VARIATIONAL LATENT REASONING GUIDED BY RENDERED CHAIN-OF-THOUGHT

Fanmeng Wang[1,2] **Haotian Liu**[1] **Guojiang Zhao**[2] **Hongteng Xu**[1,3,4*] **Zhifeng Gao**[2*]

[1]Gaoling School of Artificial Intelligence, Renmin University of China

[2]DP Technology

[3]Beijing Key Laboratory of Research on Large Models and Intelligent Governance

[4]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE

ABSTRACT

While Chain-of-Thought (CoT) significantly enhances the performance of Large Language Models (LLMs), explicit reasoning chains introduce substantial computational redundancy. Recent latent reasoning methods attempt to mitigate this by compressing reasoning processes into the high-dimensional latent space, but suffer from severe performance degradation due to the lack of appropriate compression guidance. In this study, we propose **Re**ndered CoT-**Gu**ided variational **La**tent **R**easoning (**ReGuLaR**), a simple yet novel learning paradigm resolving this issue. Fundamentally, we formulate latent reasoning within the Variational Auto-Encoding (VAE) framework, sampling the current latent reasoning state from the posterior distribution conditioned on previous ones. Specifically, when learning this variational latent reasoning model, we render explicit reasoning chains as images, from which we extract dense visual-semantic representations to regularize the posterior distribution, thereby achieving efficient compression with minimal information loss. Extensive experiments demonstrate that ReGuLaR significantly outperforms existing latent reasoning methods across both computational efficiency and reasoning effectiveness, and even surpasses CoT through multi-modal reasoning, providing a new and insightful solution to latent reasoning. Code is available at `https://github.com/FanmengWang/ReGuLaR`.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated exceptional performance in solving complex problems, a success largely attributed to the adoption of Chain-of-Thought (CoT) techniques (Wei et al., 2022; Jin et al., 2024). By eliciting LLMs to generate intermediate reasoning steps in natural language, CoT effectively decomposes complex problems, significantly bolstering accuracy on challenging queries (Fei et al., 2023; Wang et al., 2024). However, such reasoning processes suffer from inherent inefficiency because they rely on explicit token-by-token generation, and many tokens can be redundant for improving reasoning (Kang et al., 2025). This results in prohibitive computational overhead and increased inference latency, fundamentally limiting the scalability of LLM reasoning.

In this context, recent studies have explored latent reasoning as a compelling alternative to explicit CoT (Zhu et al., 2025b). By operating directly on continuous representations, latent reasoning compresses reasoning processes into the high-dimensional latent space, thereby circumventing the overhead of decoding intermediate reasoning tokens (Zhu et al., 2025a). To instantiate this paradigm, several representative frameworks have been proposed, e.g., Coconut (Hao et al., 2025) and CoLaR (Tan et al., 2025). However, while alleviating computational burdens, existing latent reasoning methods often suffer from severe performance degradation, primarily because *the compression of reasoning processes lacks appropriate guidance.* Specifically, these methods typically rely on recursively or dynamically utilizing the hidden states of reasoning tokens to propagate logical dependencies. In the absence of discrete tokens to anchor the reasoning trajectory, this unconstrained

*Correspondence to: Hongteng Xu (hongtengxu@ruc.edu.cn) and Zhifeng Gao (gaozf@dp.tech)

recursive process becomes highly susceptible to error accumulation, leading to significant information loss and semantic drift.

In this study, we propose a novel and insightful paradigm to resolve the above challenge, learning a **Re**ndered CoT-**Gu**ided variational **La**tent **R**easoning (**ReGuLaR**) model. Fundamentally, we formulate latent reasoning as a probabilistic modeling task within the Variational Auto-Encoding (VAE) framework (Kingma & Welling, 2014). In this formulation, the latent reasoning process is achieved by sampling the current latent reasoning state from the posterior distribution conditioned on previous ones. Here, we optimize this model by maximizing its Evidence Lower Bound (ELBO) (Neal & Hinton, 1998), wherein the prior distribution of the latent reasoning state plays a critical role in regularizing the posterior distribution. As illustrated in Figure 1, we render the explicit reasoning chain as images and then leverage the visual encoder to extract visual representations with dense semantics. This rendering step is lossless, so we utilize these visual representations to regularize the posterior distribution of the latent reasoning state during training, thereby leading to our ReGuLaR with compressed but semantically meaningful latent reasoning states.

To the best of our knowledge, ReGuLaR is the first work that applies the VAE framework to understanding and modeling latent reasoning. With this framework, we demonstrate the importance of the latent reasoning state prior and propose a promising approach to designing a semantically meaningful and information-preserving prior for latent reasoning states. Extensive experiments demonstrate that ReGuLaR provides a new and insightful solution to latent reasoning. Specifically, it significantly outperforms existing latent reasoning methods, achieving state-of-the-art performance with minimal reasoning length. Furthermore, ReGuLaR natively supports multimodality within its latent reasoning processes by rendering various non-textual elements alongside text, enabling it to surpass explicit CoT in complicated reasoning scenarios.

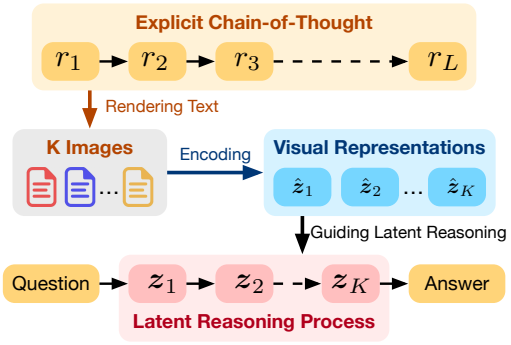


Figure 1: Illustration of our modeling principle. Given an explicit reasoning chain of length L, we render it onto K images ($K \ll L$) and extract their visual representations to guide the latent reasoning process with K steps.

2 Related Work

2.1 LLM Reasoning and Latent Reasoning

The reasoning capabilities of LLMs have been advanced by CoT techniques, which prompt the generation of intermediate reasoning steps in natural language (Wei et al., 2022; Jin et al., 2024). Building on this, subsequent research has explored various CoT constructions, including Tab-CoT (Ziqi & Lu, 2023), ToT (Yao et al., 2023), and GoT-Rationale (Besta et al., 2024). While these methods manifest reasoning in different explicit forms, verbose intermediate reasoning steps inevitably incur substantial computational cost and inference latency (Kang et al., 2025; Sui et al., 2025).

To mitigate this bottleneck, iCoT (Deng et al., 2024) internalizes intermediate reasoning steps by progressively removing them during training. Moreover, the latent reasoning paradigm has emerged, which transforms intermediate reasoning tokens into continuous representations and eliminates the overhead of language-decoding steps by executing latent reasoning processes (Zhu et al., 2025b). In particular, Coconut (Hao et al., 2025) pioneers this direction by recursively utilizing the last-layer hidden states of LLMs as the continuous latent thought, functioning as the next input embedding to drive subsequent reasoning. Meanwhile, CODI (Shen et al., 2025) further employs self-distillation to align the hidden activations of latent thoughts with explicit CoT trajectories. Most recently, CoLaR (Tan et al., 2025) achieves state-of-the-art performance by leveraging training mechanisms with variable compression factors to support flexible reasoning length. However, these methods suffer from severe performance degradation compared with explicit CoT, primarily due to the lack of appropriate compression guidance, thereby limiting their practical utility.

2.2 Empowering LLMs via Visual-Text Compression

LLMs typically rely on discrete tokenization to process text inputs, a mechanism that inevitably fragments the global semantic topology while incurring substantial computational cost (Zhao et al., 2023). To overcome these inefficiencies, the paradigm of visual-text compression (Zhao et al., 2025b) has been explored. It renders textual content as images and embeds them via the visual encoder, thereby exploiting the high information density of the visual modality. In particular, Vis-InContext (Wang et al., 2024) introduces a visualized in-context text processing framework that leverages compact visual tokens to replace long textual contexts, effectively expanding context windows without additional computational burden. Subsequently, VIST (Xing et al., 2025) proposes a fast-path compression mechanism that leverages the lightweight visual encoder to process rendered images of distant contexts for rapid skimming, significantly improving efficiency. Recently, DeepSeek-OCR (Wei et al., 2025) further brings this paradigm to the forefront by validating its feasibility and scalability on massive textual data, enabling the mapping of extensive textual contexts into ultra-compact visual tokens with high compression ratios.

While these works primarily focus on compressing input contexts, they provide strong evidence for visual representations as high-density carriers of textual information. In this context, such visual-text compression should also be useful in empowering latent reasoning, as our work verifies.

3 Proposed Method

3.1 Problem Statement and Preliminaries

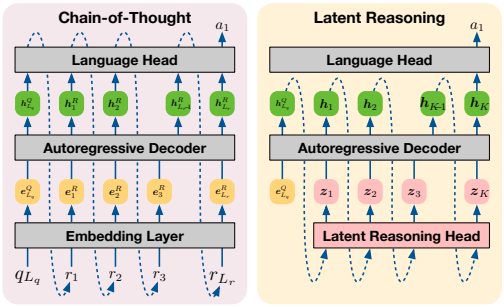
Formally, suppose that we have a reasoning dataset $\mathcal{D} = \{(\mathcal{Q}, \mathcal{R}, \mathcal{A})\}$, in which each tuple contains an input question $\mathcal{Q}$, an intermediate reasoning chain $\mathcal{R}$, and the final answer $\mathcal{A}$. Here, we represent each element in the tuple as a token sequence, i.e., $\mathcal{Q} = \{q_i\}_{i=1}^{L_q}$, $\mathcal{R} = \{r_i\}_{i=1}^{L_r}$, and $\mathcal{A} = \{a_i\}_{i=1}^{L_a}$, where L_q, L_r, and L_a are the respective sequence lengths. Additionally, for an arbitrary token sequence $\mathcal{T}$, we denote the corresponding subsequence before the i-th token as $\mathcal{T}_{<i}$.

Learning an LLM with explicit CoT. Under the Chain-of-Thought (CoT) paradigm, the LLM explicitly generates the reasoning chain token by token before producing the final answer, thereby bridging the logical gap between the question and the corresponding answer. Accordingly, given $(\mathcal{Q}, \mathcal{R}, \mathcal{A})$, we can learn the LLM via the Maximum Likelihood Estimation (MLE) as follows:

$$\max_{\theta} \underbrace{\sum_{i=1}^{L_r} \log p_\theta(r_i|\mathcal{Q}, \mathcal{R}_{<i})}_{\mathcal{L}_{\text{reasoning}}} + \underbrace{\sum_{i=1}^{L_a} \log p_\theta(a_i|\mathcal{Q}, \mathcal{R}, \mathcal{A}_{<i})}_{\mathcal{L}_{\text{answer}}}, \tag{1}$$

where θ denotes the model parameters, p_θ represents the conditional probability of each token given its history. In practice, we implement $\mathcal{L}_{\text{reasoning}}$ and $\mathcal{L}_{\text{answer}}$ as two Cross-Entropy (CE) losses.

As illustrated in Figure 2a, for the LLM using explicit reasoning, all tokens are first mapped into continuous embedding vectors (denoted as $\boldsymbol{e}^Q_{1:L_q}$, $\boldsymbol{e}^R_{1:L_r}$, and $\boldsymbol{e}^A_{1:L_a}$), and subsequently transformed by the model layers into the last hidden states (denoted as $\boldsymbol{h}^Q_{1:L_q}$, $\boldsymbol{h}^R_{1:L_r}$, and $\boldsymbol{h}^A_{1:L_a}$). In this context, the LLM necessitates the explicit token-by-token generation of the intermediate reasoning chain during inference, resulting in substantial computational overhead and inference latency.



(a) Explicit reasoning (b) Latent reasoning

Figure 2: Comparison of CoT-based explicit reasoning and latent reasoning, where the autoregressive decoder is the underlying LLM.

Learning an LLM with latent reasoning. The inefficiency of CoT further motivates the latent reasoning paradigm illustrated in Figure 2b. Specifically, latent reasoning replaces discrete reasoning tokens $\mathcal{R} = \{r_i\}_{i=1}^{L_r}$ with continuous latent reasoning states $\boldsymbol{Z} = \{\boldsymbol{z}_k\}_{k=1}^{K}$. In this context,

the LLM employs an additional latent reasoning head to derive the latent reasoning state from the current hidden state, which functions as the subsequent input embedding, thereby eliminating the overhead of decoding those intermediate reasoning tokens. Moreover, the sequence of latent reasoning states can be much shorter than the corresponding reasoning chain ($K \ll L_r$), which helps improve inference efficiency significantly.

In this context, learning such a latent reasoning model in fact corresponds to the following optimization problem, i.e.,

$$\max_{\theta'=\{\theta,\tau\}} \sum_{i=1}^{L_a} \log p_{\theta'}(a_i \mid \mathcal{Q}, \boldsymbol{Z}, \mathcal{A}_{<i}), \tag{2}$$

where the final answer is conditioned on the latent reasoning state sequence $\boldsymbol{Z}$ rather than the explicit reasoning chain $\mathcal{R}$. The parameter $\theta' = \{\theta, \tau\}$, where τ corresponds to the latent reasoning head.

Compared to equation 1, this optimization problem is inherently challenging due to the absence of ground-truth supervision for latent reasoning states. To mitigate this, Coconut (Hao et al., 2025) employs the multi-stage curriculum to progressively replace discrete reasoning steps with the last hidden state of the preceding context. However, since it relies on distilling knowledge from the original CoT, its performance is fundamentally bounded. Furthermore, CoLaR (Tan et al., 2025) directly constructs latent reasoning states by dynamically compressing the embeddings of original reasoning tokens. Nevertheless, its token grouping strategy introduces arbitrary inductive biases, and this simple aggregation inevitably leads to semantic information loss.

To overcome these problems and achieve effective latent reasoning, **we need to regularize latent reasoning states, and further impose additional information during training to ensure semantically meaningful**, which motivates the proposed ReGuLaR method.

3.2 The Variational Latent Reasoning Framework

Suppose that we would like to learn an LLM with the latent reasoning mechanism, employing the latent reasoning state sequence $\boldsymbol{Z}$ of length K to replace the explicit reasoning chain $\mathcal{R}$ of length L_r. For such an LLM, we decompose its parameters into two parts, i.e., $\theta' = \{\psi, \phi\}$, where ψ denotes the parameters of the language head that outputs discrete tokens and ϕ denotes the remaining parameters that comprise the latent reasoning head deriving $\boldsymbol{Z}$. In this context, we build a *variational latent reasoning process*: the LLM samples each latent reasoning state from its posterior distribution given the question and the previous ones, i.e.,

$$\boldsymbol{z}_k \sim p_\phi(\cdot|\mathcal{Q}, \boldsymbol{Z}_{<k}), \text{ for } k = 1, ..., K, \tag{3}$$

where $\boldsymbol{Z}_{<1} = \emptyset$. In this study, we model $p_\phi(\cdot|\mathcal{Q}, \boldsymbol{Z}_{<k})$ as a normal distribution $\mathcal{N}(\boldsymbol{\mu}_k, \text{diag}(\boldsymbol{\sigma}_k^2))$. Applying the reparametrization trick (Kingma & Welling, 2014), we leverage the latent reasoning head of the LLM to output $\boldsymbol{\mu}_k$ and $\log \boldsymbol{\sigma}_k$ based on $\mathcal{Q}$ and $\boldsymbol{Z}_{<k}$, and then sample $\boldsymbol{z}_k = \boldsymbol{\mu}_k + \boldsymbol{\sigma}_k \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and $\odot$ denotes the Hadamard product operation.

Desideratum. Ideally, $\boldsymbol{Z}$ should have the same information as the original reasoning chain $\mathcal{R}$. More specifically, for $k = 1, ..., K$, the latent reasoning state $\boldsymbol{z}_k$ should correspond to the k-th segment of $\mathcal{R}$, denoted as $\mathcal{R}_k$, where $\mathcal{R}_k \cap \mathcal{R}_{k'} = \emptyset$ for $k \neq k'$ and $\mathcal{R} = \cup_{k=1}^K \mathcal{R}_k$. Accordingly, we should be able to sample tokens in $\mathcal{R}_k$ through the distribution conditioned on $\boldsymbol{z}_k$, i.e.,

$$r \sim p_\psi(\cdot|\boldsymbol{z}_k) \text{ and } r \in \mathcal{R}_k, \text{ for } k = 1, ..., K. \tag{4}$$

Here, p_ψ is modeled using Softmax Regression, and the language head of the LLM is used to output corresponding logit values of sampled tokens.

Motivated by this, we introduce the following conditional independence assumption.

Assumption 1 (Conditional Independence)**.** *For $k = 1, ..., K$, i) the token $r \in \mathcal{R}_k$ is independent with tokens in $\{\mathcal{Q}, \mathcal{R}\} \setminus \mathcal{R}_k$ conditioned on the latent reasoning state $\boldsymbol{z}_k$, and ii) the latent reasoning state $\boldsymbol{z}_k$ is independent with tokens in $\{\mathcal{Q}, \mathcal{R}\} \setminus \mathcal{R}_k$ conditioned on the tokens in $\mathcal{R}_k$.*

Suppose that r is the i-th token in $\mathcal{R}$, which corresponds to the j-th token in $\mathcal{R}_k$. For the conditional probability used in explicit CoT, we can rewrite it based on the assumption:

$$\begin{aligned} p(r|\mathcal{Q}, \mathcal{R}_{<i}) &= \int_{\boldsymbol{z}_k} p(r|\boldsymbol{z}_k, \mathcal{Q}, \mathcal{R}_{<i}) p(\boldsymbol{z}_k|\mathcal{Q}, \mathcal{R}_{<i}) \mathrm{d}\boldsymbol{z}_k \\ &= \int_{\boldsymbol{z}_k} p_\psi(r|\boldsymbol{z}_k) p_\gamma(\boldsymbol{z}_k|\mathcal{R}_{k,<j}) \mathrm{d}\boldsymbol{z}_k. \end{aligned} \tag{5}$$

Here, $p_\psi(r|\boldsymbol{z}_k)$ is the probability of the token conditioned on $\boldsymbol{z}_k$, which is parametrized by the language head of the LLM. $p_\gamma(\boldsymbol{z}_k|\mathcal{R}_{k,<j})$ is the distribution of $\boldsymbol{z}_k$ conditioned on the partial information (i.e., before the j-th token) of the segment $\mathcal{R}_k$, whose parameters are denoted as γ.

Furthermore, when considering the posterior distribution in equation 3, we can derive the Evidence Lower Bound (ELBO) for $\log p(r|\mathcal{Q}, \mathcal{R}_{<i})$ as follows:

$$\begin{aligned}\log p(r|\mathcal{Q}, \mathcal{R}_{<i}) &\geq \mathbb{E}_{\boldsymbol{z}_k \sim p_\phi(\cdot|\mathcal{Q}, \boldsymbol{Z}_{<k})}\Big[\log \frac{p_\psi(r|\boldsymbol{z}_k) p_\gamma(\boldsymbol{z}_k|\mathcal{R}_{k,<j})}{p_\phi(\boldsymbol{z}_k|\mathcal{Q}, \boldsymbol{Z}_{<k})}\Big] \\ &= \mathbb{E}_{\boldsymbol{z}_k \sim p_\phi(\cdot|\mathcal{Q}, \boldsymbol{Z}_{<k})}[\log p_\psi(r|\boldsymbol{z}_k)] - \mathrm{KL}[\underbrace{p_\phi(\cdot|\mathcal{Q}, \boldsymbol{Z}_{<k})}_{\text{Posterior of } \boldsymbol{z}_k} \,\|\, \underbrace{p(\cdot|\mathcal{R}_{k,<j})}_{\text{Prior of } \boldsymbol{z}_k}].\end{aligned} \tag{6}$$

The ELBO of $\log p(r|\mathcal{Q}, \mathcal{R}_{<i})$ leads to a variational auto-encoding framework of latent reasoning. For the LLM, its autoregressive module with the latent reasoning head, whose parameters are ϕ, works as the encoder embedding the question and the previous latent reasoning states to the current latent reasoning state. Its language head ψ works as the decoder generating tokens based on the latent reasoning states. In equation 6, the first term corresponds to the latent reasoning loss, measuring the likelihood of reasoning tokens given the sampled latent reasoning states. The second term is the KL divergence between the posterior and prior distributions, regularizing the posterior distribution.

Considering the ELBO in equation 6 with the loss in equation 2, we can learn this variational latent reasoning model by solving the following optimization problem:

$$\begin{aligned}\max_{\theta'=\{\phi,\psi\},\gamma} &\sum\nolimits_{i=1}^{L_a} \underbrace{\mathbb{E}_{\boldsymbol{Z}\sim p_\phi(\cdot|\mathcal{Q})}[\log p_{\theta'}(a_i|\mathcal{Q}, \boldsymbol{Z}, \mathcal{A}_{<i})]}_{\mathcal{L}^{\text{Latent}}_{\text{answer}}} \\ &+ \sum\nolimits_{k=1}^{K} \sum\nolimits_{r_j \in \mathcal{R}_k} \underbrace{\mathbb{E}_{\boldsymbol{z}_k \sim p_\phi(\cdot|\mathcal{Q}, \boldsymbol{Z}_{<k})}[\log p_\psi(r_j|\boldsymbol{z}_k)]}_{\mathcal{L}^{\text{Latent}}_{\text{reasoning}}} \\ &- \sum\nolimits_{k=1}^{K} \underbrace{\mathrm{KL}[p_\phi(\cdot|\mathcal{Q}, \boldsymbol{Z}_{<k}) \| p_\gamma(\cdot|\mathcal{R}_k)]}_{\text{Regularizer of posterior}}.\end{aligned} \tag{7}$$

Here, $\boldsymbol{Z} \sim p_\phi(\cdot|\mathcal{Q})$ is implemented by sequential sampling shown in equation 3. $p_\gamma(\cdot|\mathcal{R}_k)$ denotes the distribution of $\boldsymbol{z}_k$ conditioned on $\mathcal{R}_k$, whose parameters are denoted as γ. Similar to (Tomczak & Welling, 2018; Xu et al., 2020), we learn the prior distribution of the latent reasoning state p_γ together with the latent reasoning model. Obviously, this learning problem is analogous to that of explicit CoT in equation 1, which maximizes the likelihoods of both reasoning and answer tokens. Furthermore, the posterior distribution p_ϕ is optimized under the guidance of p_γ.

Remark. In equation 7, we replace the $p(\cdot|\mathcal{R}_{k,<j})$ in equation 6 with $p_\gamma(\cdot|\mathcal{R}_k)$. Such a modification leads to a "stable" prior distribution invariant with the selection of the reasoning token $r_j \in \mathcal{R}_k$, which is reasonable in practice. Firstly, as aforementioned, an ideal latent reasoning state $\boldsymbol{z}_k$ should cover the information of $\mathcal{R}_k$, so that modeling its distribution conditioned on $\mathcal{R}_k$ rather than $\mathcal{R}_{k,<j}$ can impose more information to the model, leading to better regularization. Secondly, if the distribution p_γ changes with respect to the selection of the reasoning tokens, we would have to recompute the KL divergence in equation 7 for each $r_j \in \mathcal{R}_k$, which would cause significant training overhead. Therefore, applying $p_\gamma(\cdot|\mathcal{R}_k)$ is reasonable and efficient in practice.

3.3 Implementing the Framework via ReGuLaR

As analyzed in Section 3.1, the crux of learning latent reasoning models lies in guiding the posterior of latent reasoning states with less information loss. Therefore, designing and learning the semantically meaningful prior distribution p_γ is critical for our variational latent reasoning model.

Inspired by recent advances (Xing et al., 2025; Wei et al., 2025) establishing visual representations as compact carriers of textual information, we propose ReGuLaR, a rendered CoT-guided variational latent reasoning method, to implement the VAE framework in equation 7. As illustrated in Figure 3, ReGuLaR parametrizes the prior distribution of latent reasoning states based on the visual

representation of rendered CoT. Formally, we can render K segments of the reasoning chain $\mathcal{R}$ into K images and extract their visual representation as follows: For $k = 1, ..., K$,

$$1)\ \underbrace{\mathcal{I}_k = f(\mathcal{R}_k)}_{\text{Rendering}},\ \ 2)\ \underbrace{\boldsymbol{v}_k = v(\mathcal{I}_k)}_{\text{Embedding}},\ \ 3)\ \underbrace{\hat{\boldsymbol{z}}_k = g_\gamma(\boldsymbol{v}_k)}_{\text{Adaptation}}. \tag{8}$$

Here, f is the predefined rendering function, which maps an arbitrary token sequence to an image with a size $H \times W$. v is the pretrained visual encoder, which transforms pixel-wise images into visual representations. We employ the trainable adapter $g_\gamma : \mathbb{R}^{d_v} \mapsto \mathbb{R}^{d_h}$ to map visual representations $\boldsymbol{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_K]$ to the proposed latent reasoning space.

In this study, we directly adopt the optimal rendering configuration identified in Glyph (Cheng et al., 2025) for f, which maximizes the semantic density. Meanwhile, we implement v as the pretrained visual encoder in DeepSeek-OCR (Wei et al., 2025) since it has been architecturally optimized for visual-text compression, enabling it to encode high-resolution and text-dense inputs into compact representations with minimal semantic loss. Additionally, the adapter g_γ is instantiated as a multi-layer perception (MLP) with parameters γ. Notably, as f and v are frozen in our work, **we can pre-compute these visual representations offline before training**, significantly reducing computational overhead.

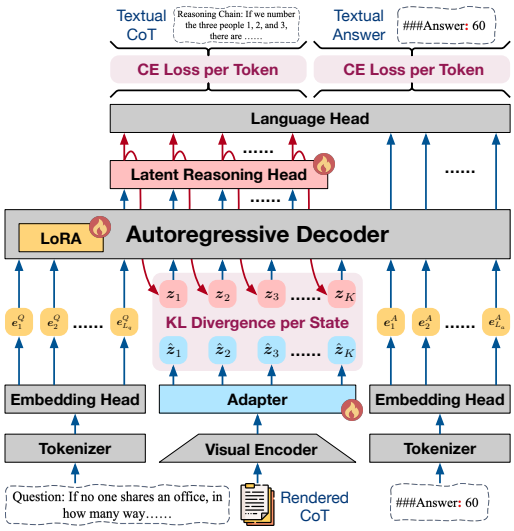


Figure 3: Illustration of the proposed ReGuLaR. Here, only the latent reasoning head, adapter, and LoRA module are trainable. The blue arrows "→" indicate deterministic outputs, while the red arrows "→" indicate probabilistic outputs achieved by sampling. The special token "###" triggers the transition from the reasoning process to answer generation during inference.

As a result, given the reasoning chain $\mathcal{R}$, we first segment it into K parts[1] and then model $p_\gamma(\cdot|\mathcal{R}_k)$ as a normal distribution $\mathcal{N}(\hat{\boldsymbol{z}}_k, \boldsymbol{I})$ for $k = 1, ..., K$, whose mean is determined by equation 8 and variance is fixed as an identity matrix.

Accordingly, for $k = 1, ..., K$, the KL divergence (i.e., $\mathrm{KL}[p_\phi \| p_\gamma]$) in equation 7 becomes

$$\frac{\|\boldsymbol{\mu}_k - \hat{\boldsymbol{z}}_k\|_2^2 + \|\boldsymbol{\sigma}_k\|_2^2}{2} - \log|\mathrm{diag}(\boldsymbol{\sigma}_k)|. \tag{9}$$

Following (Tan et al., 2025), we approximate the KL divergence during training as follows:

$$\frac{1}{2}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\|\boldsymbol{\mu}_k + \boldsymbol{\sigma}_k \odot \boldsymbol{\epsilon} - \hat{\boldsymbol{z}}_k\|_2^2] - \log|\mathrm{diag}(\boldsymbol{\sigma}_k)|. \tag{10}$$

Notably, the LLM trained by ReGuLaR still follows the standard latent reasoning workflow initiated solely by the textual input question: the model utilizes the trained latent reasoning head to generate latent reasoning states until the special termination token is encountered, which acts as a signal triggering the language head to decode the final answer. Algorithms 1 and 2 in Appendix A.3 have presented training and inference schemes of ReGuLaR in detail.

3.4 Advantages over Existing Methods

Unlike the token grouping strategy used in CoLaR (Tan et al., 2025), ReGuLaR renders explicit reasoning chains into images to preserve semantic integrity. The visual representations of these rendered images provide better regularization than the aggregation of grouped token embeddings, resulting in less information loss. As previously noted, the inference phase remains consistent with standard latent reasoning, accepting pure text inputs and imposing no extra computational cost.

[1]We regarding each sentence in $\mathcal{R}$ as a segment in Table 1 and further analyze the impact of compression rate (i.e., $|\mathcal{R}|/K$) in Figure 4b.

Table 1: Performance comparison on four math reasoning datasets using LLaMA-3.2-1B-Instruct as the LLM backbone, where we report the averaged number and 95% confidence interval ($\pm$) on Accuracy (Acc. %) and Reasoning Length (# L) via five independent evaluations.

Method	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average	
	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L
iCoT	$19.8_{\pm 0.23}$	$0.00_{\pm 0.00}$	$3.87_{\pm 0.16}$	$0.00_{\pm 0.00}$	$36.4_{\pm 0.51}$	$0.00_{\pm 0.00}$	$38.2_{\pm 0.66}$	$0.00_{\pm 0.00}$	24.6	0.00
CODI	$13.3_{\pm 0.62}$	$6.00_{\pm 0.00}$	$2.97_{\pm 0.24}$	$6.00_{\pm 0.00}$	$21.7_{\pm 0.73}$	$6.00_{\pm 0.00}$	$19.2_{\pm 0.83}$	$6.00_{\pm 0.00}$	14.3	6.00
Coconut	$20.5_{\pm 0.68}$	$6.00_{\pm 0.00}$	$4.86_{\pm 0.30}$	$6.00_{\pm 0.00}$	$39.8_{\pm 0.71}$	$6.00_{\pm 0.00}$	$41.4_{\pm 0.69}$	$6.00_{\pm 0.00}$	26.6	6.00
CoLaR*	$26.6_{\pm 0.18}$	$5.63_{\pm 0.01}$	$6.23_{\pm 0.14}$	$7.01_{\pm 0.05}$	$47.1_{\pm 0.30}$	$2.96_{\pm 0.02}$	$87.0_{\pm 0.21}$	$3.23_{\pm 0.01}$	41.7	4.70
ReGuLaR	$\mathbf{34.9}_{\pm 0.26}$	$\mathbf{3.69}_{\pm 0.21}$	$\mathbf{8.27}_{\pm 0.14}$	$\mathbf{4.12}_{\pm 0.48}$	$\mathbf{50.1}_{\pm 0.39}$	$\mathbf{2.02}_{\pm 0.18}$	$\mathbf{89.2}_{\pm 0.27}$	$\mathbf{2.28}_{\pm 0.27}$	**45.6**	**3.03**

*Results here use the maximum compression rate of 5, while comparison across varying rates is in Figure 4b

Moreover, non-textual elements (e.g., charts, graphs, and diagrams) can also be rendered and encoded alongside text. Therefore, in addition to mitigating textual information loss, ReGuLaR natively supports the use of multi-modal information in its latent reasoning processes. In complex tasks involving multi-modal reasoning information, this advantage enables ReGuLaR not only to outperform existing latent reasoning methods but also to surpass explicit textual CoT.

4 Experiments

4.1 Experimental Setup

Datasets. In line with the previous work (Tan et al., 2025), we primarily train and evaluate models on GSM8K-Aug (Deng et al., 2023), and additionally evaluate trained models using three out-of-domain math reasoning datasets: GSM-Hard (Gao et al., 2023), SVAMP (Patel et al., 2021), and MultiArith (Roy & Roth, 2015). Meanwhile, we also train and evaluate on GSM8K-Aug-NL, the variant of GSM8K-Aug that preserves natural language explanations, to explore their extreme compression capabilities. Furthermore, we conduct experiments on AQUA-RAT (Ling et al., 2017) and MATH (Hendrycks et al., 2021) to verify their performance in more challenging problems.

Baselines. We employ various latent reasoning methods as baselines, including iCoT (Deng et al., 2024), CODI (Shen et al., 2025), Coconut (Hao et al., 2025), and CoLaR (Tan et al., 2025). Specifically, all these baselines are implemented following their default configurations within the unified framework provided by the state-of-the-art method CoLaR, ensuring fairness and consistency.

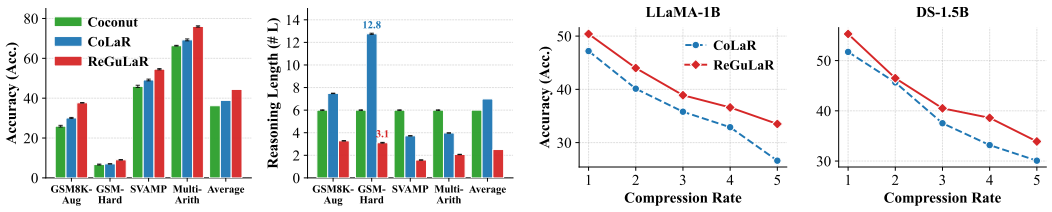
Evaluation. We adopt evaluation metrics widely used in this domain to assess performance, including i) Accuracy (Acc.), which evaluates reasoning effectiveness by calculating the percentage of correctly predicted answers; and ii) Reasoning Length (# L), which evaluates reasoning efficiency by calculating the number of reasoning steps per question. Specifically, all these evaluations are repeated over five independent runs with different random seeds, ensuring statistical reliability.

Implementation. We employ LLaMA-3.2-1B-Instruct as the LLM backbone unless otherwise specified. Specifically, following established baselines, we keep the LLM backbone frozen and exclusively optimize LoRA (Hu et al., 2022) modules, which are configured with $r = 128$ and $\alpha = 32$.

More details about datasets, baselines, and implementation have been provided in Appendix A.

4.2 Main Results

Performance Comparison. Table 1 presents the results of various methods on four math reasoning datasets, where ReGuLaR achieves state-of-the-art performance. Specifically, compared with the strongest baseline CoLaR, ReGuLaR consistently delivers substantial accuracy gains across all datasets while simultaneously reducing the average reasoning length by approximately 35% (from 4.70 to 3.03). The results suggest that ReGuLaR successfully compresses the reasoning chain into a more compact and informative latent reasoning state sequence, underscoring both its computational efficiency and reasoning effectiveness.



(a) Generalizability Analysis (b) Compression Analysis

Figure 4: (a) Generalizability analysis using DeepSeek-R1-Distill-Qwen-1.5B as the LLM backbone. (b) Compression Analysis on the GSM8K-Aug dataset using LLaMA-3.2-1B-Instruct (left) and DeepSeek-R1-Distill-Qwen-1.5B (right) as the LLM backbone, where the compression rate represents the number of explicit reasoning tokens corresponding to a single latent reasoning state.

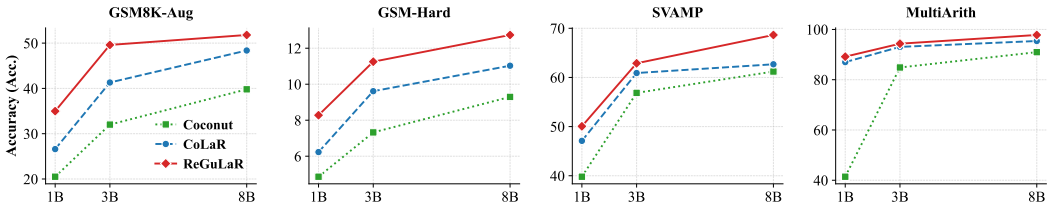


Figure 5: Scalability analysis across varying model sizes, where we employ LLaMA-3.2 (1B, 3B) and LLaMA-3.1 (8B) Instruct variants as the LLM backbones. Comprehensive results, including reasoning length, are provided in Figure 6.

Generalizability Analysis. To verify the generalizability of ReGuLaR to different LLM backbones, we replace the underlying model (i.e., LLaMA-3.2-1B-Instruct) with DeepSeek-R1-Distill-Qwen-1.5B. As shown in Figure 4a, ReGuLaR consistently maintains its superiority across all datasets, achieving the highest accuracy with the shortest reasoning length. Especially on the challenging GSM-Hard dataset, while the strongest baseline CoLaR requires an average of 12.8 reasoning steps, ReGuLaR achieves higher accuracy with only 3.1 steps.

Compression Analysis. While the strongest baseline CoLaR employs token embedding compression and ReGuLaR leverages visual-text compression, both map a sequence of explicit reasoning tokens to a single latent reasoning state. Given this commonality, we assess their performance under identical compression rates (i.e., the number of tokens condensed into one latent reasoning state). As shown in Figure 4b, although accuracy naturally decreases with higher compression rates, ReGuLaR consistently outperforms CoLaR across all settings on both LLM backbones, verifying its advantage in preserving semantic information.

Scalability Analysis. To evaluate the scaling potential of ReGuLaR, we conduct experiments across varying model sizes within the LLaMA-3 family, ranging from 1B to 8B (i.e., instruct variants of LLaMA-3.2 1B/3B and LLaMA-3.1 8B). As shown in Figure 5, ReGuLaR demonstrates strong positive scaling behavior, consistently maintaining a significant performance margin over the top-performing baselines (CoLaR and Coconut) across all model scales and datasets, demonstrating its seamless scalability and potential for broader application in large-scale foundation models.

More experimental results, including ablation studies, have been provided in Appendix B.

4.3 Extreme Compression with ReGuLaR

As expressed in equation 8, the reasoning chain is decomposed into K segments, which are subsequently rendered into images to yield K corresponding latent reasoning states during training. In Table 1, we follow the natural linguistic partition, treating each sentence as a segment. In this section, we further investigate the limit of model performance by introducing an extreme compression setting. Specifically, we directly render the entire reasoning chain into a single image, compressing all reasoning information into one latent reasoning state (i.e., $K = 1$). We conduct this experiment

Table 2: Extreme compression performance of ReGuLaR, where CoLaR is implemented following its default configuration as a reference.

Dataset	Method	LLaMA-1B		LLaMA-3B		LLaMA-8B		DS-1.5B	
		Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L
GSM8K-Aug-NL	CoLaR	$18.7_{\pm 0.34}$	$13.1_{\pm 0.04}$	$31.0_{\pm 0.26}$	$14.8_{\pm 0.11}$	$31.7_{\pm 0.21}$	$13.1_{\pm 0.07}$	$16.4_{\pm 0.23}$	$15.4_{\pm 0.03}$
	ReGuLaR	$\mathbf{20.2}_{\pm 0.71}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{32.6}_{\pm 0.43}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{38.6}_{\pm 0.33}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{31.3}_{\pm 0.09}$	$\mathbf{1.00}_{\pm 0.00}$
AQUA-RAT	CoLaR	$24.2_{\pm 0.37}$	$19.4_{\pm 0.31}$	$31.7_{\pm 1.27}$	$28.5_{\pm 0.66}$	$35.8_{\pm 0.90}$	$20.5_{\pm 1.66}$	$33.2_{\pm 1.04}$	$26.7_{\pm 0.54}$
	ReGuLaR	$\mathbf{37.1}_{\pm 0.61}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{39.7}_{\pm 0.50}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{41.2}_{\pm 0.34}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{41.0}_{\pm 0.48}$	$\mathbf{1.00}_{\pm 0.00}$
MATH	CoLaR	$3.65_{\pm 0.13}$	$58.1_{\pm 0.29}$	$8.03_{\pm 0.25}$	$60.4_{\pm 0.19}$	$9.06_{\pm 0.20}$	$67.7_{\pm 0.59}$	$10.3_{\pm 0.22}$	$62.6_{\pm 0.21}$
	ReGuLaR	$\mathbf{6.62}_{\pm 0.18}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{11.8}_{\pm 0.03}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{13.9}_{\pm 0.10}$	$\mathbf{1.00}_{\pm 0.00}$	$\mathbf{15.6}_{\pm 0.14}$	$\mathbf{1.00}_{\pm 0.00}$

Table 3: Performance comparison on the molecular captioning task. Note that "w/o 2D" denotes the variant of ReGuLaR trained with the original textual reasoning chains for ablation.

Method		BLEU-2↑	BLEU-4↑	METEOR↑	ROUGE-1↑	ROUGE-2↑	ROUGE-L↑	# L
LLaMA-1B	CoT	$0.2828_{\pm 0.002}$	$0.1804_{\pm 0.002}$	$0.3778_{\pm 0.001}$	$0.4632_{\pm 0.002}$	$0.2675_{\pm 0.003}$	$0.3914_{\pm 0.001}$	$314.825_{\pm 4.143}$
	CoLaR	$0.0995_{\pm 0.002}$	$0.0400_{\pm 0.001}$	$0.1774_{\pm 0.002}$	$0.2306_{\pm 0.004}$	$0.0911_{\pm 0.002}$	$0.1900_{\pm 0.003}$	$212.347_{\pm 0.660}$
	ReGuLaR	$\mathbf{0.3673}_{\pm 0.002}$	$\mathbf{0.2692}_{\pm 0.002}$	$\mathbf{0.4593}_{\pm 0.002}$	$\mathbf{0.5319}_{\pm 0.002}$	$\mathbf{0.3502}_{\pm 0.002}$	$\mathbf{0.4635}_{\pm 0.002}$	$\mathbf{1.000}_{\pm 0.000}$
	w/o 2D	$0.2872_{\pm 0.002}$	$0.1845_{\pm 0.002}$	$0.3777_{\pm 0.001}$	$0.4678_{\pm 0.001}$	$0.2716_{\pm 0.001}$	$0.3962_{\pm 0.000}$	$1.000_{\pm 0.000}$
LLaMA-3B	CoT	$0.3167_{\pm 0.003}$	$0.2146_{\pm 0.003}$	$0.4117_{\pm 0.003}$	$0.4948_{\pm 0.001}$	$0.3011_{\pm 0.003}$	$0.4230_{\pm 0.002}$	$316.829_{\pm 3.502}$
	CoLaR	$0.1734_{\pm 0.002}$	$0.0861_{\pm 0.002}$	$0.2487_{\pm 0.003}$	$0.3381_{\pm 0.002}$	$0.1576_{\pm 0.001}$	$0.2840_{\pm 0.002}$	$200.267_{\pm 5.437}$
	ReGuLaR	$\mathbf{0.4329}_{\pm 0.002}$	$\mathbf{0.3394}_{\pm 0.002}$	$\mathbf{0.5158}_{\pm 0.002}$	$\mathbf{0.5860}_{\pm 0.002}$	$\mathbf{0.4164}_{\pm 0.003}$	$\mathbf{0.5208}_{\pm 0.002}$	$\mathbf{1.000}_{\pm 0.000}$
	w/o 2D	$0.3182_{\pm 0.003}$	$0.2119_{\pm 0.003}$	$0.4172_{\pm 0.002}$	$0.4940_{\pm 0.002}$	$0.2961_{\pm 0.003}$	$0.4184_{\pm 0.002}$	$1.000_{\pm 0.000}$
LLaMA-8B	CoT	$0.3410_{\pm 0.001}$	$0.2378_{\pm 0.001}$	$0.4377_{\pm 0.001}$	$0.5138_{\pm 0.001}$	$0.3212_{\pm 0.001}$	$0.4403_{\pm 0.001}$	$295.736_{\pm 4.105}$
	CoLaR	$0.2031_{\pm 0.003}$	$0.1385_{\pm 0.002}$	$0.2803_{\pm 0.003}$	$0.3883_{\pm 0.001}$	$0.2340_{\pm 0.002}$	$0.3449_{\pm 0.002}$	$163.655_{\pm 3.981}$
	ReGuLaR	$\mathbf{0.4610}_{\pm 0.001}$	$\mathbf{0.3691}_{\pm 0.001}$	$\mathbf{0.5479}_{\pm 0.002}$	$\mathbf{0.6094}_{\pm 0.001}$	$\mathbf{0.4428}_{\pm 0.001}$	$\mathbf{0.5439}_{\pm 0.001}$	$\mathbf{1.000}_{\pm 0.000}$
	w/o 2D	$0.3403_{\pm 0.002}$	$0.2364_{\pm 0.001}$	$0.4394_{\pm 0.002}$	$0.5132_{\pm 0.002}$	$0.3187_{\pm 0.002}$	$0.4367_{\pm 0.003}$	$1.000_{\pm 0.000}$
DS-1.5B	CoT	$0.2682_{\pm 0.002}$	$0.1659_{\pm 0.002}$	$0.3637_{\pm 0.002}$	$0.4469_{\pm 0.001}$	$0.2523_{\pm 0.001}$	$0.3774_{\pm 0.001}$	$344.067_{\pm 6.366}$
	CoLaR	$0.0968_{\pm 0.002}$	$0.0516_{\pm 0.001}$	$0.1644_{\pm 0.002}$	$0.2149_{\pm 0.004}$	$0.0916_{\pm 0.001}$	$0.1801_{\pm 0.003}$	$580.558_{\pm 9.779}$
	ReGuLaR	$\mathbf{0.3536}_{\pm 0.002}$	$\mathbf{0.2545}_{\pm 0.001}$	$\mathbf{0.4397}_{\pm 0.001}$	$\mathbf{0.5187}_{\pm 0.002}$	$\mathbf{0.3347}_{\pm 0.001}$	$\mathbf{0.4514}_{\pm 0.001}$	$\mathbf{1.000}_{\pm 0.000}$
	w/o 2D	$0.2672_{\pm 0.002}$	$0.1640_{\pm 0.002}$	$0.3764_{\pm 0.002}$	$0.4645_{\pm 0.002}$	$0.2499_{\pm 0.002}$	$0.3476_{\pm 0.003}$	$1.000_{\pm 0.000}$

on the GSM8K-Aug-NL dataset, which preserves natural language explanations within the reasoning process, inherently yielding relatively long reasoning chains. The AQUA-RAT and MATH datasets are also incorporated to verify the performance on more challenging problems.

Table 2 presents the performance of ReGuLaR under the extreme compression setting. Specifically, despite being constrained to a single latent reasoning step, ReGuLaR still outperforms the strongest baseline CoLaR across all model scales and datasets. This advantage is particularly evident on the MATH dataset, underscoring its superior compression capability in complex reasoning scenarios.

5 Latent Reasoning Beyond Textual Domain

As discussed in Section 3.4, non-textual elements can also be rendered alongside text, making ReGuLaR support multi-modality within latent reasoning while maintaining the standard textual I/O interface. Here, we conduct experiments to investigate the efficacy of this extended capability.

Dataset. Unlike existing multi-modal datasets that rely on image-based inputs and textual reasoning chains, we require datasets that maintain purely textual I/O while integrating multi-modal reasoning chains as intermediate bridges. To this end, we employ the **molecule captioning benchmark** from MolReasoner (Zhao et al., 2025a), which requires the LLM to generate natural language descriptions of the given molecules. In particular, while the original dataset only provides textual reasoning chains, we utilize RDKit (Landrum et al., 2013) to generate the corresponding 2D molecular graphs and combine them with the original reasoning chains, constructing multi-modal reasoning chains.

Baselines and Evaluation. Following Section 4.3, we also render the entire multi-modal reasoning chain into a single image to train ReGuLaR. Baseline methods include CoT and CoLaR, both of which apply the original textual reasoning chains for training due to their inherent lack of multi-

modal support. Performance is evaluated by **BLEU**, **METEOR**, and **ROUGE** metrics, which quantify the n-gram overlap and semantic similarity between the generated and reference captions.

Results. As presented in Table 3, ReGuLaR achieves state-of-the-art performance across all metrics and backbones. Specifically, despite being constrained to a single latent reasoning step, ReGuLaR significantly outperforms not only the strongest latent reasoning baseline CoLaR, but also the explicit CoT method, both of which apply hundreds of reasoning steps. Notably, although CoT and CoLaR are trained using the original textual reasoning chains due to their inherent lack of multi-modal support, the comparison between our method and these two methods is fair to some extent because the 2D graphs in the multi-modal reasoning chains only convert the data format, which do not provide any additional information. To ensure a strictly fair comparison, we construct an ablation setting (denoted as "w/o 2D") where only textual reasoning chains are rendered. Even in this setting, ReGuLaR maintains performance comparable to CoT while drastically reducing the reasoning steps. These results further validate the extreme compression capability of ReGuLaR and highlight its unique advantage of unifying textual and non-textual elements for comprehensive reasoning.

6 Conclusion

In this paper, we propose a new and insightful latent reasoning paradigm that models latent reasoning within the VAE framework and learns it guided by rendered CoT. Our method significantly outperforms existing latent reasoning methods across both computational efficiency and reasoning ability, and even surpasses explicit CoT through supporting multi-modal latent reasoning.

Future work. Currently, standard benchmarks like GSM8K and GSM8K-Aug may limit the assessment of advanced reasoning capabilities because they have limited data sizes and overly simple reasoning chains. We plan to address this gap by developing a large-scale and high-quality reasoning dataset to evaluate latent reasoning methods in more demanding settings. In addition, we will further explore latent reasoning and study whether and how it can outperform explicit CoT in theory.

Acknowledgements

This work was supported in part by the Beijing Major Science and Technology Project under Contract No. Z251100008425002F and Beijing Municipal Science & Technology Commission, Administrative Commission of Zhongguancun Science Park under Contract No. Z251100007525009. It was also supported by the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China, and the Public Computing Cloud, Renmin University of China. We acknowledge the support provided by the fund for building world-class universities (disciplines) of Renmin University of China and by the funds from Beijing Key Laboratory of Research on Large Models and Intelligent Governance, Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education, and from Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China.

References

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 17682–17690, 2024.

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

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.

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.

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.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. Reasoning implicit sentiment with chain-of-thought prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1171–1182, 2023.

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*, pp. 10764–10799, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason E Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *Second Conference on Language Modeling*, 2025.

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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Mingyu Jin, Qinkai Yu, Shu Dong, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, and Mengnan Du. The impact of reasoning step length on large language models. In *Findings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024*, pp. 1830–1842, 2024.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8(31.10):5281, 2013.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Radford M Neal and Geoffrey E Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer, 1998.

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, 2021.

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, 2015.

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

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jakub Tomczak and Max Welling. Vae with a vampprior. In *International conference on artificial intelligence and statistics*, pp. 1214–1223. PMLR, 2018.

Alex Jinpeng Wang, Linjie Li, Yiqi Lin, Min Li, Lijuan Wang, and Mike Zheng Shou. Leveraging visual tokens for extended text contexts in multi-modal learning. *Advances in Neural Information Processing Systems*, pp. 14325–14348, 2024.

Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025.

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

Ling Xing, Alex Jinpeng Wang, Rui Yan, Xiangbo Shu, and Jinhui Tang. Vision-centric token compression in large language model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Hongteng Xu, Dixin Luo, Ricardo Henao, Svati Shah, and Lawrence Carin. Learning autoencoders with relational regularization. In *International Conference on Machine Learning*, pp. 10576–10586. PMLR, 2020.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36:11809–11822, 2023.

Guojiang Zhao, Sihang Li, Zixiang Lu, Zheng Cheng, Haitao Lin, Lirong Wu, Hanchen Xia, Hengxing Cai, Wentao Guo, Hongshuai Wang, et al. Molreasoner: Toward effective and interpretable reasoning for molecular llms. *arXiv preprint arXiv:2508.02066*, 2025a.

Hongbo Zhao, Meng Wang, Fei Zhu, Wenzhuo Liu, Bolin Ni, Fanhu Zeng, Gaofeng Meng, and Zhaoxiang Zhang. Vtcbench: Can vision-language models understand long context with vision-text compression? *arXiv preprint arXiv:2512.15649*, 2025b.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Rui-Jie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen Xue, Xuanliang Zhang, Yong Shan, et al. A survey on latent reasoning. *arXiv preprint arXiv:2507.06203*, 2025b.

Jin Ziqi and Wei Lu. Tab-cot: Zero-shot tabular chain of thought. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 10259–10277, 2023.

A More Experimental Details

A.1 Datasets

Following previous work (Tan et al., 2025), we primarily train and evaluate our method on the GSM8K-Aug dataset (Deng et al., 2023), and additionally evaluate trained models using three out-of-domain math reasoning datasets: GSM-Hard (Gao et al., 2023), SVAMP (Patel et al., 2021), and MultiArith (Roy & Roth, 2015). Meanwhile, we also train and evaluate on the GSM8K-Aug-NL dataset, a variant of the GSM8K-Aug dataset that preserves natural language explanations within reasoning processes, to demonstrate the extreme compression capability. In addition, we extend our experiments to the AQUA-RAT dataset (Ling et al., 2017) and MATH dataset (Hendrycks et al., 2021) to verify the performance on more challenging problems.

Here, the detailed description of the above datasets is provided below:

- **GSM8K-Aug** (Deng et al., 2023): This dataset is an augmented version of the GSM8K dataset (Cobbe et al., 2021), constructed by prompting GPT-4 to extend the original training set to 385k samples. In particular, it eliminates natural language descriptions within reasoning processes, formalizing reasoning steps as mathematical expressions only.
- **GSM8K-Aug-NL** (Deng et al., 2023): Similar to GSM8K-Aug, this dataset is also constructed by prompting GPT-4 to extend the original training set of the GSM8K dataset to 385k samples. The distinguishing feature is that GSM8K-Aug-NL preserves natural language explanations within reasoning processes, formalizing reasoning steps as natural language sentences. Consequently, compared to GSM8K-Aug, this dataset exhibits longer reasoning chains, with a reasoning style that more closely resembles verbose CoTs.
- **GSM8K-Hard** (Gao et al., 2023): This dataset is the harder version of the GSM8K dataset, constructed by replacing numbers in the original test set with larger numbers that are less common. In particular, this dataset serves as the out-of-domain dataset in our experiments.
- **SVAMP** (Patel et al., 2021): This dataset comprises 1,000 math word problems, designed to evaluate the robustness of models in solving fundamental mathematical problems. In particular, this dataset serves as the out-of-domain dataset in our experiments.
- **MultiArith** (Roy & Roth, 2015): This dataset comprises 600 multi-step arithmetic problems, designed to evaluate the capability of models in handling tasks that require multi-step reasoning. In particular, this dataset serves as the out-of-domain dataset in our experiments.
- **AQUA-RAT** (Ling et al., 2017): This dataset comprises about 100,000 algebraic word problems, where each problem is equipped with a step-by-step natural language explanation that details the logical derivation leading to the answer.
- **MATH** (Hendrycks et al., 2021): This dataset comprises 12,500 highly challenging mathematics competition problems, where each problem is equipped with a comprehensive step-by-step solution. Due to its high difficulty and detailed explanations, it stands as one of the mainstream datasets for evaluating mathematical reasoning capabilities.

A.2 Baselines

We employ various respective latent-based methods as baselines, including iCoT (Deng et al., 2024), CODI (Shen et al., 2025), Coconut (Hao et al., 2025), and CoLaR (Tan et al., 2025).

Here, the detailed description of the above baselines is provided below:

- **iCoT** (Deng et al., 2024): This baseline gradually removes reasoning steps during finetuning, thereby internalizing the reasoning process while maintaining high performance.
- **CODI** (Shen et al., 2025): This baseline directly employs self-distillation to align the hidden activations of latent thoughts with explicit reasoning trajectories, thereby transferring reasoning capabilities into latent space.
- **Coconut** (Hao et al., 2025): This baseline recursively utilizes the last hidden state of LLMs as latent thought, thereby functioning as the next input to drive subsequent reasoning.

- **CoLaR** (Tan et al., 2025): This baseline dynamically compresses embeddings of consecutive reasoning tokens and autoregressively predicts the subsequent compressed embeddings, thereby supporting flexible reasoning lengths.

A.3 IMPLEMENTATION

Rendering Configuration. As expressed in equation 8, the rendering function Φ is parameterized by a specific configuration vector θ^* to map an arbitrary token sequence to an image. To ensure consistent visual encoding and maximize the semantic density, we directly adopt the optimal rendering configuration identified in Glyph (Cheng et al., 2025). Specifically, we utilize the `Verdana` font family and set a tight layout (i.e., 9pt font size with 10pt line height) to compact the logical topology. The detailed rendering specifications are summarized in Table 4.

Table 4: Detailed rendering configuration used in our experiments.

Parameter	Value	Description
Canvas & Layout		
Page Size	595×842	Standard A4 dimension (points)
DPI	72	Screen resolution density
Margins (X, Y)	10, 10	Minimal padding to maximize content area
Background Color	#FFFFFF	White background
Auto Crop	True	Crop to content bounding box
Typography		
Font Family	Verdana	Sans-serif font for optical clarity
Font Size	9	Compact size for high density
Line Height	10	Tight vertical spacing
Font Color	#000000	Black text for high contrast
Alignment	LEFT	Standard left-aligned text
Spacing & Indentation		
Indent (First/Left/Right)	0	No indentation
Spacing (Before/After)	0	No paragraph spacing
Border Width	0	Borderless rendering

Visual Encoder. As expressed in equation 8, rendered images obtained from the preceding stage are mapped into continuous space, transforming pixel-based inputs into visual-semantic vectors. To encode high-resolution and text-dense inputs into compact representations with minimal semantic loss, we adopt the trained visual encoder from DeepSeek-OCR (Wei et al., 2025). Specifically, this visual encoder integrates a SAM-Base backbone (80M) for fine-grained local perception (via window attention) and a CLIP-Large backbone (300M) for high-level semantic extraction (via global attention), bridged by a $16\times$ convolutional compressor. Besides, it has been trained via the generative objective on the massive corpus of diverse optical data (including multilingual documents and synthesized charts/formulas), making it align perfectly with our requirements. In our experiments, we keep this visual encoder frozen and utilize the standard Tiny mode (resolution 512×512). This configuration initially maps each rendered image into 64 visual tokens, which are subsequently aggregated via mean pooling to derive a single and compact visual-semantic representation.

Hyperparameters. We primarily leverage LLaMA-3.2-1B-Instruct (Grattafiori et al., 2024) as the LLM backbone unless otherwise specified. Specifically, we keep the LLM backbone frozen and exclusively optimize LoRA (Hu et al., 2022) modules, which are configured with $r = 128$ and $\alpha = 32$ following established baselines. In addition, both the adapter and the latent reasoning head are instantiated as Multi-Layer Perceptrons (MLPs), where the adapter maps the visual encoder's output dimension (d_v=1280) to the LLM's hidden dimension (d_h=2048) and the latent reasoning head directly operates within the LLM's hidden dimension (d_h=2048). For training, we optimize the model using the AdamW optimizer (Loshchilov & Hutter, 2019) with a weight decay of 0.01 and a learning rate of 1e-4, employing a constant schedule with a 1,000-step warmup phase. Specifically,

we utilize Distributed Data Parallel across eight NVIDIA A100 GPUs to ensure training stability and efficiency. For inference, we employ a stochastic generation strategy using nucleus sampling with top-p of 0.9 and temperature of 1.0 to extract answers. Specifically, we perform five independent runs using distinct random seeds (from 0 to 4) to ensure reproducibility and reliability.

Additionally, the training and inference schemes of ReGuLaR are presented in Algorithms 1 and 2.

Algorithm 1 Training Scheme of ReGuLaR

Require: Training dataset $\mathcal{D} = \{(\mathcal{Q}, \mathcal{R}, \mathcal{A})\}$, Rendering function f, Visual Encoder v, Adapter g_γ, LLM with parameters θ'=$\{\phi, \psi\}$,

Stage 1: Offline Pre-computation
for each sample $(\mathcal{Q}, \mathcal{R}, \mathcal{A}) \in \mathcal{D}$ **do**
  Divide reasoning chain $\mathcal{R}$ into K segments $\{\mathcal{R}_1, ..., \mathcal{R}_K\}$.
  for $k = 1$ to K **do**
    Render segment to image: $\mathcal{I}_k \leftarrow f(\mathcal{R}_k)$
    Extract visual representation: $\boldsymbol{v}_k \leftarrow v(\mathcal{I}_k)$
  end for
  Store pre-computed representations $\boldsymbol{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_K]$.
end for
Stage 2: Training
while not converged **do**
  Sample batch of $(\mathcal{Q}, \mathcal{R}, \mathcal{A}, \boldsymbol{V})$ from $\mathcal{D}$.
  Initialize total loss $\mathcal{L}_{total} \leftarrow 0$.
  Initialize latent reasoning state history $\boldsymbol{Z}_{<1} \leftarrow \emptyset$.
  for $k = 1$ to K **do**
    // 1. Construct Prior
    Compute prior mean via adapter: $\hat{\boldsymbol{z}}_k \leftarrow g_\gamma(\boldsymbol{v}_k)$
    // 2. Sample Posterior (corresponds to equation 3)
    Predict posterior parameters: $\boldsymbol{\mu}_k, \log \boldsymbol{\sigma}_k \leftarrow p_\phi(\mathcal{Q}, \boldsymbol{Z}_{<k})$
    Sample latent reasoning state: $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, $\boldsymbol{z}_k \leftarrow \boldsymbol{\mu}_k + \boldsymbol{\sigma}_k \odot \boldsymbol{\epsilon}$
    Update history: $\boldsymbol{Z}_{<k+1} \leftarrow \boldsymbol{Z}_{<k} \cup \{\boldsymbol{z}_k\}$
    // 3. Compute Step-wise Losses (corresponds to the last two terms in equation 7)
    Latent Reasoning Loss:
    Sample a token $r_j \in \mathcal{R}_k$ and compute $\mathcal{L}^{(k)}_{reasoning} \leftarrow -\log p_\psi(r_j \mid \boldsymbol{z}_k)$
    Regularizer Loss (KL):
    Compute $\mathcal{L}^{(k)}_{KL}$ between $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2)$ and $\mathcal{N}(\hat{\boldsymbol{z}}_k, \boldsymbol{I})$ using equation 10
    Accumulate: $\mathcal{L}_{total} \leftarrow \mathcal{L}_{total} + \mathcal{L}^{(k)}_{reasoning} + \mathcal{L}^{(k)}_{KL}$
  end for
  // 4. Compute Answer Loss (corresponds to the first term in equation 7)
  Compute $\mathcal{L}_{answer} \leftarrow -\sum_{i=1}^{L_a} \log p_{\theta'}(a_i \mid \mathcal{Q}, \boldsymbol{Z}, \mathcal{A}_{<i})$
  $\mathcal{L}_{total} \leftarrow \mathcal{L}_{total} + \mathcal{L}_{answer}$
  Update Parameters: $\theta', \gamma \leftarrow \text{Optimizer}(\nabla \mathcal{L}_{total})$
end while

B More Experimental Results.

B.1 Ablation Studies on Rendering Configuration

In our standard implementation, we adopt the optimal rendering configuration (i.e., those summarized in Table 4) identified in Glyph (Cheng et al., 2025) to ensure consistent visual encoding and maximize the semantic density. Here, to verify the robustness and generalizability of our method across varying rendering configurations, we conduct ablation studies on two pivotal parameters:

Font Size. We compare the performance of our proposed ReGuLaR by varying the font size from 9pt to 20pt. As presented in Table 5, ReGuLaR demonstrates remarkable stability across different font sizes. Specifically, despite substantial variations in font size, the average accuracy fluctuates only

Algorithm 2 Inference Scheme of ReGuLaR

Require: Question $\mathcal{Q}$, Trained LLM with parameters θ'=$\{\phi, \psi\}$, Max reasoning steps K_{max}.
Initialization: Latent reasoning state history $\boldsymbol{Z}_{<1} \leftarrow \emptyset$, Latent reasoning step $k \leftarrow 1$.
// Phase 1: Latent Reasoning
while $k \leq K_{max}$ **do**
  Predict posterior parameters: $\boldsymbol{\mu}_k, \log \boldsymbol{\sigma}_k \leftarrow p_\phi(\mathcal{Q}, \boldsymbol{Z}_{<k})$
  Sample latent reasoning state: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \boldsymbol{z}_k \leftarrow \boldsymbol{\mu}_k + \boldsymbol{\sigma}_k \odot \boldsymbol{\epsilon}$
  Decode representative token: $\hat{r} \sim p_\psi(r \mid \boldsymbol{z}_k)$
  if $\hat{r}$ == <EOS_Reasoning> **then**
    Break
  end if
  Update history: $\boldsymbol{Z}_{<k+1} \leftarrow \boldsymbol{Z}_{<k} \cup \{\boldsymbol{z}_k\}$
  $k \leftarrow k + 1$
end while
// Phase 2: Answer Generation
Initialize answer sequence $\mathcal{A}_{<1} \leftarrow \emptyset$, Answer generation step $i \leftarrow 1$.
while answer not finished **do**
  Sample token: $a_i \sim p_{\theta'}(a \mid \mathcal{Q}, \boldsymbol{Z}, \mathcal{A}_{<i})$
  if a_i == <EOS> **then**
    Break
  end if
  Update answer: $\mathcal{A}_{<i+1} \leftarrow \mathcal{A}_{<i} \cup \{a_i\}$
  $i \leftarrow i + 1$
end while
return Answer $\mathcal{A}$

Table 5: Performance comparison of our proposed ReGuLaR across different font sizes, where we report the averaged number and 95% confidence interval (±) on Accuracy (Acc. %) and Reasoning Length (# L) on four math reasoning datasets.

Font Size	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average	
	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L
20pt	$34.4_{\pm 0.25}$	$4.46_{\pm 0.17}$	$8.02_{\pm 0.15}$	$4.11_{\pm 0.45}$	$48.4_{\pm 0.23}$	$2.17_{\pm 0.36}$	$88.4_{\pm 0.11}$	$2.62_{\pm 0.28}$	44.8	3.34
16pt	$33.2_{\pm 0.22}$	$4.15_{\pm 0.20}$	$8.74_{\pm 0.07}$	$5.23_{\pm 0.33}$	$48.9_{\pm 0.26}$	$1.86_{\pm 0.15}$	$87.2_{\pm 0.25}$	$2.09_{\pm 0.09}$	44.5	3.33
12pt	$34.0_{\pm 0.33}$	$4.01_{\pm 0.25}$	$8.51_{\pm 0.09}$	$4.81_{\pm 0.56}$	$51.1_{\pm 0.41}$	$2.51_{\pm 0.74}$	$87.6_{\pm 0.19}$	$2.16_{\pm 0.06}$	45.3	3.37
9pt	$34.9_{\pm 0.26}$	$3.69_{\pm 0.21}$	$8.27_{\pm 0.14}$	$4.12_{\pm 0.48}$	$50.1_{\pm 0.39}$	$2.02_{\pm 0.18}$	$89.2_{\pm 0.27}$	$2.28_{\pm 0.27}$	45.6	3.03

marginally (ranging from 44.5% to 45.6%), while the reasoning length remains largely consistent. These results indicate that our method effectively captures semantic information regardless of the text's scale, ensuring robust performance without requiring precise font size tuning.

Rendering Density (DPI). We investigate the sensitivity of our proposed ReGuLaR to information density by adjusting the DPI from 72 to 300. As presented in Table 6, the performance remains highly consistent across this wide range. Notably, ReGuLaR achieves comparable average accuracy at both the lowest density (45.6% at 72 DPI) and the highest density (45.2% at 300 DPI). This suggests that ReGuLaR is resilient to variations in image clarity and pixel density, capable of extracting reliable features under diverse DPI settings.

B.2 Ablation Studies on Visual Encoder Modes

In our standard implementation, we employ the trained visual encoder from DeepSeek-OCR (Wei et al., 2025) to encode each rendered image into a sequence of visual tokens, which are subsequently aggregated via mean pooling to derive a single and compact visual-semantic representation. In particular, this trained visual encoder supports four modes: Tiny, Small, Base, and Large, corresponding to resolutions of 512×512, 640×640, 1024×1024, and 1280×1280, resulting in 64, 100, 256, and 400 vision tokens. Depending on the selected mode, the input rendered images are processed via either adaptive resizing (for Tiny/Small) or padding (for Base/Large) to align with the specific

Table 6: Performance comparison of our proposed ReGuLaR across different rendering density (DPI) settings, where we report the averaged number and 95% confidence interval (±) on Accuracy (Acc. %) and Reasoning Length (# L) on four math reasoning datasets.

DPI	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average	
	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L
300	$33.3_{\pm 0.18}$	$3.93_{\pm 0.35}$	$8.07_{\pm 0.15}$	$3.73_{\pm 0.41}$	$50.7_{\pm 0.20}$	$2.16_{\pm 0.28}$	$88.9_{\pm 0.28}$	$2.24_{\pm 0.31}$	45.2	3.02
144	$33.4_{\pm 0.07}$	$4.87_{\pm 0.36}$	$7.67_{\pm 0.06}$	$5.05_{\pm 0.39}$	$49.9_{\pm 0.42}$	$2.19_{\pm 0.31}$	$87.5_{\pm 0.12}$	$3.07_{\pm 0.19}$	44.6	3.80
96	$32.6_{\pm 0.16}$	$4.13_{\pm 0.16}$	$7.87_{\pm 0.13}$	$5.19_{\pm 0.42}$	$48.2_{\pm 0.11}$	$2.27_{\pm 0.03}$	$88.3_{\pm 0.16}$	$2.12_{\pm 0.04}$	44.2	3.43
72	$34.9_{\pm 0.26}$	$3.69_{\pm 0.21}$	$8.27_{\pm 0.14}$	$4.12_{\pm 0.48}$	$50.1_{\pm 0.39}$	$2.02_{\pm 0.18}$	$89.2_{\pm 0.27}$	$2.28_{\pm 0.27}$	45.6	3.03

Table 7: Performance comparison of our proposed ReGuLaR across different visual encoder modes, where we report the averaged number and 95% confidence interval (±) on Accuracy (Acc. %) and Reasoning Length (# L) on four math reasoning datasets.

Mode	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average	
	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L
Large	$33.0_{\pm 0.36}$	$3.77_{\pm 0.24}$	$7.35_{\pm 0.17}$	$4.28_{\pm 0.29}$	$48.5_{\pm 0.41}$	$2.42_{\pm 0.25}$	$88.4_{\pm 0.38}$	$3.08_{\pm 0.12}$	44.3	3.39
Base	$34.3_{\pm 0.46}$	$3.86_{\pm 0.48}$	$7.67_{\pm 0.09}$	$4.60_{\pm 0.36}$	$51.6_{\pm 0.33}$	$2.23_{\pm 0.22}$	$88.6_{\pm 0.41}$	$2.41_{\pm 0.32}$	45.5	3.28
Small	$34.0_{\pm 0.29}$	$3.64_{\pm 0.34}$	$7.53_{\pm 0.12}$	$4.07_{\pm 0.45}$	$52.1_{\pm 0.31}$	$2.71_{\pm 0.29}$	$89.5_{\pm 0.23}$	$2.23_{\pm 0.28}$	45.8	3.16
Tiny	$34.9_{\pm 0.26}$	$3.69_{\pm 0.21}$	$8.27_{\pm 0.14}$	$4.12_{\pm 0.48}$	$50.1_{\pm 0.39}$	$2.02_{\pm 0.18}$	$89.2_{\pm 0.27}$	$2.28_{\pm 0.27}$	45.6	3.03

resolution. Therefore, while the input rendered images remain identical in their original resolution, selecting different modes forces the visual encoder to process them at varying internal resolutions and token budgets. Here, we conduct ablation studies across these four modes to investigate the impact of visual encoding granularity on our method.

As presented in Table 7, the results reveal a counter-intuitive yet profound insight into the visual-semantic compression mechanism: the Tiny mode, despite internally resizing the input rendered images to 512×512 and utilizing only 64 intermediate tokens, achieves performance comparable to that of the high-resolution modes. We attribute this remarkable robustness to the information aggregation nature of our method. Since the intermediate visual tokens are ultimately pooled into a single and compact visual-semantic representation, it relies more on high-level semantic abstraction rather than fine-grained pixel-level details. Therefore, we adopt the Tiny mode as the default configuration, effectively reducing the visual processing overhead by approximately $6\times$ compared with the Large mode while ensuring that the final visual representation remains semantically rich and accurate.

B.3 Ablation Studies on Learning Paradigms

In our standard implementation, we train the proposed ReGuLaR via the unified objective function defined in equation 7, which integrates three critical components to optimize the model jointly: the answer generation loss ($\mathcal{L}^{\text{Latent}}_{\text{answer}}$) directly ensures answer correctness, the latent reasoning loss ($\mathcal{L}^{\text{Latent}}_{\text{reasoning}}$) preserves semantic integrity, and the KL divergence term ($\mathcal{L}^{\text{Latent}}_{\text{KL}}$) regularizes the posterior distribution. To investigate the distinct contribution of each component, we conduct ablation studies by selectively removing the latter two terms. Notably, the answer generation loss (i.e., the first term in equation 7) is retained across all variants, as it serves as the fundamental supervision signal for the reasoning task.

As presented in Table 8, the absence of the KL divergence term ($\mathcal{L}^{\text{Latent}}_{\text{KL}}$) leads to catastrophic failure (i.e., accuracy$< 14\%$), regardless of the latent reasoning loss. In stark contrast, introducing $\mathcal{L}^{\text{Latent}}_{\text{KL}}$ alone significantly boosts performance to 41.9%. This result empirically corroborates our analysis in Section 3.2: without the constraint imposed by the prior distribution, neither the distant supervision from the final answer nor the semantic supervision from textual reconstruction is sufficient. In addition, combining all components achieves the peak performance of 45.6%, demonstrating that the semantic richness from text reconstruction and the structural guidance from distribution regularization are synergistic and mutually indispensable.

Table 8: Performance comparison of our proposed ReGuLaR across different learning paradigms, where we report the averaged number and 95% confidence interval ($\pm$) on Accuracy (Acc. %) and Reasoning Length (# L) on four math reasoning datasets.

$\mathcal{L}_{\text{KL}}^{\text{Latent}}$	$\mathcal{L}_{\text{reasoning}}^{\text{Latent}}$	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average	
		Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L
×	×	$5.69_{\pm 0.13}$	$4.73_{\pm 0.03}$	$1.46_{\pm 0.11}$	$5.11_{\pm 0.05}$	$35.7_{\pm 1.22}$	$2.94_{\pm 0.03}$	$8.33_{\pm 0.11}$	$2.40_{\pm 0.23}$	12.8	3.81
×	✓	$6.52_{\pm 0.26}$	$4.57_{\pm 0.16}$	$1.68_{\pm 0.16}$	$5.54_{\pm 0.12}$	$34.4_{\pm 0.81}$	$1.86_{\pm 0.05}$	$9.93_{\pm 0.39}$	$1.93_{\pm 0.19}$	13.1	3.47
✓	×	$27.3_{\pm 0.38}$	$3.87_{\pm 0.10}$	$6.03_{\pm 0.22}$	$4.71_{\pm 0.34}$	$47.7_{\pm 0.27}$	$1.84_{\pm 0.06}$	$86.4_{\pm 0.57}$	$1.96_{\pm 0.14}$	41.9	3.10
✓	✓	$34.9_{\pm 0.26}$	$3.69_{\pm 0.21}$	$8.27_{\pm 0.14}$	$4.12_{\pm 0.48}$	$50.1_{\pm 0.39}$	$2.02_{\pm 0.18}$	$89.2_{\pm 0.27}$	$2.28_{\pm 0.27}$	45.6	3.03

Table 9: Performance comparison of our proposed ReGuLaR across different modeling strategies, where we report the averaged number and 95% confidence interval ($\pm$) on Accuracy (Acc. %) and Reasoning Length (# L) on four math reasoning datasets.

Strategy	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average	
	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L
Deterministic	$32.6_{\pm 0.34}$	$3.45_{\pm 0.02}$	$7.24_{\pm 0.06}$	$5.65_{\pm 0.52}$	$48.9_{\pm 0.37}$	$2.72_{\pm 0.25}$	$88.1_{\pm 0.14}$	$2.22_{\pm 0.29}$	44.2	3.51
Probabilistic	$34.9_{\pm 0.26}$	$3.69_{\pm 0.21}$	$8.27_{\pm 0.14}$	$4.12_{\pm 0.48}$	$50.1_{\pm 0.39}$	$2.02_{\pm 0.18}$	$89.2_{\pm 0.27}$	$2.28_{\pm 0.27}$	45.6	3.03

B.4 Ablation Studies on Modeling Strategies

In our standard implementation, we model the latent reasoning process as a probabilistic transition (i.e., equation 3), leveraging the latent reasoning head to predict the distribution parameters $\boldsymbol{\mu}$ and $\log \boldsymbol{\sigma}$ of the next latent reasoning state. To verify the effectiveness of this probabilistic modeling strategy, we conduct ablation studies comparing it against the deterministic variant. Specifically, this deterministic variant directly leverages the latent reasoning head to predict the next latent reasoning state, which is functionally equivalent to a greedy strategy that selects the most likely mean vector.

As presented in Table 9, the results demonstrate the clear superiority of the probabilistic modeling strategy. Specifically, it outperforms the deterministic variant across all datasets, achieving an average accuracy of 45.6%. We attribute the performance drop in the deterministic variant to the "mean collapse" phenomenon. Since the reasoning process often allows for multiple valid subsequent steps, a deterministic predictor tends to output the average of all possible outcomes to minimize the reconstruction error. This results in blurred semantic representations that fail to capture the precise logic required for complex reasoning. In contrast, our probabilistic strategy models the underlying distribution, enabling the sampling of sharp and distinct latent states that preserve semantic integrity.

B.5 Ablation Studies on Regularization Strategies

In our standard implementation, we decompose the reasoning chain $\mathcal{R}$ into K segments and subsequently render them into images to yield K visual representations, which serve as dense semantic anchors to regularize the posterior distribution of the latent reasoning state during training (i.e., illustrated in Figure 1). To verify the distinct contribution of this vision-based regularization, we conduct ablation studies comparing it against the text-based variant. Specifically, this text-based variant directly aggregates the embeddings of tokens within the same segment into one textual representation to regularize the posterior distribution, while keeping all other settings invariant.

As presented in Table 10, the results demonstrate a substantial performance advantage for the vision-based regularization strategy. Specifically, it significantly outperforms the text-based variant across all datasets, increasing the average accuracy from 42.3% to 45.6%. We attribute this superiority to the dense information compression capability of the visual modality. The text-based variant, which relies on pooling token embeddings, tends to dilute the structural and topological details of the reasoning chain (e.g., the spatial layout of arithmetic operations), resulting in a "blurred" semantic target. In contrast, the vision-based approach compels the model to align its latent reasoning state with the corresponding rendered image, which serves as a highly compact and structured semantic anchor. This cross-modal constraint forces the model to capture the holistic logic of the segment

Table 10: Performance comparison of our proposed ReGuLaR across different regularization strategies, where we report the averaged number and 95% confidence interval (±) on Accuracy (Acc. %) and Reasoning Length (# L) on four math reasoning datasets.

Strategy	GSM8K-Aug		GSM-Hard		SVAMP		MultiArith		Average	
	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L	Acc.	# L
Text-based	$28.3_{\pm 0.20}$	$3.53_{\pm 0.31}$	$6.53_{\pm 0.11}$	$4.21_{\pm 0.03}$	$47.5_{\pm 0.18}$	$1.97_{\pm 0.03}$	$86.9_{\pm 0.37}$	$2.07_{\pm 0.13}$	42.3	2.95
Vision-based	$34.9_{\pm 0.26}$	$3.69_{\pm 0.21}$	$8.27_{\pm 0.14}$	$4.12_{\pm 0.48}$	$50.1_{\pm 0.39}$	$2.02_{\pm 0.18}$	$89.2_{\pm 0.27}$	$2.28_{\pm 0.27}$	45.6	3.03

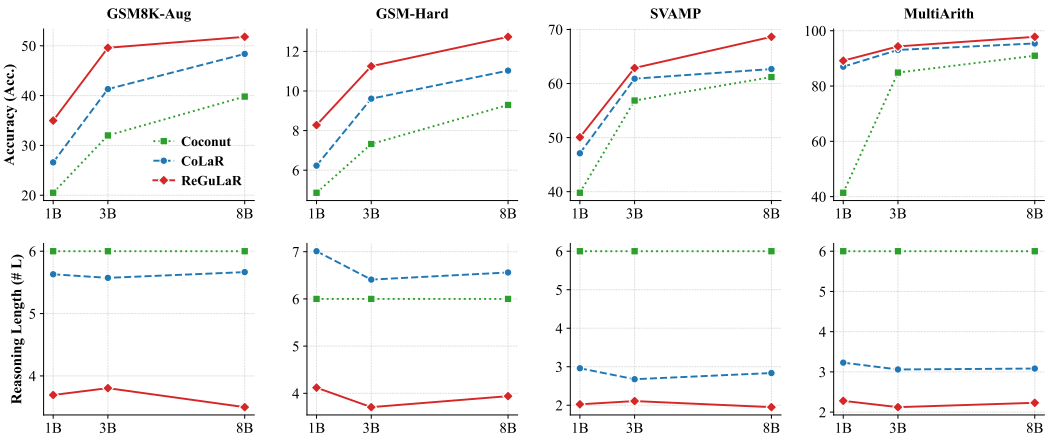


Figure 6: Scalability analysis across varying model sizes, where we employ LLaMA-3.2 (1B, 3B) and LLaMA-3.1 (8B) Instruct variants as the LLM backbones.

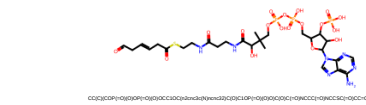
rather than just the average meaning of its tokens, thereby providing a more robust signal for regularization. Notably, as discussed in Section 3.4, these visual representations are strictly confined to the training phase, meaning that they incur no additional cost during inference.

B.6 Further Scalability Analysis of ReGuLaR

Figure 6 further illustrates the performance of ReGuLaR across varying model sizes, where ReGuLaR demonstrates strong positive scaling behavior. Specifically, compared with the top-performing baselines (i.e., CoLaR and Coconut), ReGuLaR consistently maintains the highest accuracy with the shortest reasoning length across all model scales and datasets, highlighting its seamless scalability and potential for broader application in large-scale foundation models.

C Examples of Rendered Reasoning Chains

In the molecular captioning task, we utilize RDKit to generate the corresponding 2D molecular graph for each textual reasoning chain, thereby constructing multi-modal reasoning chains. Subsequently, these reasoning chains are rendered into images, from which visual representations are extracted to regularize the posterior distribution of the latent reasoning state during training. Here, Figure 7 presents two examples of rendered reasoning chains, each shown in two variants: rendered multi-model reasoning chains with explicit 2D molecular graphs and original textual reasoning chains without 2D molecular graphs. For rendered multi-model reasoning chains that include 2D molecular graphs (e.g., Figures 7a and 7c), we position the corresponding 2D molecular graph at the top of the rendered image and annotate it below with its SMILES string. The remainder of the rendering is identical to the counterpart without 2D graphs, consisting solely of textual reasoning steps.



CC(C)(COP(=O)(O)OP(=O)(O)OCC1OC(n2cnc3c(N)ncnc32)C(O)C1OP(=O)(O)O)C(O)C(=O)NCCC(=O)NCCSC(=O)CC=CCC=O

Alright, let's break this down step by step to arrive at a chemically accurate and concise description of the molecule:

Step 1: Decode the SELFIES String and Structural Features

The SELFIES string provided encodes a molecule with multiple functional groups and structural features. Here's what the molecule contains based on its encoding and supplementary structural details:

- **Phosphate groups** (e.g., O=P(O)(O)OP(=O)(O)O), indicating it includes an advanced functionality such as a Coenzyme A (CoA) scaffold.
- **Carbonyl fragments** (C=O), which are commonly found in acyl chains or intermediates in metabolic pathways.
- **Aromatic ring systems** (e.g., c1ncc2nc[nH]c2n1), indicative of purine or pyrimidine bases such as adenine or nicotinamide components in biochemical frameworks.
- **Sulfur-containing groups** (e.g., C=CS), characteristic of CoA thioesters.
- **Double bond fragments** (C=C), which suggest the molecule may belong to a family of unsaturated fatty acyl chains.

Step 2: Interpretation of Fragments

Examining the fragments reveals several telling features:

1. **CoA Core**: The molecule exhibits structural components consistent with a fatty acyl-CoA species, including a nucleotide-based handle (e.g., adenine-like aromatic ring) and thioester functionalities.
2. **Adipoyl Chain Derivative**: Adipoyl refers to a C6 dicarboxylic acid structure, and the molecule appears related to metabolic intermediates derived from adipoyl-CoA processing.
3. **Monounsaturated Chain**: The presence of a single double bond (C=C) suggests a monounsaturation, which aligns with cis-3-enoyl groups often characterized by this feature.
4. **Semialdehyde Conjugation**: Fragments like NC=O further point to semialdehyde functionalities common in biochemical dehydrogenation steps.
5. **Charge State**: The molecule seems to exist in its acid form rather than the deprotonated conjugate base, as described by the conjugate acid notation.

Step 3: Analyze Significant Features

Integrating all structural information and fragments:

- CoA molecules typically serve as acyl transfer facilitators in metabolic processes.
- The cis-3 double bond and adipoyl origin indicate the molecule is likely an intermediate in fatty acid beta-oxidation or elongation processes.
- Conjugate acids of semialdehydes are chemically stable intermediates in biosynthetic pathways.

Step 4: Natural Language Summary

Combining everything into a cohesive description:

<answer> The molecule is a cis-3-enoyl-CoA and a monounsaturated fatty acyl-CoA. It derives from an adipoyl-CoA. It is a conjugate acid of a cis-3,4-didehydroadipoyl-CoA semialdehyde(4-). </answer>

(a) Rendered reasoning chains with 2D (example 1)

Alright, let's break this down step by step to arrive at a chemically accurate and concise description of the molecule:

Step 1: Decode the SELFIES String and Structural Features

The SELFIES string provided encodes a molecule with multiple functional groups and structural features. Here's what the molecule contains based on its encoding and supplementary structural details:

- **Phosphate groups** (e.g., O=P(O)(O)OP(=O)(O)O), indicating it includes an advanced functionality such as a Coenzyme A (CoA) scaffold.
- **Carbonyl fragments** (C=O), which are commonly found in acyl chains or intermediates in metabolic pathways.
- **Aromatic ring systems** (e.g., c1ncc2nc[nH]c2n1), indicative of purine or pyrimidine bases such as adenine or nicotinamide components in biochemical frameworks.
- **Sulfur-containing groups** (e.g., C=CS), characteristic of CoA thioesters.
- **Double bond fragments** (C=C), which suggest the molecule may belong to a family of unsaturated fatty acyl chains.

Step 2: Interpretation of Fragments

Examining the fragments reveals several telling features:

1. **CoA Core**: The molecule exhibits structural components consistent with a fatty acyl-CoA species, including a nucleotide-based handle (e.g., adenine-like aromatic ring) and thioester functionalities.
2. **Adipoyl Chain Derivative**: Adipoyl refers to a C6 dicarboxylic acid structure, and the molecule appears related to metabolic intermediates derived from adipoyl-CoA processing.
3. **Monounsaturated Chain**: The presence of a single double bond (C=C) suggests a monounsaturation, which aligns with cis-3-enoyl groups often characterized by this feature.
4. **Semialdehyde Conjugation**: Fragments like NC=O further point to semialdehyde functionalities common in biochemical dehydrogenation steps.
5. **Charge State**: The molecule seems to exist in its acid form rather than the deprotonated conjugate base, as described by the conjugate acid notation.

Step 3: Analyze Significant Features

Integrating all structural information and fragments:

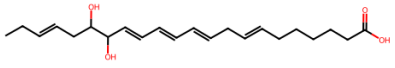
- CoA molecules typically serve as acyl transfer facilitators in metabolic processes.
- The cis-3 double bond and adipoyl origin indicate the molecule is likely an intermediate in fatty acid beta-oxidation or elongation processes.
- Conjugate acids of semialdehydes are chemically stable intermediates in biosynthetic pathways.

Step 4: Natural Language Summary

Combining everything into a cohesive description:

<answer> The molecule is a cis-3-enoyl-CoA and a monounsaturated fatty acyl-CoA. It derives from an adipoyl-CoA. It is a conjugate acid of a cis-3,4-didehydroadipoyl-CoA semialdehyde(4-). </answer>

(b) Rendered reasoning chains without 2D (example 1)



CCC=CCC(O)C(O)C=CC=CC=CCC=CCCCCCC(=O)O

To interpret the molecular structure represented by the provided SELFIES and structural information, we will follow a systematic approach:

1. **Identification of Key Features**:
- Start with the provided structural features which indicate the absence of any rings, including aromatic rings.
- The molecular weight is noted as approximately 362.51 g/mol, which gives us a clue about the size of the molecule.

2. **Analyzing the Fragments**:
- The fragments indicate several features:
- Multiple carbon-carbon double bonds (as evidenced by the C=C fragments).
- Hydroxyl groups (indicated by the presence of O, especially in the O=CO and branching labels).
- A long carbon chain (as indicated by fragments like C=C and CC featuring a sequence of connected carbon atoms).

3. **Constructing the Molecular Backbone**:
- By interpreting the fragments marked with branches, we can ascertain that the structure is composed of a long, linear or slightly branched chain of carbon atoms modified with double bonds and hydroxyl groups.
- The identification of a conjugate acid suggests that the molecule may possess acidic properties, likely due to the carboxylic acid functional group.

4. **Linking to Biological Significance**:
- The mention of "specialised pro-resolving mediators" indicates it is biologically relevant. The inclusion of "xenobiotic metabolite" underscores the molecule may interact with biological systems, possibly as a signaling molecule in inflammatory pathways.

5. **Final Identification Based on Structure and Function**:
- The structure suggested corresponds to an unsaturated fatty acid with specific stereochemistry (noted by Z and E designations). The hydroxyl groups at specific positions confirmed it is a polyunsaturated fatty acid derivative.

6. **Molecule Naming**:
- Given the fragments and biological context, I deduce the complete name. Dihydroxydocosapentaenoic acid (DiHDPE) fits as it represents the
- It carries hydroxy groups at the C16 and C17 positions on the docosapentaenoic acid backbone, producing the final connectivity and stereochemistry mentioned.

After running through these points, the concise description can be articulated as follows:

<answer> The molecule is a dihydroxydocosapentaenoic acid (DiHDPE) that is (7Z,10Z,13E,14E,19Z)-docosapentaenoic acid carrying two hydroxy substituents at the 16 and 17-S positions. It is an intermediate of specialised proresolving mediators. It has a role as a human xenobiotic metabolite and a specialised pro-resolving mediator. It is a conjugate acid of a 16,17(S)-dihydroxy-(7Z,10Z,13E,14E,19Z)-docosapentaenoate. </answer>

(c) Rendered reasoning chains with 2D (example 2)

To interpret the molecular structure represented by the provided SELFIES and structural information, we will follow a systematic approach:

1. **Identification of Key Features**:
- Start with the provided structural features which indicate the absence of any rings, including aromatic rings.
- The molecular weight is noted as approximately 362.51 g/mol, which gives us a clue about the size of the molecule.

2. **Analyzing the Fragments**:
- The fragments indicate several features:
- Multiple carbon-carbon double bonds (as evidenced by the C=C fragments).
- Hydroxyl groups (indicated by the presence of O, especially in the O=CO and branching labels).
- A long carbon chain (as indicated by fragments like C=C and CC featuring a sequence of connected carbon atoms).

3. **Constructing the Molecular Backbone**:
- By interpreting the fragments marked with branches, we can ascertain that the structure is composed of a long, linear or slightly branched chain of carbon atoms modified with double bonds and hydroxyl groups.
- The identification of a conjugate acid suggests that the molecule may possess acidic properties, likely due to the carboxylic acid functional group.

4. **Linking to Biological Significance**:
- The mention of "specialised pro-resolving mediators" indicates it is biologically relevant. The inclusion of "xenobiotic metabolite" underscores the molecule may interact with biological systems, possibly as a signaling molecule in inflammatory pathways.

5. **Final Identification Based on Structure and Function**:
- The structure suggested corresponds to an unsaturated fatty acid with specific stereochemistry (noted by Z and E designations). The hydroxyl groups at specific positions confirmed it is a polyunsaturated fatty acid derivative.

6. **Molecule Naming**:
- Given the fragments and biological context, I deduce the complete name. Dihydroxydocosapentaenoic acid (DiHDPE) fits as it represents the
- It carries hydroxy groups at the C16 and C17 positions on the docosapentaenoic acid backbone, producing the final connectivity and stereochemistry mentioned.

After running through these points, the concise description can be articulated as follows:

<answer> The molecule is a dihydroxydocosapentaenoic acid (DiHDPE) that is (7Z,10Z,13E,14E,19Z)-docosapentaenoic acid carrying two hydroxy substituents at the 16 and 17-S positions. It is an intermediate of specialised proresolving mediators. It has a role as a human xenobiotic metabolite and a specialised pro-resolving mediator. It is a conjugate acid of a 16,17(S)-dihydroxy-(7Z,10Z,13E,14E,19Z)-docosapentaenoate. </answer>

(d) Rendered reasoning chains without 2D (example 2)

Figure 7: Examples of rendered reasoning chains for the molecular captioning task.