is

$$\text{Sim}(\widehat{c}_i, c_i) = \frac{2M(\widehat{c}_i, c_i)}{|\widehat{c}_i| + |c_i|}. \tag{34}$$

Lengths are character counts, and scores are averaged over problems.

For arithmetic-step recovery, we use each reference result in c_i to locate candidate passages in $\widehat{c}_i$, then verify the corresponding operands and arithmetic relation within the local clause or sentence. Adjacent

context and previously matched results help resolve explicit references. Matching normalizes numeric formatting and arithmetic symbols and allows natural-language expressions, reversed equalities, and commutative operand order. Thus, $4 + 24 = 28$ and "adding 4 to 24 gives 28" match the same step, whereas merely mentioning 28 does not. Let m_i denote the number of arithmetic steps in the reference chain c_i, and let $\mathcal{A}_i$ be a maximum-cardinality one-to-one matching between the arithmetic steps in c_i and those expressed in $\widehat{c}_i$ under these rules. We compute

$$\mathrm{Rec}(\widehat{c}_i, c_i) = \frac{|\mathcal{A}_i|}{m_i}. \tag{35}$$

Scores are averaged across problems. The *same-problem* evaluation compares $\widehat{c}_i$ with c_i, whereas the *other-problem* evaluation holds $\widehat{c}_i$ fixed and applies the same procedure to another problem's reference $c_{j(i)}$, where $j(i) \neq i$. Answer accuracy is evaluated under the common answer-evaluation setting used by the three systems and reported separately from chain-content metrics.

D.4 Generation-Feedback Evaluation

Final diagnostics use the paired feedback construction and boundary-averaged effect in Appendix B.1, with all 1,319 GSM8K-Aug problems and question masks disabled. Each system is evaluated through its prescribed answer route, using the same readout within each native–intervened pair.

Table 15 compares the autoregressively trained producer with the parallel-trained control under the same four-block rollout, with $K = 32$ latents in four blocks of eight. Zero replaces feedback embeddings with zeros. Shuffle implements swap by a derangement across the example dimension within a rollout microbatch of 16: each problem receives another problem's prefix while latent-position order is preserved. Question tokens and latent queries remain unchanged within each pair, and all subsequent blocks are generated freely.

The all-latent view reads the complete 32-latent trajectory; the last-block view supplies only the final eight generated latents. Each effect is measured relative to the native accuracy for that view and averaged over intervention boundaries, as in Equation 23. The last-block comparison reads $\mathbf{z}_B$ and $\widehat{\mathbf{z}}^{S}_{B|b}$ under the same readout protocol.

D.5 Frozen-Reader Evaluation

Table 14 fixes the external readout $\bar{\rho}$ and withholds the question. Its four inputs are the raw DeepSeek-OCR2 features V of the rendered reference derivation, their resampled reference trajectory $z^\star$, the problem-conditioned output $G_\theta(x)$ of the unified-trained CALR producer, and blank-page features V^{blank}. Each optical feature sequence passes through the same fixed interface P. The 62.5% accuracy therefore measures the unified-trained producer's performance through the external reader $\bar{\rho}$.

D.6 Component Ablations

Training components. Table 4 evaluates unified CALR on GSM8K-Aug with 32 latents. Every variant retains the internal reader. External supervision uses the frozen question-blind readout, and the reader-mask column specifies whether p_r is active in the current training phase. Autoregressive variants warm-start from externally supervised parallel training. A dash in the AR-supervision column indicates that the configuration contains only the parallel phase.

Resampling strategies. Table 5 compares three ways to resample optical features from rendered reference derivations. Global cross-attention operates on the complete feature sequence; fixed anchors assign equal feature counts to each output latent; dynamic anchors adjust segment widths to balance cumulative information. The trained representations are decoded by DeepSeek-OCR2 on GSM8K-Aug. Similarity uses Equation 34, reported in percent; CE is decoder cross-entropy against the reference derivation.

Output and feedback interfaces. In Table 16, CALR uses the fixed maps $P = sU$ and $H = s^{-1}U^\top$, with $U^\top U = I_{d_z}$ as in Equation 12. The second configuration replaces H with a trainable MLP and uses a frozen copy of the reader projection for feedback. The third trains both the output head and the feedback projection. The producer parameters W_θ remain trainable

in all three configurations. Each uses four-block autoregressive rollouts and the paired evaluation in Appendix D.4, with interventions applied to the actual feedback embeddings after projection. Accuracies and prefix effects use the same readout protocol across interface configurations.

E Further Experiments and Visualization Analysis

E.1 Complete Benchmark Results

Table 10 extends Table 1 with the Qwen3-4B language model, iCoT, and DLR's reinforcement-learning variant. All CALR configurations use supervised training.

Table 10: Complete results on four grade-school reasoning benchmarks. Acc. denotes accuracy (%), and #L denotes reasoning length. Average is the unweighted mean across the four benchmarks.

	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average	
Method	Acc.	#L	Acc.	#L	Acc.	#L	Acc.	#L	Acc.	#L
Qwen3-4B (LLM)										
Direct-SFT	$27.4_{\pm.4}$	$0.0_{\pm.0}$	$9.2_{\pm.2}$	$0.0_{\pm.0}$	$69.4_{\pm.4}$	$0.0_{\pm.0}$	$86.3_{\pm.3}$	$0.0_{\pm.0}$	48.1	0.0
CoT-SFT	$83.1_{\pm.3}$	$119.8_{\pm.8}$	$52.3_{\pm.3}$	$189.4_{\pm1.1}$	$82.9_{\pm.3}$	$53.7_{\pm.7}$	$98.5_{\pm.3}$	$57.6_{\pm.8}$	79.2	105.1
iCoT	$13.5_{\pm.2}$	$0.0_{\pm.0}$	$4.1_{\pm.2}$	$0.0_{\pm.0}$	$36.9_{\pm.2}$	$0.0_{\pm.0}$	$49.2_{\pm.7}$	$0.0_{\pm.0}$	25.9	0.0
Coconut (CoLM 2025)	$16.9_{\pm.3}$	$6.0_{\pm.0}$	$5.4_{\pm.3}$	$6.0_{\pm.0}$	$43.6_{\pm.5}$	$6.0_{\pm.0}$	$60.3_{\pm.7}$	$6.0_{\pm.0}$	31.6	6.0
CODI (EMNLP 2025)	$7.3_{\pm.5}$	$6.0_{\pm.0}$	$2.2_{\pm.2}$	$6.0_{\pm.0}$	$11.0_{\pm.6}$	$6.0_{\pm.0}$	$18.3_{\pm.8}$	$6.0_{\pm.0}$	9.7	6.0
CoLaR (NeurIPS 2025)	$40.0_{\pm.2}$	$39.6_{\pm.1}$	$9.2_{\pm.1}$	$47.4_{\pm.2}$	$57.7_{\pm.2}$	$19.2_{\pm.1}$	$82.2_{\pm.1}$	$21.1_{\pm.1}$	47.3	31.8
DLR (SFT)	$47.4_{\pm.4}$	$6.4_{\pm.3}$	$13.7_{\pm.3}$	$6.8_{\pm.2}$	$70.3_{\pm.2}$	$3.6_{\pm.3}$	$96.1_{\pm.3}$	$4.2_{\pm.1}$	56.9	5.3
DLR (RL)	$62.2^{*}_{\pm.4}$	$6.5_{\pm.2}$	$\mathbf{19.3}^{*}_{\pm.3}$	$7.0_{\pm.3}$	$72.3^{*}_{\pm.2}$	$3.8_{\pm.2}$	$97.8^{*}_{\pm.3}$	$4.3_{\pm.3}$	62.9	5.4
Ours, decoupled (Par.)	$49.8_{\pm.3}$	$32.0_{\pm.0}$	$14.5_{\pm.4}$	$32.0_{\pm.0}$	$70.1_{\pm.2}$	$32.0_{\pm.0}$	$96.8_{\pm.2}$	$32.0_{\pm.0}$	57.8	32.0
Ours, unified (Par.)	$51.2_{\pm.3}$	$32.0_{\pm.0}$	$15.1_{\pm.3}$	$32.0_{\pm.0}$	$71.5_{\pm.2}$	$32.0_{\pm.0}$	$97.4_{\pm.1}$	$32.0_{\pm.0}$	58.8	32.0
Ours, decoupled (AR)	$60.7_{\pm.2}$	$32.0_{\pm.0}$	$16.9_{\pm.3}$	$32.0_{\pm.0}$	$74.8_{\pm.3}$	$32.0_{\pm.0}$	$98.0_{\pm.2}$	$32.0_{\pm.0}$	62.6	32.0
Ours, unified (AR)	$\mathbf{62.3}_{\pm.3}$	$32.0_{\pm.0}$	$17.6_{\pm.3}$	$32.0_{\pm.0}$	$\mathbf{76.2}_{\pm.2}$	$32.0_{\pm.0}$	$\mathbf{98.8}_{\pm.1}$	$32.0_{\pm.0}$	**63.7**	32.0
Qwen3-VL-4B-Instruct (LMM)										
Direct-SFT	$26.2_{\pm.3}$	$0.0_{\pm.0}$	$9.5_{\pm.1}$	$0.0_{\pm.0}$	$70.0_{\pm.3}$	$0.0_{\pm.0}$	$85.6_{\pm.4}$	$0.0_{\pm.0}$	47.8	0.0
CoT-SFT	$81.2_{\pm.4}$	$127.3_{\pm1.0}$	$53.4_{\pm.3}$	$191.1_{\pm1.6}$	$84.3_{\pm.3}$	$55.9_{\pm.6}$	$98.3_{\pm.4}$	$59.1_{\pm.6}$	79.3	108.4
Coconut (CoLM 2025)	$16.1_{\pm.4}$	$6.0_{\pm.0}$	$5.2_{\pm.3}$	$6.0_{\pm.0}$	$42.0_{\pm.3}$	$6.0_{\pm.0}$	$58.9_{\pm.2}$	$6.0_{\pm.0}$	30.6	6.0
CODI (EMNLP 2025)	$7.1_{\pm.3}$	$6.0_{\pm.0}$	$2.0_{\pm.4}$	$6.0_{\pm.0}$	$10.7_{\pm.1}$	$6.0_{\pm.0}$	$17.8_{\pm.2}$	$6.0_{\pm.0}$	9.4	6.0
CoLaR (NeurIPS 2025)	$39.2_{\pm.2}$	$38.8_{\pm.5}$	$8.2_{\pm.4}$	$47.8_{\pm.7}$	$52.0_{\pm.4}$	$18.4_{\pm.6}$	$81.0_{\pm.3}$	$22.1_{\pm.3}$	45.1	31.8
RoT (ACL 2026)	$37.8_{\pm.3}$	$32.0_{\pm.0}$	$14.1_{\pm.2}$	$32.0_{\pm.0}$	$72.7_{\pm.4}$	$32.0_{\pm.0}$	$97.2_{\pm.4}$	$32.0_{\pm.0}$	55.5	32.0
DLR (SFT)	$48.0_{\pm.4}$	$6.3_{\pm.4}$	$14.7_{\pm.3}$	$6.8_{\pm.5}$	$68.7_{\pm.2}$	$3.6_{\pm.1}$	$97.8_{\pm.3}$	$4.2_{\pm.2}$	57.3	5.2
DLR (RL)	$63.3^{*}_{\pm.3}$	$6.5_{\pm.1}$	$\mathbf{18.3}^{*}_{\pm.4}$	$7.0_{\pm.6}$	$72.7_{\pm.3}$	$3.8_{\pm.4}$	$98.3^{*}_{\pm.2}$	$4.3_{\pm.3}$	63.2	5.4
Ours, decoupled (Par.)	$51.2_{\pm.2}$	$32.0_{\pm.0}$	$13.8_{\pm.3}$	$32.0_{\pm.0}$	$71.3_{\pm.2}$	$32.0_{\pm.0}$	$97.6_{\pm.1}$	$32.0_{\pm.0}$	58.5	32.0
Ours, unified (Par.)	$52.4_{\pm.3}$	$32.0_{\pm.0}$	$14.3_{\pm.2}$	$32.0_{\pm.0}$	$72.4_{\pm.1}$	$32.0_{\pm.0}$	$98.1_{\pm.1}$	$32.0_{\pm.0}$	59.3	32.0
Ours, decoupled (AR)	$62.1_{\pm.2}$	$32.0_{\pm.0}$	$16.1_{\pm.2}$	$32.0_{\pm.0}$	$76.6_{\pm.3}$	$32.0_{\pm.0}$	$98.8_{\pm.1}$	$32.0_{\pm.0}$	63.4	32.0
Ours, unified (AR)	$\mathbf{63.8}_{\pm.2}$	$32.0_{\pm.0}$	$16.8_{\pm.3}$	$32.0_{\pm.0}$	$\mathbf{77.4}_{\pm.1}$	$32.0_{\pm.0}$	$\mathbf{99.7}_{\pm.1}$	$32.0_{\pm.0}$	**64.4**	32.0

Bold marks the highest accuracy among latent reasoning methods within each model group. Blue and green shading distinguish decoupled and unified CALR. Par. and AR denote parallel drafting and autoregressive refinement, respectively. Asterisks retain the significance markers from DLR's original comparison.

E.2 Analysis of Latent Reasoning

We examine whether CALR makes latent reasoning both load-bearing for the answer and specific to the problem, then study how autoregressive training and fixed interfaces support this behavior. Accuracies are percentages and intervention effects are percentage-point drops. Appendix D gives the measurement and evaluation details.

Latent reliance. Table 11 extends the preliminary interventions to unified CALR. Its native accuracy is 64.3%, but replacing or removing its latents reduces accuracy to 0.8–1.7%, with the question still available. Under cross-problem replacement, only 1.3% of its answers remain unchanged, compared

with 98% for RoT and 93% for DLR-SFT. CALR therefore depends on the trajectory associated with the current problem, rather than merely the presence of latent slots. Removing the answer-side question instead preserves 61.9% accuracy, a drop of only 2.4 points. Together, these interventions establish the load-bearing role of CALR's latents (C1) and show that they retain the information needed for answer prediction.

Table 12 characterizes the corresponding answer-stage interaction. CALR allocates 69% of its attention over question and latent inputs to the latents, versus 39% for RoT and 6% for DLR-SFT. Its latent-to-question gradient ratio is 0.26, between DLR-SFT's 0.13 and RoT's 1.12. RoT thus has the largest relative latent gradient, while CALR has the strongest dependence on problem-specific latent content. The intervention and attention measurements together identify an answer process that actively uses the generated trajectory.

Table 11: Answer-input interventions (%). U_{swap}: unchanged-answer rate under swap; No x: answer-side question removed.

		Content		Interface			
System	Base	Swap	Const.	Rand.	Del.	No x	U_{swap}
RoT	43.0	42.0	43.0	1.0	1.0	2.0	98.0
DLR-SFT	53.0	53.0	53.0	52.0	51.0	1.0	93.0
CALR	**64.3**	1.4	1.1	1.7	0.8	**61.9**	**1.3**

Table 12: Answer-stage latent attention and latent-to-question gradient-norm ratios.

System	Latent attn. (%)	Gradient ratio
RoT	39	1.12
DLR-SFT	6	0.13
CALR	**69**	0.26

Problem-specific reasoning. We decode each system's latents with DeepSeek-OCR2 to examine the reasoning content they carry (Table 13). CALR recovers 74% of the current problem's intermediate values, compared with 7% when the same decoded text is matched against another problem. Its full-chain character similarity reaches 0.34, compared with 0.07 for RoT and 0.09 for our DLR-SFT reproduction, while answer accuracy reaches 63.5%. DLR-SFT likewise carries problem-specific intermediate values (0.53 versus 0.13), whereas RoT's recovery is 0.01 in both conditions. This contrast connects the decoded reasoning content to C2 while preserving the distinction from C1: DLR-SFT represents intermediate information that its answer model largely bypasses; CALR both represents and uses it.

Table 14 examines the fidelity of the anchored representation through the frozen external reader. Raw optical features support 98.9% accuracy, and resampled reference latents retain 96.7%, a difference of 2.2 points. Latents from the unified-trained producer yield 62.5%, compared with 0.5% for a blank page. The reference results establish that the compressed latent representation preserves the answer information needed by the frozen reader. The gap between reference and generated latents thus locates the principal performance bottleneck in problem-conditioned latent generation.

Table 13: Decoded reasoning: character similarity, same/other-problem arithmetic-step recovery, and answer accuracy (%). * denotes our DLR-SFT reproduction.

	Chain	Value recovery		Answer
System	similarity	Same	Other	acc.
RoT	0.07	0.01	0.01	38.2
DLR-SFT*	0.09	0.53	0.13	44.0
CALR	**0.34**	**0.74**	0.07	**63.5**

Table 14: Frozen external-reader accuracy (%) across input representations.

Source	State	Acc.
Raw	V	98.9
Resampled	$z^\star$	96.7
Generated	$G_\theta(x)$	**62.5**
Blank	V^{blank}	0.5

Table 15: Feedback interventions during generation. Zero and Swap denote zeroing and cross-problem replacement; Λ^{ar} is the accuracy drop.

		Accuracy (%)			Effect (pp)	
Producer	View	Base	Zero	Swap	$\Lambda^{\mathrm{ar}}_{\mathrm{zero}}$	$\Lambda^{\mathrm{ar}}_{\mathrm{swap}}$
CALR	All 32	**63.8**	38.7	45.4	+25.1	+18.4
	Last 8	28.2	6.1	9.8	+22.1	+18.4
Parallel	All 32	51.1	51.3	51.0	−0.2	+0.1

Table 16: Output (H) and feedback (P) interface ablations. Fixed/tied denotes CALR; MLP is trainable.

			Effect (pp)	
H	P	Acc.	$\Lambda^{\mathrm{ar}}_{\mathrm{zero}}$	$\Lambda^{\mathrm{ar}}_{\mathrm{swap}}$
Fixed	Tied	**63.8**	**+25.1**	**+18.4**
MLP	Fixed copy	59.2	+15.2	+13.3
Learned	Learned	56.33	+6.2	+3.5

Autoregressive dependence. Table 15 tests whether preceding latent blocks shape subsequent generation. Zeroing or replacing their feedback reduces CALR's accuracy by 25.1 or 18.4 points. The parallel-trained producer, evaluated under the same four-block rollout, remains near its 51.1% baseline under both interventions. Autoregressive training thus develops a dependence on preceding latent content. This dependence persists when the reader receives only the final block: accuracy falls from 28.2% to 6.1% or 9.8%. Earlier blocks therefore influence what the final block represents, beyond their direct contribution to the reader. Together with the decoded intermediate values, these results show that CALR organizes problem-specific reasoning into a dependent latent sequence.

The role of a fixed interface. Table 16 compares output and feedback mappings. The fixed, tied scaled-isometric interface achieves 63.8% accuracy with zero/swap effects of 25.1/18.4 points. Training the output map reduces accuracy to 59.2% and the effects to 15.2/13.3; training both maps further reduces them to 56.33% and 6.2/3.5. A stable interface therefore supports both answer accuracy and prefix dependence. Together, these analyses show that CALR couples answer reliance with problem-specific reasoning, which autoregressive refinement develops through successive latent blocks.

E.3 Intermediate-State Substitution

We test whether information carried by an intermediate latent block is used in subsequent generation by replacing it with a block generated for another problem. We pair the original problem x with a donor x' that has the same final answer, $y' = y$, but a different first intermediate quantity, $u'_1 \neq u_1$. Let F_x denote the composition of the original problem's remaining operations. We select pairs for which applying these operations to the donor's intermediate quantity yields an expected answer distinct from their shared final answer:

$$y_{\mathrm{exp}} = F_x(u'_1), \qquad y_{\mathrm{exp}} \neq y = y'. \tag{36}$$

Producing y_{exp} therefore requires a result distinct from either problem's original final answer.

With the original question visible to the producer, we replace the second latent block in its feedback with the donor's corresponding block and regenerate the third and fourth blocks. The frozen question-blind external reader receives the original first two blocks followed by the regenerated suffix; the reader-only control instead replaces the second block directly at readout, retaining all other original blocks without regeneration. A further control uses donors with the same intermediate quantity but a different final answer, while self-replacement uses the original block in place of a donor block.

Table 17: Intermediate-state substitution on two disjoint sets of 120 problems each. Expected denotes the percentage of outputs matching y_{exp}; Orig. denotes accuracy against the original answer y. Regeneration retains the original first two blocks at readout. All values are percentages.

Condition	Metric	Set 1	Set 2
Feedback substitution + suffix regeneration	Expected	75.0	80.0
Reader-input substitution only	Expected	0.8	0.8
Same intermediate, different answer	Orig.	87.5	82.5

Results. Table 17 shows that feedback substitution followed by suffix regeneration produces the expected answer in 75.0% and 80.0% of cases across the two sets, respectively, with the original first

two blocks retained at readout. Replacing the second block only at the reader input yields the expected answer in 0.8% of cases on each set. Donors with the same intermediate quantity but different final answers preserve original-answer accuracy at 87.5% and 82.5%, matching the self-replacement controls. These results show that substituted intermediate information shapes subsequent latent generation, with its effect reaching the reader through the regenerated suffix.

E.4 Per-Example Attention Maps

Figure 4 complements the aggregate attention measurements in Figure 2(a) with answer-step-level views of a test question. Each row corresponds to one generated answer token (a_t), and columns separate question tokens from latent positions. Attention is averaged over heads and layers, and normalized row-wise as shown by the attention share color map. The continuous trajectory contains 32 latent slots, whereas the discrete trajectory consists of 6 code slots.

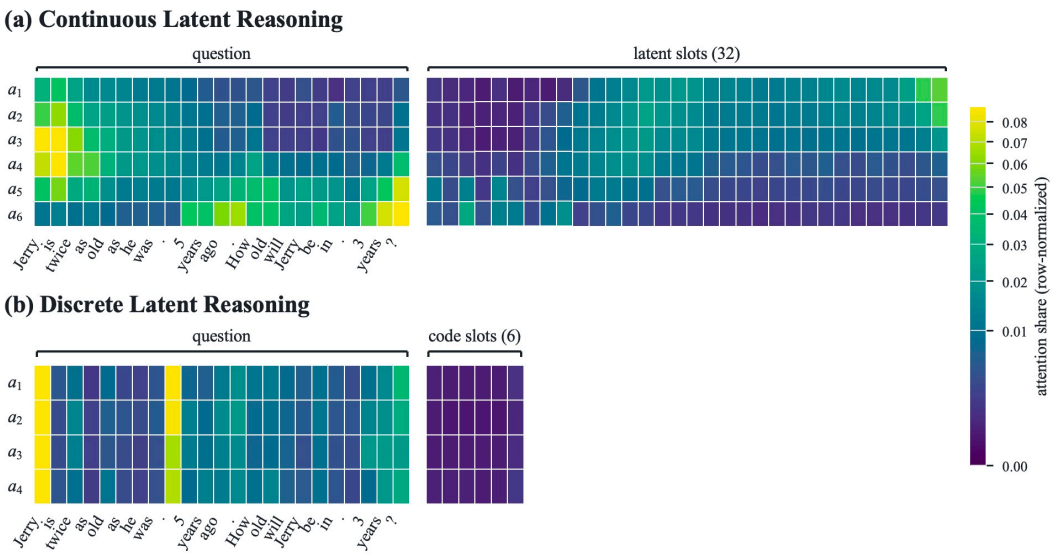

Figure 4: Per-example answer-stage attention in continuous and discrete latent reasoning. Rows represent generated answer steps (a_t); columns separate question tokens from latent/code slots. Both panels share the same row-normalized attention share color map.

In this example, the continuous system distributes attention across both the question and latent slots, whereas the discrete system concentrates attention almost exclusively on specific question tokens (leaving code slots with near-zero attention). Read together with the content interventions in Figure 2(b), the maps separate the allocation of attention from reliance on problem-specific latent content: attending to a latent trajectory does not establish that its content determines the answer. The combined evidence motivates assessing both latent reliance (C1) and reasoning capability (C2).

E.5 Decoded Continuous Latent Reasoning

We next examine the reasoning content of individual CALR trajectories. Figures 5 and 6 compare reference derivations with frozen DeepSeek-OCR2 transcriptions of 32 generated latents, produced in four autoregressive blocks. The decoder receives the latent canvas without the question. Each case also reports the answer obtained separately from the question-blind external readout, linking visible intermediate content with answer recovery from the same trajectory.

Recovering intermediate quantities. Figure 5 exposes problem-specific values preceding the final answer. In Case #133, the decoded trajectory recovers the 24-mile distance and the 28-mile subtotal before obtaining 13 miles. Cases #118 and #34 likewise recover the reference sequences $10, 20, 4$ and $20, 25, 23$. The agreement spans successive intermediate results, making the numerical-recovery measurement in Table 13 visible at the case level. Across the six cases, the transcriptions retain these numerical intermediates while exhibiting discrepancies in entity names, units, temporal labels, and

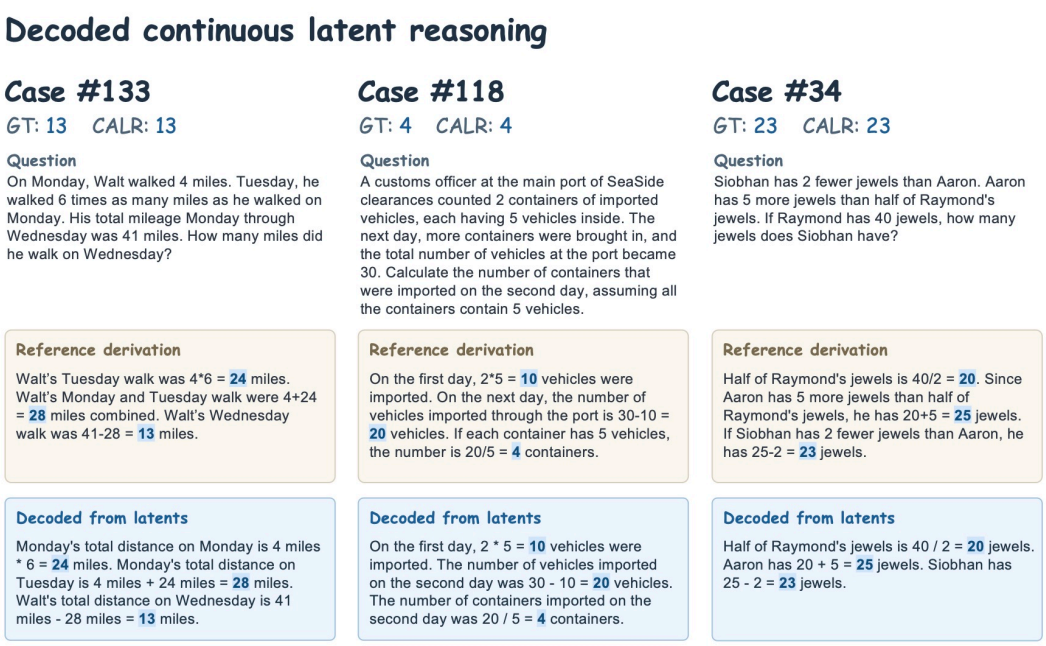

Figure 5: Decoded CALR trajectories for Cases #133, #118, and #34. Blue highlights mark numerical results shared with the reference. GT denotes the ground-truth answer; CALR denotes the question-blind external readout's answer.

some equation expressions. This distinction explains why arithmetic-step recovery and full-chain character similarity provide complementary views of the represented content.

Arithmetic structure across problem types. Figure 6 shows how these quantities participate in different computations. Case #42 recovers $5 \times 8 = 40$ followed by $40 - 14 = 26$; Case #48 recovers the conversion to 48 inches and the subsequent division $48/6 = 8$; Case #40 uses the intermediate age 2 before obtaining 8. Across all six cases, the external readout returns the correct answer from the generated latents alone. Together with the intervention results in Table 11, these examples connect the two functional requirements: CALR exposes recoverable, problem-specific intermediate content (C2), while latent reliance is established by the answer-input interventions (C1).

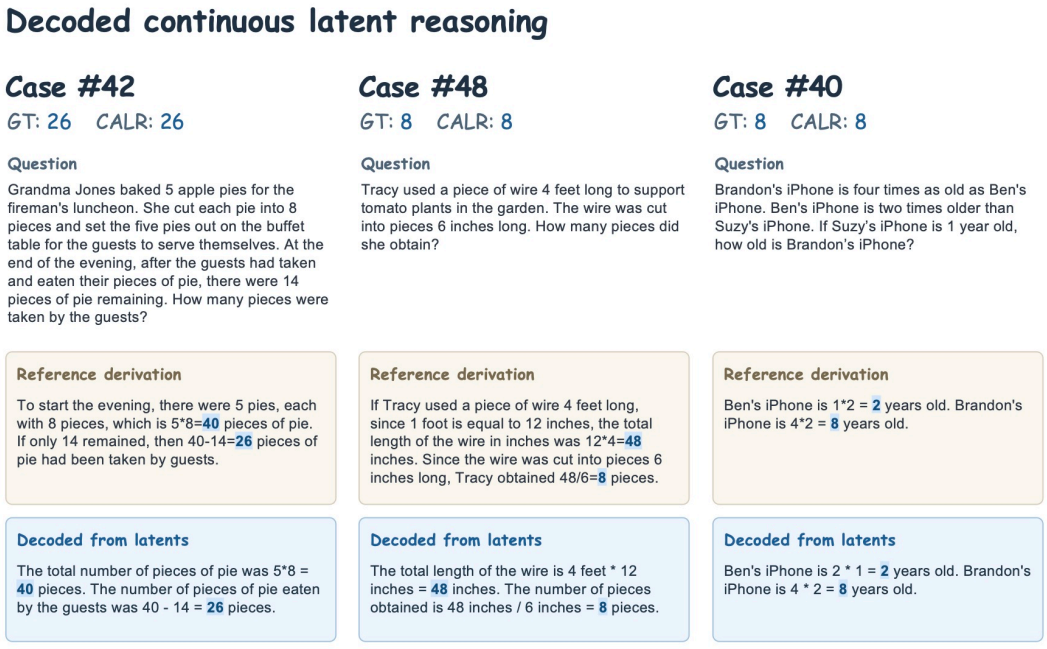

Figure 6: Decoded CALR trajectories for Cases #42, #48, and #40, covering counting, unit conversion, and multiplicative relations. Colors and answer labels follow Figure 5.