LaDiR: Latent Diffusion Enhances LLMs for Text Reasoning

Anonymous authors
Paper under double-blind review

Abstract

Large Language Models (LLMs) demonstrate their reasoning ability through chain-of-thought (CoT) generation. However, LLM's autoregressive decoding may limit the ability to revisit and refine earlier tokens in a holistic manner, which can also lead to inefficient exploration for diverse solutions. In this paper, we propose *LaDiR* (**La**tent **Di**ffusion **R**easoner), a novel reasoning framework that unifies the expressiveness of continuous latent representation with the iterative refinement capabilities of latent diffusion models for an existing LLM. We first construct a structured latent reasoning space using a Variational Autoencoder (VAE) that encodes text reasoning steps into blocks of thought tokens, preserving semantic information and interpretability while offering compact but expressive representations. Subsequently, we utilize a latent diffusion model that learns to denoise a block of latent *thought tokens* with a blockwise bidirectional attention mask, enabling longer horizon and iterative refinement with adaptive test-time compute. This design, combined with explicit diversity guidance during diffusion inference, enables the generation of multiple diverse reasoning trajectories that explore distinct regions of the latent space, rather than producing repetitive solutions as often occurs in standard autoregressive sampling. We conduct evaluations on a suite of mathematical reasoning and planning benchmarks. Empirical results show that LaDiR consistently improves accuracy, diversity, and interpretability over existing autoregressive, diffusion-based, and latent reasoning methods, revealing a new paradigm for text reasoning with latent diffusion.

1 Introduction

Large language models (LLMs) have demonstrated remarkable reasoning abilities through extensive pretraining on human languages, yet the inherent limitations of the autoregressive (AR) paradigm are becoming increasingly difficult to overlook (Zhou et al., 2024b; Bachmann & Nagarajan, 2025). As shown in Fig. 1 (top left), their sequential nature prevents revising earlier tokens, making self-refinement inefficient and difficult (Chen et al., 2024; Huang et al., 2024). Moreover, AR models with discrete CoT generate a linear chain of thought (CoT) (Dziri et al., 2023; Wei et al., 2023), which limits reasoning diversity and restricts exploration of multiple valid solutions (Naik et al., 2024; Yu et al., 2024).

Diffusion models (Ho et al., 2020), originally introduced for generation in continuous domains like images, have recently gained attention in text generation for their ability to maintain global coherence and enable iterative refinement (Ye et al., 2024b; Nie et al., 2025; Lou et al., 2023; Yu et al., 2025c; Weligalle, 2025; Sahoo et al., 2024; Gulrajani & Hashimoto, 2023). Moreover, prior works have explored continuous or latent diffusion for language generation (Li et al., 2022; Lovelace et al., 2024; Zhang et al., 2023; Lovelace et al.; Cetin et al., 2025), operating diffusion in latent spaces obtained from text autoencoders or token-embedding spaces. Existing works largely emphasize the parallelization properties of diffusion models (Israel et al., 2025; Nie et al., 2025; Weligalle, 2025) or evaluate fluency in text generation (Li et al., 2022; Lovelace et al., 2024; Zhang et al., 2023; Lovelace et al.). Arguably, a more important direction is to ask: *How can these approaches enhance the reasoning capabilities of LLMs?* We focus on one particularly promising capability: *the ability to self-correct and refine reasoning chains at semantic levels in latent space*. As shown in Fig. 1 (top right), such self-refinement cannot be achieved by discrete diffusion language models that merely transit into masked tokens.

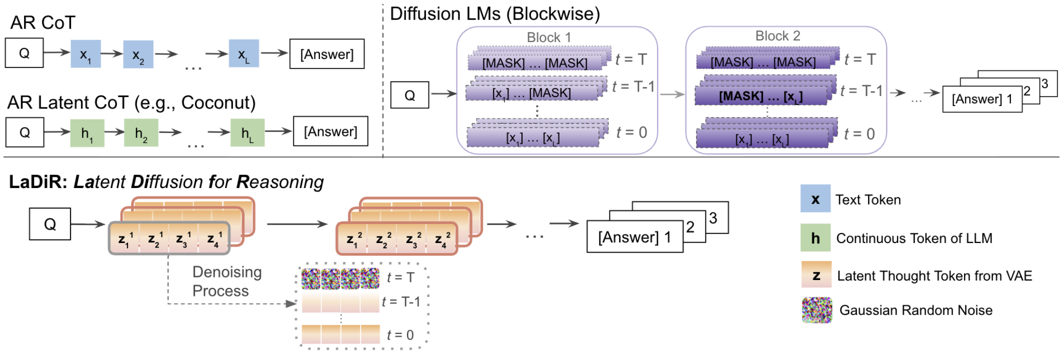


Figure 1: Comparison of reasoning paradigms: autoregressive CoT and latent CoT generate discrete or continuous tokens sequentially; diffusion LMs iteratively refine masked tokens into text in parallel; and our proposed method LaDiR reasons via latent diffusion over thought tokens, enabling iterative refinement at semantic level and diverse solution exploration.

To address this limitation, we introduce LaDiR (**La**tent **Di**ffusion **R**easoner), a flexible reasoning framework that encodes high-level semantic representations of reasoning steps into continuous latent tokens via a Variational Autoencoder (VAE) as *latent thought tokens*, and trains a latent diffusion model over them to perform reasoning. This bridges the gap between surface-level token refinement and deeper semantic reasoning. After the reasoning process, the model generates final answer tokens conditioned on the generated latent thought tokens. Unlike prior latent diffusion works for text generation (Li et al., 2022; Lovelace et al., 2024; Zhang et al., 2023), which focus on fluent text generation, our framework is explicitly designed for latent *reasoning*: it learns causal dependencies across reasoning steps through blockwise diffusion, propagates answer correctness signals back to latent tokens.

Our proposed paradigm establishes a new reasoning framework as a post-training method, bringing several distinctive advantages. First, the iterative refinement ability of diffusion enables a better trade-off between accuracy and test-time compute, as additional denoising steps can be flexibly allocated to improve performance. Second, our framework introduces a diversity-guidance mechanism that applies repulsive forces during diffusion inference, pushing latent trajectories apart within a batch to explore multiple *diverse* reasoning paths, where as AR models tend to collapse to similar trajectories. Finally, leveraging a VAE-based latent space enhances interpretability over continuous diffusion models, making the reasoning process more transparent and readable.

Experimentally, we demonstrate that diffusion-based latent reasoning is not only more accurate but also qualitatively different from prior approaches. On math reasoning benchmarks, including GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021), where Coconut (Hao et al., 2024) fails to surpass AR CoT supervised finetuning (SFT), LaDiR consistently outperforms it on average across 7 benchmarks with the LLaMA 3.1 8B model (Dubey et al., 2024). This suggests that modeling reasoning at the *semantic level*, rather than at the token level, may lead to more faithful intermediate steps that accumulate into stronger final answers. Moreover, on the Countdown planning task, LaDiR shows over 30% absolute improvement in both Pass@1 and Pass@100, indicating that latent thought tokens potentially enhance global planning ability, while *parallel diversity exploration* enables the model to generate diverse reasoning paths. Together, these findings suggest that diffusion-based latent reasoning provides a principled way to balance accuracy and diversity—key ingredients for advancing beyond sequential autoregressive reasoning.

2 Preliminaries

This section introduces key concepts and notations in VAE and latent diffusion models (Rombach et al., 2022). Detailed formulations and background information are provided in Appendix B.

2.1 Variational Autoencoder

A Variational Autoencoder (VAE) (Kingma & Welling, 2013) learns a latent representation of data by balancing reconstruction accuracy and prior regularization. Let $x \in \mathbb{R}^{L \times d_x}$ denote a sequence of token embeddings with length L and embedding dimension d_x, and $z \in \mathbb{R}^{M \times d_z}$ denote latent representations with M latent tokens of dimension d_z. We adopt the β-VAE (Higgins et al., 2017), where a scaling factor β controls this trade-off:

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] + \beta \, \mathrm{KL}\big(q_\phi(z|x) \,\|\, p(z)\big). \quad (1)$$

Here, $q_\phi(z|x)$ is the encoder distribution parameterized by ϕ, $p_\theta(x|z)$ is the decoder likelihood parameterized by θ, and $p(z) = \mathcal{N}(0, I)$ is the prior distribution over latents.

Larger β values encourage disentangled and structured latent spaces, at the cost of reconstruction fidelity. During inference, the encoder of VAE produces a mean/variance pair $\{(\mu, \sigma)\}$, and a latent token z is sampled as $z = \mu + \sigma \odot \epsilon,\ \epsilon \sim \mathcal{N}(0, I)$.

2.2 Latent Diffusion and Flow Matching

Latent diffusion models (Ho et al., 2020; Rombach et al., 2022) generate data by denoising latent variables from Gaussian noise in the latent space of a VAE, which preserves high-level semantic structure. Diffusion can also be viewed as a continuous-time generative flow trained via *flow matching* (Lipman et al., 2022), which we adopt as our primary framework for its superior performance (see Appendix C). The training and inference processes are as follows:

Training Let $\{z_t\}_{t\in[0,1]}$ denote a path interpolating between clean data $z_0 \sim p_{\text{data}}$ and noise $\epsilon \sim \mathcal{N}(0, I) : z_t = (1-t)z_0 + t\epsilon$. This path is controlled by an ordinary differential equation (ODE) $u^\star(z_t, t) = \frac{dz_t}{dt} = \epsilon - z_0$, where $u^\star$ is the target velocity field. A neural network $u_\theta(z_t, t)$ is trained to approximate $u^\star$ by minimizing the flow matching loss:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t\sim\mathcal{U}(0,1),\, z_0\sim p_{\text{data}},\, z_1\sim\mathcal{N}(0,I)} \Big[\|u_\theta(z_t, t) - u^\star(z_t, t)\|^2 \Big]. \tag{2}$$

Inference. At generation time, the process begins from Gaussian noise $z_1 \sim \mathcal{N}(0, I)$. The learned velocity field $u_\theta(z_t, t)$ is then integrated backward in time using an ODE solver as follows: $z_{t-\Delta t} = z_t - \Delta t\, u_\theta(z_t, t)$, with steps from $t = 1$ to $t = 0$. The final state z_0 corresponds to a clean latent representation. This procedure naturally supports *iterative refinement*, as each integration step progressively transforms noise into a coherent latent z.

2.3 Block Diffusion

To support flexible and variable-length sequence generation, we employ a *block diffusion* scheme (Arriola et al., 2025) that integrates autoregressive modeling with diffusion. Instead of applying diffusion to individual latent tokens or full sequence, the sequence is divided into contiguous blocks, and diffusion is performed at the block level. This hybrid design retains the open-ended generation of autoregressive models while introducing global coherence within each block. See Appendix B.4 for details.

3 Methodology

Our approach separates reasoning from answering. A variational autoencoder (VAE) constructs a latent space of intermediate reasoning steps, encoding each step as a block of *thought tokens*. We further utilize a reasoning model that predicts and refines *thought tokens* via latent diffusion, and then generates the final answer tokens conditioned on the denoised latent tokens.

3.1 Architecture

We employ a VAE to construct the latent space of intermediate reasoning steps, and a reasoning model that predicts latent tokens via diffusion and generates the final text answer.

Blockization. We separate the chain-of-thought (CoT) reasoning and the final answer in the dataset using the prefix `‘‘The answer is’’`. The text preceding the prefix is treated as CoT c, while the text following is treated as the final answer y. We then split c into individual sentences, each treated as a *block* of latent tokens with block size L_b:

$$\mathbf{Z}^{(b)} = \{z_1^{(b)}, \ldots, z_{L_b}^{(b)}\}, \quad b = 1, \ldots, N.$$

This one-sentence-per-block design ensures that each reasoning step is localized in latent space.

VAE architecture. As shown in Figure 2 (left), our VAE encoder is initialized from a pretrained LLM and fine-tuned with all parameters, along with L_b learnable embeddings. The encoder's last hidden state is passed through two linear projections to obtain the mean μ and variance σ^2, from which we sample $\mathbf{Z}^{(b)} \sim \mathcal{N}(\mu, \sigma^2)$. The decoder is a *frozen* pretrained LLM that conditions on the sampled $\mathbf{Z}^{(b)}$ to reconstruct the corresponding block of text. This design enables the encoder to compress each reasoning step into a structured latent representation aligned with the semantic space of the language model.

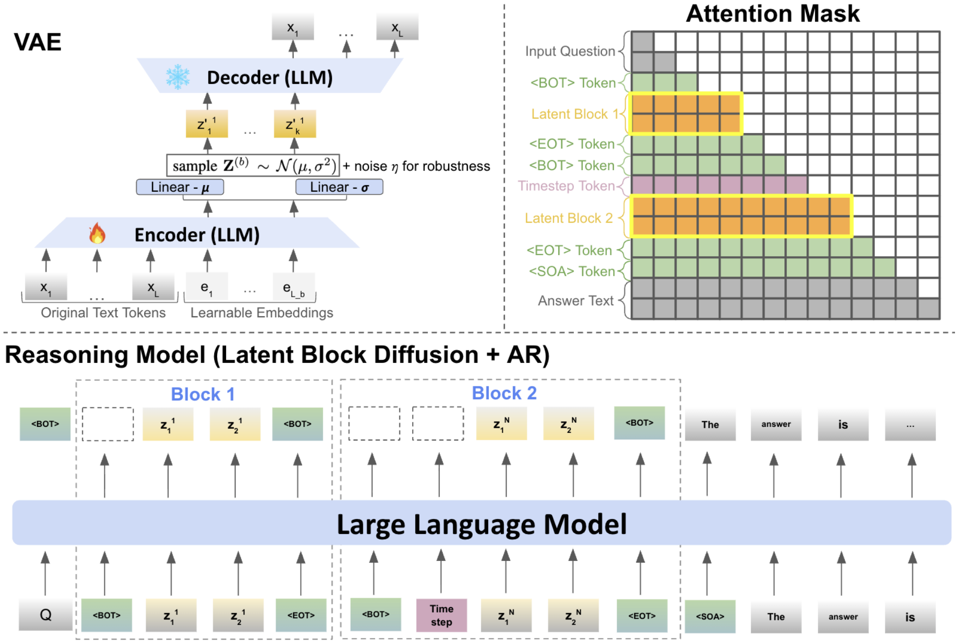


Figure 2: Illustration of our block-wise latent reasoning framework. A question Q is first input as condition to generate latent blocks, each delimited by <BOT> and <EOT>. For each block, the model iteratively denoises latent tokens $\hat{\mathbf{Z}}^{(b)}$ across timesteps, with bidirectional attention inside a block and causal attention across blocks. The reasoning process terminates when the model emits the <SOA> token, after which the model generates the answer text autoregressively.

Reasoning model architecture. We utilize an existing LLM as our reasoning model. As illustrated in Fig. 2, consider the prediction of the second latent block. After the input question Q, we insert a special token <BOT> to mark the start of a block, followed by the first block tokens $\mathbf{Z}^{(\mathbf{1})}$, and a token <EOT> to mark its end. For the second block, since it is being predicted, we add a timestep embedding between <BOT> and $z_1^{(2)}$ to encode the timestep information. Once the latent reasoning process is complete, we switch to text generation mode by appending a <SOA> token to indicate the start of the answer, which is then generated autoregressively. To balance lookahead and variable-length generation, we adopt a hybrid attention mask $\mathcal{M}$ (Fig. 2, top right). Within each block, tokens attend *bidirectionally*, enabling the model to internally *reason* over a horizon defined by the block size and capture richer local dependencies. Across blocks, attention is strictly *causal*, so later steps depend on earlier ones in an autoregressive manner.

3.2 Training

We train the two components separately: the VAE is first trained to learn latent representations of *thought tokens*, after which the reasoning model is trained to predict these *thought tokens*. We describe each stage in turn, beginning with VAE training and followed by reasoning model training.

3.2.1 VAE Training

We build on the standard β-VAE training and inference framework described in Section 2.1, with the following adaptations tailored to our task.

Robustness augmentations. To improve generalization and make the latent space resilient to noise and input variability, we introduce two augmentation strategies during training:

- **Latent Gaussian noise.** For each latent token $z_i^{(b)}$, we inject isotropic Gaussian perturbations:

$$z'^{(b)}_i = z_i^{(b)} + \eta_i, \quad \eta_i \sim \mathcal{N}(0, k^2 I)$$

  where we find $k = 3$ achieve the best downstream performance. This enhances robustness by smoothing the latent space and mitigating sensitivity to small semantic variations.

- **Input token substitution.** For the encoder input sequence, with probability $p = 0.3$ we replace a token with another randomly chosen token (sampled uniformly from the LLM vocabulary). This forces the encoder to learn invariances to paraphrasing, typos, or minor corruptions in input text, ensuring that latent representations capture semantic content rather than exact lexical form.

Together, these augmentations encourage the VAE to build a smoother latent space that is both robust to perturbations and expressive enough to encode thought-level reasoning steps. A more detailed diagram of the VAE can be seen in Appendix F.

3.3 Reasoning Model Training

After constructing a latent reasoning space with the VAE, we train a latent diffusion model f_ψ from the same pretrained LLM as in our VAE model to denoise latent blocks, gradually transforming noisy latent representations to coherent reasoning blocks. Empirically, we observe that training with the flow-matching objective yields the best performance (see Appendix C), and therefore adopt it as our default training objective in the paper.

Answer Token Loss. While f_ψ learns to predict latent reasoning trajectories, to avoid explicitly decoding these steps through the VAE decoder for efficiency during inference, we use the same transformer backbone ψ with a LM head to autoregressively predict answer text tokens conditioned on the latent reasoning blocks. To this end, given the question q, the reasoning blocks $\mathbf{Z}^{(\leq B)}$, and the past answer tokens $y_{<w}$, the model predicts the next answer token y_w with distribution $p_\psi(y_w \mid q, \mathbf{Z}^{(\leq B)}, y_{<w})$. The training objective for those answer tokens is the cross-entropy loss:

$$\mathcal{L}_{\text{Ans}} = -\sum_{w=1}^{W} \log p_\psi(y_t \mid q, \mathbf{Z}^{(\leq B)}, y_{<w}). \tag{3}$$

Special Token Loss. To explicitly control the number of latent blocks, we introduce a special binary classification head on top of the same LLM transformer backbone ψ. It predicts whether the next block begins with a <SOA> (start-of-answer) or <BOT> (begin-of-thought) token whenever an <EOT> (end-of-thought) token is generated. Formally, let τ index positions of <EOT> tokens in the output. For each τ, the model produces a distribution $p_\psi(s_\tau \mid q, \mathbf{Z}^{(\leq B)}, y_{\leq\tau})$, $s_\tau \in \{$<SOA>, <BOT>$\}$, and we minimize the corresponding classification loss, which supervises the model to predict special tokens given the question q and latent reasoning blocks up to position τ:

$$\mathcal{L}_{\text{Spec}} = -\sum_{\tau \in \mathcal{T}_{\text{EOT}}} \log p_\psi(s_\tau \mid q, \mathbf{Z}^{(\leq B)}). \tag{4}$$

3.3.1 Stage 1: Teacher-forcing training

In the first stage, the model is trained under a *teacher-forcing* regime, where it has access to oracle latent blocks produced by the VAE encoder, denoted as $\mathbf{Z}^{(1:B)}$. At every step, these oracle latents are concatenated between special tokens <BOT> and <EOT> and provided as context to the flow-matching model f_ψ. The overall training objective jointly optimizes flow matching on latent blocks and cross-entropy supervision on both final answers and special tokens:

$$\mathcal{L} = \lambda_{\text{FM}}\,\mathcal{L}_{\text{FM}} + \lambda_{\text{Ans}}\,\mathcal{L}_{\text{Ans}} + \lambda_{\text{Spec}}\,\mathcal{L}_{\text{Spec}}, \tag{5}$$

where $\mathcal{L}_{\text{FM}}$ is defined in Eq. 2.

3.3.2 Stage 2: Rollout training

After Stage 1, there is a mismatch between training and inference. During inference, the model must be conditioned on previous self-generated latents without access to oracle latents, suffering from error accumulation issue. To address this issue, Stage 2 adopts an *rollout* training. We keep the same number of blocks B as in the ground truth, but instead of conditioning on oracle latents, the model generates its own latents $\tilde{\mathbf{Z}}^{(1:B)}$ from random noise using a fewer denoising steps (i.e., 50 → 10, following FlowGRPO (Liu et al., 2025)). We keep the gradients on $\tilde{\mathbf{Z}}^{(1:B)}$ during denoising, allowing answer supervision to backpropagate through the trajectory and directly *shape* latent predictions. To avoid latent collapse as in Coconut w/o curriculum learning (Hao et al., 2024), we keep the flow matching loss. Therefore, the training objective is same as Eq. 5.

3.4 Reasoning Model Inference

At inference time, the model generates a chain of latent reasoning blocks and subsequently produces the final answer in text space. The process unfolds in two phases: (i) latent block generation via iterative denoising, and (ii) answer generation via autoregressive decoding.

Iterative denoising. Following the inference process of the standard latent diffusion model as in 2.2, for each block, we initialize with a Gaussian noise, and we gradually transforms the noise into a semantically coherent latent reasoning block $\hat{\mathbf{Z}}^{(b)}$.

Stopping criterion. Latent block generation continues until the model explicitly predicts the special token `<SOA>` (*start of answer*). This token signals that sufficient reasoning has been performed and the model should transition from block diffusion to final answer generation.

Answer generation. Once reasoning terminates, conditioned on the generated latent reasoning sequence $\hat{\mathbf{Z}}^{(1:\hat{B})}$ and the input question x, the model predicts output tokens $y = (y_1, \dots, y_T)$ autoregressively.

Diversity improvement in parallel. Unlike AR models that generate a single reasoning trajectory sequentially, our framework can generate multiple diverse reasoning trajectories in parallel within a batch. To encourage exploration of alternative solutions, we incorporate two complementary mechanisms:

1. **Increased initial noise.** By sampling with an increased variance $\tilde{\sigma}^2$ as initial noise scale, we broaden the distribution of starting points for latent trajectories. This enables the same input question to yield diverse reasoning sequences across runs, improving coverage of alternative solution strategies.

2. **Diversity gradient guidance.** At each denoising step, we enhance diversity by adding a repulsion term to push the latent tokens in a batch apart. First, we compute a bandwidth parameter σ as the median pairwise distance between the latent tokens in a batch at the current step $\sigma = \mathrm{median}_{i<j} \; \|z_i - z_j\|_2$.

   The repulsion force field for a latent token z_i is then defined as

   $$\mathbf{F}(z_i) = \sum_{j \neq i} 2 \left(1 - \frac{\|z_i - z_j\|_2^2}{\sigma^2}\right) \exp\Big(-\tfrac{\|z_i - z_j\|_2^2}{\sigma^2}\Big)\,(z_i - z_j), \forall j \leq B, \tag{6}$$

   where z_j is any other latent token in the same batch with batch size B. We apply strong repulsion at the beginning of inference and gradually decay its effect over time. Specifically, the time-dependent scale is defined as $\gamma_t = \gamma_{\max} \left(\frac{t}{T}\right)$, where T is the total number of inference steps, t decreases from T to 0, $\gamma_{\max}$ is the initial repulsion strength as a hyperparameter.

   Finally, the diversity-guided prediction combines the base model output with the repulsion gradient, in a form analogous to classifier-free guidance (Ho & Salimans, 2022): $\hat{z}_{t-1} = f_\psi(\mathbf{x}_t, t, x) + \gamma_t \, \mathbf{F}(z)$, where $f_\psi(\mathbf{x}_t, t, x)$ is the model's prediction at step t.

Together, these mechanisms enhance the stochasticity and coverage of latent reasoning while preserving convergence to valid solutions.

4 Experiments

We evaluate LaDiR across two domains: mathematical reasoning (7 datasets) and puzzle planning (Countdown), comparing to AR, latent, and diffusion baselines. Our experiments demonstrate its effectiveness on benchmark datasets, while ablation studies in Section 4.3 and analyses 4.4 provide further insights into the contributions of individual components. See Appendix E for experimental details.

4.1 Mathematical Reasoning

We begin by assessing LaDiR on a range of mathematical reasoning benchmarks, covering both in-domain datasets, where training and test distributions are closely aligned, and out-of-domain benchmarks that require generalization to unseen problems.

Datasets We fine-tune pretrained LLMs on the **DART-MATH** dataset (Tong et al., 2024b), a large-scale dataset synthesized to enhance mathematical reasoning. For evaluation, we adopt two in-domain benchmarks, **Math** (Hendrycks et al., 2021) and **GSM8K** (Cobbe et al., 2021), and five out-of-domain benchmarks to assess generalization: **College-Math** (Tang et al., 2024b), **DeepMind-Math** (Saxton et al., 2019), **OlympiaBench-Math** (He et al., 2024), **TheoremQA** (Chen et al., 2023), and **Fresh-Gaokao-Math-2023** (Tang et al., 2024b). Detailed dataset descriptions are provided in Appendix E.

Method	**In-Domain**		**Out-of-Domain**					**Avg.**
	MATH	GSM8K	Gaokao	DM-Math	College	Olympia	TheoremQA	
			Masked Diffusion Methods - LLaDA 8B					
Base Model	36.2	77.3	10.2	29.8	30.2	3.0	16.6	29.0
CoT SFT	39.0	82.3	20.1	43.7	38.9	5.9	20.9	35.8
			Autoregressive Methods - LLaMA 3.1 8B					
Sol-Only SFT	13.3	16.4	0.0	18.2	15.9	4.7	16.9	12.2
CoT SFT	43.1	**84.5**	30.7	**47.8**	45.7	10.1	21.2	40.4
iCoT	35.2	61.8	30.0	30.6	37.6	8.3	19.5	31.8
Pause Token	42.1	83.9	28.3	42.4	31.5	3.5	8.3	34.2
Coconut	37.3	68.3	26.8	33.5	40.2	5.8	11.4	31.9
Discrete Latent	43.2	83.9	33.3	44.7	47.1	**13.3**	20.3	40.8
			Latent Diffusion Methods - LLaMA 3.1 8B					
LD4LG	32.9	9.1	-	-	-	-	-	-
PLANNER	18.7	5.7	-	-	-	-	-	-
LaDiR (ours)	**45.2**	84.2	**33.4**	46.3	**48.6**	11.9	**22.9**	**41.8**
–w/o Stage 2	30.7	57.8	24.7	32.0	32.8	5.9	11.9	32.6

Table 1: Results for pass@1 accuracy across in-domain and out-of-domain math benchmarks.

Baselines We compare our approach against a diverse set of reasoning methods spanning both autoregressive and diffusion-based paradigms. For autoregressive models, we include **Sol-Only**, trained only on question–solution pairs without intermediate reasoning steps, and **CoT**, trained with full chain-of-thought supervision. We further evaluate several latent reasoning methods: **Implicit CoT (iCoT)** (Deng et al., 2023), which gradually removes explicit CoT tokens through curriculum learning; **Pause Token** (Goyal et al., 2023), which introduces a learnable pause token to provide additional computation before answering; **Coconut** (Hao et al., 2024), which leverages hidden states as latent reasoning tokens through curriculum learning; and **Discrete Latent** (Su et al., 2025), which compresses a span of text (e.g., 16 tokens) into discrete latent codes via VQ-VAE for reasoning. For diffusion-based language models, we compare with the open-sourced **LLaDA 8B** (Nie et al., 2025), evaluated both with and without SFT. To directly compare with prior latent diffusion methods for language generation, we also evaluate **LD4LG** (Lovelace et al., 2024) and **PLANNER** (Zhang et al., 2023). For fair comparison, we use FLAN-T5 as the encoder and LLaMA-3.1-8B as the decoder, and train them on the same reasoning datasets. We utilize LLaMA-3.1 8B (Dubey et al., 2024) as the backbone model for our framework as well as for the AR baselines for fair comparison.

Results As shown in Table 1, our method achieves the strongest overall performance on average, improving the average pass@1 accuracy by 2% over the best prior latent approach. Also, LaDiR achieves higher Pass@100 across all benchmarks, with a 6.1% absolute gain over AR CoT SFT on average (see Table 8 in Appendix). Compared to text-based CoT baselines, our latent reasoning consistently yields more robust solutions, particularly on harder benchmarks such as DM-Math and College-level datasets, where direct text reasoning often struggles with long-horizon consistency. This suggests that reasoning in a latent space at semantic level learns more abstract reasoning patterns. Compared to prior latent approaches (e.g., Coconut), the latent diffusion objective provides a more principled objective for modeling continuous trajectories, leading to stronger generalization to out-of-domain settings such as TheoremQA. Also, incorporating the stage 2 rollout training notably improves performance across all benchmarks, showing its effectiveness in mitigating error accumulation. Moreover, prior latent diffusion methods of LD4LG and PLANNER perform poorly on reasoning tasks, indicating that effective latent reasoning requires more than architectural changes, such as blockwise variable-length diffusion and rollout training for answer alignment. Taken together, these results indicate that LaDiR combines the interpretability benefits of CoT-style reasoning (see Appendix D) and expressiveness of continuous latent space, producing generalizable reasoning traces.

4.2 Puzzle Planning – Countdown

We evaluate the planning ability of our method using *Countdown*, a combinatorial arithmetic game. Given a set of input numbers, the goal is to reach a target in $[10, 100]$ by applying basic operations $\{+, -, \times, \div\}$. Solving a problem thus demands decomposing the target into intermediate subgoals and chaining them correctly. For example, given input numbers $\{97, 38, 3, 17\}$ and target 14, one valid solution is: $97 - 38 = 59$, $59 - 17 = 42$, $42 \div 3 = 14$. Following Gandhi et al. (2024), we construct a dataset of 500k examples, holding out 10% of target numbers for *out-of-distribution*

Model	CD-4 P@1	CD-4 P@100	CD-4 Div.	CD-5 P@1	CD-5 P@100	CD-5 Div.
Dream 7B Base*	16.0	24.7	4.1	4.2	10.3	5.6
MGDM†	**91.5**	95.2	3.2	**46.6**	70.4	4.9
LLaDA 8B SFT	51.2	75.2	5.4	34.4	45.2	6.2
LLaMA 8B SFT	46.7	65.3	3.0	8.9	15.4	3.5
LaDiR	76.6	**96.4**	**7.3**	38.5	**75.2**	**8.9**

Table 2: Results on Countdown tasks. We report Pass@1, Pass@100, and Diversity (Div.). Best results are in **bold**, and second-best are underlined. *Dream 7B Base refers to the open-sourced base model without finetuning on this task.†MGDM is a task-specific small discrete diffusion model rather than a general-purpose language model.

evaluation as test set. We study two settings of growing complexity: CD-4 and CD-5, which use four and five input numbers, respectively.

Baselines We compare our method against both autoregressive and diffusion-based approaches. For the autoregressive setting, we include (1) LLaMA 8B SFT, which shares the same base model as ours and is finetuned on the same dataset. For diffusion-based baselines, we consider (2) LLaDA 8B SFT (Nie et al., 2025) and (3) Dream 7B Base (Ye et al., 2025b), two diffusion-based general-purpose language models (the latter evaluated without finetuning), as well as (4) MGDM (Ye et al., 2024a), a small task-specific multinomial-guided diffusion model trained for Countdown.

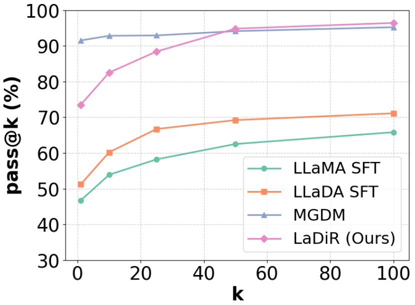


Figure 3: Results for *pass@k* performance on Countdown-4 with $k \in \{1, 10, 25, 50, 100\}$.

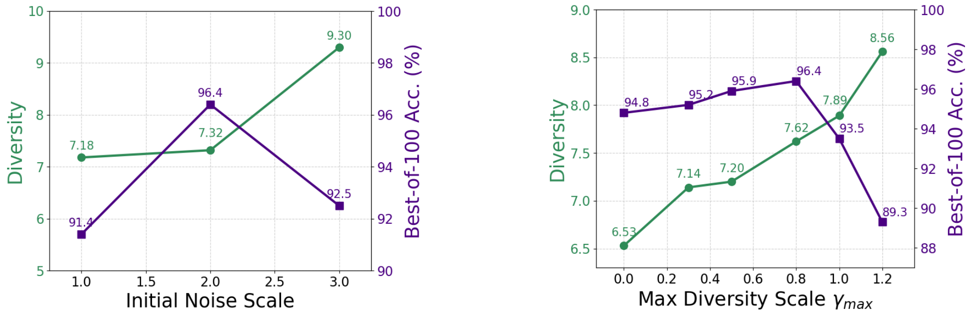
Metrics. We report Pass@1 and Pass@100 accuracy, using an exact string match between the generated arithmetic equations and the ground-truth solution. Pass@k reflects the accuracy that at least one valid solution is found among k samples. In addition, we report Diversity, measured as the number of unique valid solutions discovered among 100 samples. All models are evaluated with a decoding temperature of 1.0.

Implementation Details In this setting, we deliberately *disable* the answer generation and restrict the reasoning process to a single latent block, which is compressed into a fixed-size representation (4 tokens). The model is trained under a teacher-forcing regime and evaluated on decoded text tokens from our VAE, thereby isolating the latent diffusion model's capacity to capture planning dynamics without autoregressive supervision. During inference, we set the initial noise scale to 2 and the maximum diversity guidance scale to 0.8.

Results On the Countdown tasks, as shown in Table 2, our method outperforms autoregressive baselines and remains competitive with specialized diffusion models. In CD-4, it improves Pass@1 by more than 25 points over LLaMA 8B SFT and over 20 points over LLaDA SFT, demonstrating stronger *planning* ability beyond token-by-token generation, while also delivering the best Pass@100 and over two points higher *diversity* than any baseline. On the more challenging CD-5 task, our model surpasses AR baselines by nearly 30 points in Pass@1 and over 30 points in Pass@100, again with the highest diversity. In addition, as shown in Figure 3, our pass@k curve rises steeply with k, surpassing MGDM at larger k. This high pass@k reflects both diverse trajectory exploration and strong potential for reinforcement learning for post-training (Yue et al., 2025).

4.3 Ablation Study

Diversity Scale and Initial Noise We study how inference-time stochasticity and diversity guidance affect solution diversity and accuracy by varying (i) the *initial noise scale*, which controls the variance of Gaussian initialization, and (ii) the *maximum diversity scale* $\gamma_{\max}$, which regulates the repulsion strength among latent tokens (see Sec. 3.4). We evaluate both the average number of unique solutions and best-of-100 accuracy. Table 4 shows that increasing noise from 1 to 2 improves both diversity and accuracy, but excessive noise (scale 3) harms convergence despite higher diversity. For diversity guidance, removing repulsion ($\gamma_{\max} = 0$) yields the lowest diversity, while moderate values (0.3–0.5) strike the best trade-off. Stronger repulsion ($\gamma_{\max} \geq 1.0$) further boosts



(a) Effect of Initial Noise Scale $\tilde{\sigma}^2$ ($\gamma_{max} = 0.8$).

(b) Effect of Max Diversity Scale $\gamma_{max}(\tilde{\sigma}^2 = 2)$.

Figure 4: Ablation study on the hyperparameters for diversity during inference on Countdown-4.

diversity but causes accuracy to drop, suggesting that over-dispersing latents destabilizes reasoning. See Appendix C for additional ablation studies.

4.4 Analysis

Iterative Refinement Table 3 shows how the our flow-matching model refines its reasoning across denoising steps. From pure noise at $T = 1$, the model quickly produces structured equations, though early steps contain arithmetic errors (e.g., off-by-one mistakes). As denoising progresses, partial results stabilize—such as $43 + 9 = 52$ appearing consistently from $T = 0$ onward—and later steps are gradually corrected. By $T = 0.25$, the full reasoning matches the ground truth and remains stable through $T = 0$. This demonstrates that our method exhibits the same iterative refinement ability as reasoning models (Shao et al., 2024), progressively correcting previously generated steps. See Table 10 for an example on GSM8K.

Input	43, 9, 54, 25, 81
GT Answer	43+9=52, 54-25=29, 52+29=81
Decode($\hat{Z}_{t=1}$)	"I .. ex1 ..." (random noise)
Decode($\hat{Z}_{t=0.8}$)	43+8=51, 54-24=30, 51+30=81
Decode($\hat{Z}_{t=0.7}$)	43+10=53, 54-25=28, 53+28=81
Decode($\hat{Z}_{t=0.6}$)	43+9=52, 54-27=27, 52+27=79
Decode($\hat{Z}_{t=0.5}$)	43+9=52, 54-25=29, 52+28=80
Decode($\hat{Z}_{t=0.4}$)	43+9=52, 54-25=29, 52+30=82
Decode($\hat{Z}_{t=0.25}$)	43+9=52, 54-25=29, 52+29=81
Decode($\hat{Z}_{t=0}$)	43+9=52, 54-25=29, 52+29=81

Table 3: Examples of iterative self-refinement of decoded text from the VAE decoder on the Countdown-4 dataset across different denoising timesteps (t).

Adaptive Test-Time Compute. As shown in Figure 5, using more denoising steps consistently improves accuracy across different math benchmarks. For example, increasing from 5 to 10 steps (a 2× compute increase) yields a large jump of +11.7 points in accuracy on average of 7 benchmarks. Starting from 10 steps, tripling the compute to 30 steps provides an additional +4.8 points on average, while a 5× compute increase to 50 steps brings a total gain of +9.8 points on average. These results demonstrate that our method can flexibly trade test-time compute for higher performance as an alternative paradigm in reasoning for long CoT of existing reasoning models (Jaech et al., 2024; Muennighoff et al., 2025; Liu et al., 2024a; Li et al., 2025). This may motivate adaptive policies that dynamically assign more denoising steps to harder queries, maximizing the overall accuracy–compute trade-off.

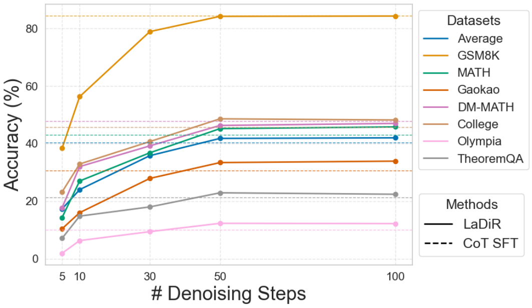


Figure 5: Effect of number of denoising steps on downstream reasoning performance on the math reasoning tasks.

Inference Efficiency Table 4 compares inference latency on the MATH dataset using identical hardware and batch size (8). LaDiR with 10 diffusion steps matches the latency of the AR baseline while achieving comparable Pass@1 and higher Pass@100. With 30 steps, LaDiR offers a flexible accuracy-compute trade-off. This efficiency stems from our compact latent representation: each latent block contains only 4 latent tokens, representing on average 22 text tokens, reducing per-step computation and context length compared to autoregressive decoding over long text sequences. We provide additional analyses on interpretability and semantic-level reasoning in Appendix D.

5 Related Works

Latent Reasoning Latent reasoning methods address token-level limits of chain-of-thought by enabling reasoning in a latent space, yielding more abstract representations through discrete special tokens that expand internal computation or capture unstated steps (Herel & Mikolov, 2024; Pfau et al., 2024; Wang et al., 2024; Zelikman et al., 2024; Jin et al., 2025). Prior works show that reasoning in latent space, rather than discrete tokens, improves performance by allowing LLMs to generate continuous tokens, either self-generated or provided by an auxiliary model (Cheng & Durme, 2024; Hao et al., 2024; Liu et al., 2024b; Shen et al., 2025; Tack et al., 2025; Zhu et al., 2025; Butt et al., 2025; Zhang et al., 2025; Wu et al., 2025). This has been further extended to recurrent or looped architectures that induce latent reasoning internally, removing the need to represent reasoning steps explicitly as tokens (Chen et al., 2025c; Geiping et al., 2025; Mohtashami et al., 2025; Saunshi et al., 2025; Yu et al., 2025b). However, prior latent reasoning approaches lack interpretability, as their continuous states are opaque and difficult to understand or control. whereas our method structures the latent space with a VAE, making each step explicit and thus more transparent.

Method	Latency	Pass@1	Pass@100
LLaMA CoT SFT	4.7	43.1	89.0
LaDiR (10 steps)	4.9	42.9	90.7
LaDiR (30 steps)	14.8	44.9	92.8

Table 4: Inference latency (seconds/example) comparison on MATH dataset.

Latent Diffusion for Language Generation Generative text modeling has recently expanded from autoregressive paradigms to diffusion-based approaches that allow for global iterative refinement. One of the first, Diffusion-LM (Li et al., 2022), frames generation as denoising continuous word embeddings to enable fine-grained control, a concept Lovelace et al. (2023); Lovelace et al. extended by performing diffusion in a compressed latent space for improved quality and diverse generation modes. For sequence-to-sequence tasks, DiffuSeq (Gong et al., 2022) enables parallel generation with high diversity, while PLANNER (Zhang et al., 2023) addresses long-form text by combining a latent semantic diffusion planner with an autoregressive decoder to reduce repetition. Similarly, Cosmos (Meshchaninov et al., 2025) learns a compressed latent space for diffusion, enabling parallel text generation with robust semantic grounding. In specialized domains, Diffusion-Dialog (Xiang et al., 2024) utilizes latent variables to handle open-ended conversations, whereas CodeFusion (Singh et al., 2023) and TreeDiff (Zeng et al., 2025) apply diffusion to code synthesis. While prior latent diffusion models focus on text generation, they lack the granularity to model the multi-step causal dependencies required for reasoning tasks. We address this by introducing block-wise variable-length diffusion and rollout training, explicitly shifting the objective from generating fluent text to optimizing reasoning trajectories that lead to correct answers. Due to the page limit, we discuss further related works in Appendix A.

6 Conclusion

We introduced LaDiR, a latent diffusion reasoner that utilizes the iterative refinement capability of latent diffusion models to perform reasoning at the semantic level, our framework offers three key benefits: (1) better tradeoff between accuracy and test-time compute through iterative denoising steps with self-refinement, (2) parallel and diverse exploration of reasoning trajectories beyond the limitations of sequential autoregression, and (3) enhanced interpretability through semantically meaningful latent representations. Our experiments on mathematical reasoning and planning benchmarks show that LaDiR consistently outperforms AR and diffusion baselines, achieving both higher accuracy and greater diversity in reasoning.

References

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

Gregor Bachmann and Vaishnavh Nagarajan. The pitfalls of next-token prediction, 2025. URL `https://arxiv.org/abs/2403.06963`.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. Forest-of-thought: Scaling test-time compute for enhancing llm reasoning, 2025. URL https://arxiv.org/abs/2412.09078.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

Natasha Butt, Ariel Kwiatkowski, Ismail Labiad, Julia Kempe, and Yann Ollivier. Soft tokens, hard truths. *arXiv preprint arXiv:2509.19170*, 2025.

Edoardo Cetin, Tianyu Zhao, and Yujin Tang. Large language models to diffusion finetuning. *arXiv preprint arXiv:2501.15781*, 2025.

Shoufa Chen, Chongjian Ge, Yuqi Zhang, Yida Zhang, Fengda Zhu, Hao Yang, Hongxiang Hao, Hui Wu, Zhichao Lai, Yifei Hu, Ting-Che Lin, Shilong Zhang, Fu Li, Chuan Li, Xing Wang, Yanghua Peng, Peize Sun, Ping Luo, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Goku: Flow based video generative foundation models, 2025a. URL https://arxiv.org/abs/2502.04896.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. *arXiv preprint arXiv:2305.12524*, 2023.

Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling, 2025b. URL https://arxiv.org/abs/2501.17811.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

Yilong Chen, Junyuan Shang, Zhenyu Zhang, Yanxi Xie, Jiawei Sheng, Tingwen Liu, Shuohuan Wang, Yu Sun, Hua Wu, and Haifeng Wang. Inner thinking transformer: Leveraging dynamic depth scaling to foster adaptive internal thinking, 2025c. URL https://arxiv.org/abs/2502.13842.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations, 2024. URL https://arxiv.org/abs/2412.13171.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 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.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. Faith and fate: Limits of transformers on compositionality, 2023. URL https://arxiv.org/abs/2305.18654.

Lijie Fan, Tianhong Li, Siyang Qin, Yuanzhen Li, Chen Sun, Michael Rubinstein, Deqing Sun, Kaiming He, and Yonglong Tian. Fluid: Scaling autoregressive text-to-image generative models with continuous tokens, 2024. URL https://arxiv.org/abs/2410.13863.

Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D Goodman. Stream of search (sos): Learning to search in language. *arXiv preprint arXiv:2404.03683*, 2024.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL https://arxiv.org/abs/2503.01307.

Silin Gao, Mete Ismayilzada, Mengjie Zhao, Hiromi Wakaki, Yuki Mitsufuji, and Antoine Bosselut. Diffucomet: Contextual commonsense knowledge diffusion, 2024. URL https://arxiv.org/abs/2402.17011.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach, 2025. URL https://arxiv.org/abs/2502.05171.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models, 2025. URL https://arxiv.org/abs/2410.17891.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.

Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36:16693–16715, 2023.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space, 2024. URL https://arxiv.org/abs/2412.06769.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

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. *arXiv preprint arXiv:2103.03874*, 2021.

David Herel and Tomas Mikolov. Thinking tokens for language modeling, 2024. URL https://arxiv.org/abs/2405.08644.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet, 2024. URL https://arxiv.org/abs/2310.01798.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. Accelerating diffusion llms via adaptive parallel decoding, 2025. URL https://arxiv.org/abs/2506.00413.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Mingyu Jin, Weidi Luo, Sitao Cheng, Xinyi Wang, Wenyue Hua, Ruixiang Tang, William Yang Wang, and Yongfeng Zhang. Disentangling memory and reasoning ability in large language models, 2025. URL https://arxiv.org/abs/2411.13504.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks, 2023. URL https://arxiv.org/abs/2210.02406.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

Tianqiao Liu, Zui Chen, Zitao Liu, Mi Tian, and Weiqi Luo. Expediting and elevating large language model reasoning via hidden chain-of-thought decoding, 2024b. URL https://arxiv.org/abs/2409.08561.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Justin Lovelace, Christian K Belardi, Sofian Zalouk, Adhitya Polavaram, Srivatsa R Kundurthy, and Kilian Q Weinberger. Stop-think-autoregress: Language modeling with latent diffusion planning. In *Second Conference on Language Modeling*.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q Weinberger. Latent diffusion for language generation. *Advances in Neural Information Processing Systems*, 36:56998–57025, 2023.

Justin Lovelace, Varsha Kishore, Yiwei Chen, and Kilian Q Weinberger. Diffusion guided language modeling. *arXiv preprint arXiv:2408.04220*, 2024.

Viacheslav Meshchaninov, Egor Chimbulatov, Alexander Shabalin, Aleksandr Abramov, and Dmitry Vetrov. Compressed and smooth latent space for text diffusion modeling. *arXiv preprint arXiv:2506.21170*, 2025.

Amirkeivan Mohtashami, Matteo Pagliardini, and Martin Jaggi. CoTFormer: A chain of thought driven architecture with budget-adaptive computation cost at inference. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=7igPXQFupX.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

Ranjita Naik, Varun Chandrasekaran, Mert Yuksekgonul, Hamid Palangi, and Besmira Nushi. Diversity of thought improves reasoning abilities of llms, 2024. URL https://arxiv.org/abs/2310.07088.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025. URL https://arxiv.org/abs/2502.09992.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2021. URL https://arxiv.org/abs/2112.00114.

Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, Ji Hou, and Saining Xie. Transfer between modalities with metaqueries, 2025. URL https://arxiv.org/abs/2504.06256.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models, 2024. URL https://arxiv.org/abs/2404.15758.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, and Tianlu Wang. Learning to plan & reason for evaluation with thinking-llm-as-a-judge, 2025. URL https://arxiv.org/abs/2501.18099.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Nikunj Saunshi, Nishanth Dikkala, Zhiyuan Li, Sanjiv Kumar, and Sashank J. Reddi. Reasoning with latent thoughts: On the power of looped transformers, 2025. URL https://arxiv.org/abs/2502.17416.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. Codi: Compressing chain-of-thought into continuous space via self-distillation, 2025. URL https://arxiv.org/abs/2502.21074.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data, 2025a. URL https://arxiv.org/abs/2406.04329.

Weijia Shi, Xiaochuang Han, Chunting Zhou, Weixin Liang, Xi Victoria Lin, Luke Zettlemoyer, and Lili Yu. Lmfusion: Adapting pretrained language models for multimodal generation, 2025b. URL https://arxiv.org/abs/2412.15188.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Mukul Singh, José Cambronero, Sumit Gulwani, Vu Le, Carina Negreanu, and Gust Verbruggen. Codefusion: A pre-trained diffusion model for code generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 11697–11708, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, Yuwei Fu, Jing Su, Ge Zhang, Wenhao Huang, Mingxuan Wang, Lin Yan, Xiaoying Jia, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Yonghui Wu, and Hao Zhou. Seed diffusion: A large-scale diffusion language model with high-speed inference, 2025. URL `https://arxiv.org/abs/2508.02193`.

DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. Token assorted: Mixing latent and text tokens for improved language model reasoning. *arXiv preprint arXiv:2502.03275*, 2025.

Jihoon Tack, Jack Lanchantin, Jane Yu, Andrew Cohen, Ilia Kulikov, Janice Lan, Shibo Hao, Yuandong Tian, Jason Weston, and Xian Li. Llm pretraining with continuous concepts, 2025. URL `https://arxiv.org/abs/2502.08524`.

Haotian Tang, Yecheng Wu, Shang Yang, Enze Xie, Junsong Chen, Junyu Chen, Zhuoyang Zhang, Han Cai, Yao Lu, and Song Han. Hart: Efficient visual generation with hybrid autoregressive transformer, 2024a. URL `https://arxiv.org/abs/2410.10812`.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. Mathscale: Scaling instruction tuning for mathematical reasoning. *arXiv preprint arXiv:2403.02884*, 2024b.

Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning, 2024a. URL `https://arxiv.org/abs/2412.14164`.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *Advances in Neural Information Processing Systems*, 37:7821–7846, 2024b.

Emile van Krieken, Pasquale Minervini, Edoardo Ponti, and Antonio Vergari. Neurosymbolic diffusion models, 2025. URL `https://arxiv.org/abs/2505.13138`.

Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordoni. Guiding language model reasoning with planning tokens. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=wi9IffRhVM`.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL `https://arxiv.org/abs/2201.11903`.

Ashen Weligalle. Discrete diffusion models for language generation. *arXiv preprint arXiv:2507.07050*, 2025.

Chünhung Wu, Jinliang Lu, Zixuan Ren, Gangqiang Hu, Zhi Wu, Dai Dai, and Hua Wu. Llms are single-threaded reasoners: Demystifying the working mechanism of soft thinking. *arXiv preprint arXiv:2508.03440*, 2025.

Jianxiang Xiang, Zhenhua Liu, Haodong Liu, Yin Bai, Jia Cheng, and Wenliang Chen. Diffusiondialog: A diffusion model for diverse dialog generation with latent space. *arXiv preprint arXiv:2404.06760*, 2024.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation, 2024. URL `https://arxiv.org/abs/2409.11340`.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning, 2025. URL `https://arxiv.org/abs/2502.14768`.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. Self-evaluation guided beam search for reasoning, 2023. URL `https://arxiv.org/abs/2305.00633`.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL `https://arxiv.org/abs/2305.10601`.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*, 2024a.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. Diffusion of thoughts: Chain-of-thought reasoning in diffusion language models, 2024b. URL `https://arxiv.org/abs/2402.07754`.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning, 2025a. URL `https://arxiv.org/abs/2410.14157`.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025b.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Quanquan Gu. Diffusion language models can perform many tasks with scaling and instruction-finetuning, 2025c. URL `https://arxiv.org/abs/2308.12219`.

Fangxu Yu, Lai Jiang, Haoqiang Kang, Shibo Hao, and Lianhui Qin. Flow of reasoning: Efficient training of llm policy with divergent thinking. *arXiv preprint arXiv:2406.05673*, 1(2):6, 2024.

Fangxu Yu, Lai Jiang, Haoqiang Kang, Shibo Hao, and Lianhui Qin. Flow of reasoning: Training llms for divergent reasoning with minimal examples, 2025a. URL `https://arxiv.org/abs/2406.05673`.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Qifan Yu, Zhenyu He, Sijie Li, Xun Zhou, Jun Zhang, Jingjing Xu, and Di He. Enhancing auto-regressive chain-of-thought through loop-aligned reasoning, 2025b. URL `https://arxiv.org/abs/2502.08482`.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025c.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Eric Zelikman, Georges Raif Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah Goodman. Quiet-STar: Language models can teach themselves to think before speaking. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=oRXPiSOGH9`.

Yiming Zeng, Jinghan Cao, Zexin Li, Yiming Chen, Tao Ren, Dawei Xiang, Xidong Wu, Shangqian Gao, and Tingting Yu. Treediff: Ast-guided code generation with diffusion llms. *arXiv preprint arXiv:2508.01473*, 2025.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=8rcFOqEud5.

Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Joshua Susskind, and Navdeep Jaitly. Planner: Generating diversified paragraph via latent language diffusion model. *Advances in Neural Information Processing Systems*, 36:80178–80190, 2023.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, Shuohang Wang, Yelong Shen, and Xin Eric Wang. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *arXiv preprint arXiv:2505.15778*, 2025.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation, 2024. URL https://arxiv.org/abs/2302.05737.

Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model, 2024a. URL https://arxiv.org/abs/2408.11039.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023. URL https://arxiv.org/abs/2205.10625.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, Shengen Yan, Guohao Dai, Xiao-Ping Zhang, Yuhan Dong, and Yu Wang. A survey on efficient inference for large language models, 2024b. URL https://arxiv.org/abs/2404.14294.

Hanlin Zhu, Shibo Hao, Zhiting Hu, Jiantao Jiao, Stuart Russell, and Yuandong Tian. Reasoning by superposition: A theoretical perspective on chain of continuous thought. *arXiv preprint arXiv:2505.12514*, 2025.

A ADDITIONAL RELATED WORKS

Diffusion Language Models for Text Reasoning Masked diffusion language models attempt to address some common limitations of autoregressive LLMs—such as rigid left-to-right decoding and inefficiency—by iteratively denoising masked tokens, enabling parallel and order-agnostic text generation. Prior studies show that these models achieve better inference efficiency compared to AR models while maintaining comparable performance on both general tasks (Zheng et al., 2024; Gong et al., 2025; Nie et al., 2025; Shi et al., 2025a; Song et al., 2025; Ye et al., 2025c;b) and reasoning benchmarks with chain-of-thought (Gao et al., 2024; Ye et al., 2024b; van Krieken et al., 2025; Ye et al., 2025a). However, these approaches remain constrained to language space, unable to capture reasoning at an abstract semantic level or revise previously generated tokens as continuous diffusion models (Ho et al., 2020; Song et al., 2020) do, and they require training on massive datasets rather than leveraging a well-trained LLM. In contrast, our method overcomes these limitations by structuring reasoning in an interpretable continuous latent space, producing abstract CoT representations with self-correction ability for an existing LLM, while keeping diffusion's strengths in parallel generation to enhance exploration and diversity.

CoT Reasoning Chain-of-thought reasoning refers to methods which elicit LLMs to generate intermediate reasoning steps in language prior to outputting a final answer in order to improve performance on reasoning tasks. This can be accomplished via prompting methods (Nye et al., 2021; Wei et al., 2023; Khot et al., 2023; Zhou et al., 2023) or through training LLMs (by SFT, RL, or a combination of the two) to output the intermediate reasoning steps (Yu et al., 2023; Shao et al., 2024; Yu et al., 2025a). Works have also extended CoT to allow LLMs to mimic various tree search algorithms such as BFS or MCTS, which especially improves performance on more complex tasks (Xie et al., 2023; Yao et al., 2023; Zhang et al., 2024; Bi et al., 2025). Beyond following specific algorithms, works that implement long chain-of-thought (combining extensive reasoning, exploration, and reflection) have also demonstrated improved reasoning performance (Shinn et al., 2023; Gandhi et al., 2025; Saha et al., 2025; Xie et al., 2025). One overarching limiting factor with these CoT methods is that they fundamentally work at a next-token-prediction level, constraining the outputs to the token space and limiting the model's horizon.

Hybrid AR+Diffusion Model Architecture Other AR-Diffusion hybrid models have shown successful results in rivaling their AR and diffusion counterparts, particularly in multimodal generation and image understanding. The Transfusion (Zhou et al., 2024a) architecture demonstrated that hybrid models could outperform standard AR models and compete with state-of-the-art diffusion models in image-generation benchmarks, a phenomenon further reinforced by other studies of hybrid models (Fan et al., 2024; Tang et al., 2024a; Xiao et al., 2024). This extends beyond image generation, with several works demonstrating the effectiveness of hybrid AR-diffusion models in other domains such as image understanding, video generation, and robot control (Black et al., 2024; Tong et al., 2024a; Chen et al., 2025a;b). Furthermore–similar to our model architecture–works have demonstrated successful adaptations of frozen models for these hybrid AR-diffusion archictures in multimodal domains (Pan et al., 2025; Shi et al., 2025b). Aside from the difference in domain from these works, many do not use block diffusion for variable-length generations as in LaDiR and we critically introduce CE loss to guide better latent predictions.

B ADDITIONAL PRELIMINARIES AND BACKGROUND

We provide more details about the background information of VAE and Diffusion models in this section.

B.1 VARIATIONAL AUTOENCODER AND β-VAE

The Variational Autoencoder (VAE) (Kingma & Welling, 2013) is a latent-variable model that learns a compressed representation of data x through an encoder–decoder pair. The encoder $q_\phi(z|x)$ maps inputs into a distribution over latent variables z, typically parameterized as a diagonal Gaussian. The decoder $p_\theta(x|z)$ reconstructs the input from z, enabling generative sampling. Training maximizes the evidence lower bound (ELBO):

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] + \mathrm{KL}\big(q_\phi(z|x) \,\|\, p(z)\big), \tag{7}$$

where the first term ensures faithful reconstruction and the second term regularizes the posterior toward a simple prior $p(z)$, usually $\mathcal{N}(0, I)$. The reparameterization trick,

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

enables low-variance gradient estimates for stochastic optimization.

β-VAE. The β-VAE (Higgins et al., 2017) introduces a hyperparameter β to control the KL weight:

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{q_\phi(z|x)}[-\log p_\theta(x|z)] + \beta\, \mathrm{KL}\big(q_\phi(z|x) \,\|\, p(z)\big). \tag{8}$$

When $\beta > 1$, the model enforces stronger alignment to the prior, which encourages disentangled and interpretable latent variables at the expense of reconstruction fidelity. This property is desirable when latent codes are later used as the substrate for generative modeling.

Why VAE for Latent Diffusion. Latent diffusion models (LDMs) (Rombach et al., 2022) operate not on raw high-dimensional inputs (e.g., images or sequences), but in a compressed latent space learned by a VAE. This design provides three key advantages:

1. **Efficiency.** Operating in latent space reduces dimensionality, leading to faster training and inference while maintaining semantic richness.
2. **Semantic abstraction.** The VAE learns to discard imperceptible details and retain high-level structure, making diffusion steps focus on meaningful features rather than pixel-level noise.
3. **Flexibility.** The decoder $p_\theta(x|z)$ ensures that even when denoising occurs in latent space, the final output remains in the original input domain. This separation enables diffusion to generalize across modalities with a shared latent backbone.

B.2 Latent Diffusion: Training and Inference

Latent diffusion operates in the compressed latent space z_0 of a pretrained VAE.

Forward process. Noise is added gradually:

$$q(z_t|z_0) = \mathcal{N}\big(z_t; \sqrt{\bar{\alpha}_t}\, z_0, (1-\bar{\alpha}_t)I\big),$$

with $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$.

Training objective. The denoiser $\epsilon_\theta(z_t, t)$ predicts the injected noise:

$$\mathcal{L}_{\text{LDM}} = \mathbb{E}_{z_0,\epsilon,t}\big[\|\epsilon - \epsilon_\theta(z_t,t)\|^2\big].$$

Inference. Generation starts from $z_T \sim \mathcal{N}(0, I)$ and denoises iteratively:

$$z_{t-1} = \frac{1}{\sqrt{\alpha_t}}\Big(z_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(z_t,t)\Big) + \sigma_t \epsilon.$$

B.3 Comparison of Parameterizations

Diffusion training can be expressed through different target parameterizations, all of which can be interpreted as variants of the same continuous-time flow. Below we summarize the most common forms:

B.3.1 ϵ-Prediction

The denoiser directly predicts the added Gaussian noise ϵ:

$$\mathcal{L}_\epsilon = \mathbb{E}_{z_0,\epsilon,t}\Big[\|\epsilon - \epsilon_\theta(z_t,t)\|^2\Big], \tag{9}$$

where $z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$. This is the standard DDPM formulation (Ho et al., 2020). It is stable but sometimes less efficient for long horizons.

B.3.2 x_0-Prediction (DDIM-x_0)

Instead of noise, the model predicts the clean latent z_0:

$$\mathcal{L}_{x_0} = \mathbb{E}_{z_0,t}\Big[\|z_0 - x_{0,\theta}(z_t,t)\|^2\Big]. \tag{10}$$

This corresponds to the DDIM formulation (Song et al., 2020), enabling deterministic sampling and fewer inference steps, but can overfit to data scale.

B.3.3 v-Prediction

Proposed by Salimans & Ho (2022), v is defined as a linear combination of noise and clean latent:

$$v = \sqrt{\bar{\alpha}_t}\,\epsilon - \sqrt{1-\bar{\alpha}_t}\,z_0, \tag{11}$$

with objective

$$\mathcal{L}_v = \mathbb{E}_{z_0,\epsilon,t}\Big[\big\|v - v_\theta(z_t,t)\big\|^2\Big]. \tag{12}$$

v-prediction is numerically better conditioned, often improving stability across timesteps.

All four parameterizations can be viewed as different instantiations of the same underlying generative flow. ϵ-prediction, x_0-prediction, and v-prediction specify *which quantity* the denoiser regresses on. Flow matching directly learns the continuous velocity field, avoiding discretization artifacts.

B.4 Block Diffusion

Suppose we are using ϵ-prediction, and let a sequence be segmented into M blocks $\{B_1,\ldots,B_M\}$, where $B_m \in \mathbb{R}^{k\times d}$ contains k latent tokens of dimension d. The forward noising process for block B_m is

$$q(B_{m,t} \mid B_{m,0}) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\,B_{m,0}, (1-\bar{\alpha}_t)I\big), \tag{13}$$

and the denoiser f_θ is trained to predict the noise at the block level:

$$\mathcal{L}_{\text{block}} = \mathbb{E}_{m,t,\epsilon}\Big[\big\|\epsilon - f_\theta(B_{m,t},t)\big\|^2\Big]. \tag{14}$$

Blocks are generated autoregressively, i.e.,

$$p(B_m \mid B_{<m}) = \int q(B_{m,0} \mid x) \prod_t p_\theta(B_{m,t-1} \mid B_{m,t}, B_{<m})\, dB_{m,0}, \tag{15}$$

so that each block is denoised iteratively while conditioning on all previously generated blocks.

C Additional Ablation Studies

Latent Prediction Objective To assess the impact of different training objectives for latent prediction, we compare several widely used formulations. The first baseline is *MSE loss*, which directly minimizes the mean-squared error between predicted and ground-truth latents but yields the weakest results. We then adopt three DDIM-based (Song et al., 2020) objectives: predicting the clean latent state (x_0), the added noise vector (ϵ), and the velocity (v). These diffusion objectives consistently improve accuracy, highlighting the advantage of explicitly modeling the denoising process rather than relying on direct predictions. Moreover, the flow matching (u) objective achieves the strongest gains, suggesting that learning the latent vector field is more effective for capturing the pattern of reasoning compared to DDIM's denoising objectives.

Objective	CD-4 Pass@1 (%)
MSE Loss	46.0
x_0	53.0
ϵ	58.0
v	62.0
u **(ours)**	**73.5**

Table 5: Ablation study on latent prediction objectives on the Countdown-4 dataset.

Effect of the Block Size. We investigate how the number of latent tokens per block (L_b) influences reconstruction quality and downstream reasoning accuracy on GSM8K. As shown in Figure 6, too few tokens (i.e., 1 token) limit the model's ability to capture necessary information, harming reconstruction. Performance improves as the number of tokens increases, reaching near-perfect reconstruction at $n = 6$. Beyond this point, however, adding more tokens introduces redundancy, which makes the latent diffusion model harder to predict accurately and leads to diminished reasoning accuracy. This reveals a trade-off between compact latent representations and effective downstream reasoning.

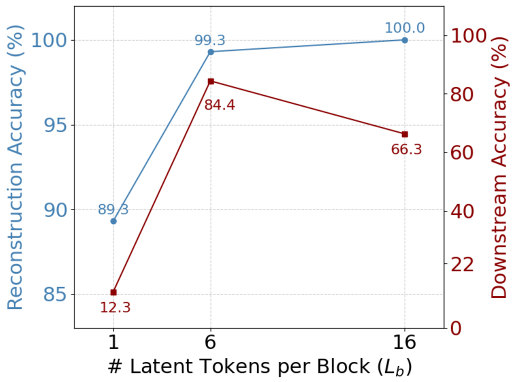


Figure 6: Ablation analysis of block size on the GSM8K benchmark.

VAE Robustness Augmentations. When training our VAE, to improve the robustness of the VAE latent space, we apply two augmentation strategies during VAE training: (1) adding Gaussian noise to latent representations with standard deviation k, and (2) randomly substituting input tokens with probability p. Table 6 shows the impact of these augmentations on GSM8K accuracy. Best performance is achieved at $k = 3$ and $p = 0.3$. Too little augmentation ($k = 0$ or $p = 0$) results in overfitting to clean inputs, while excessive augmentation ($k = 5$ or $p > 0.5$) degrades the latent space quality.

Latent Gaussian Noise (p=0.3)		**Token Substitution (k=3)**	
k (std)	GSM8K Acc (%)	p (prob.)	GSM8K Acc (%)
0	68.3	0.0	70.2
1	73.4	0.1	78.3
3	**84.2**	**0.3**	**84.2**
5	79.4	0.5	64.0
–	–	0.7	32.4

Table 6: Ablation study on VAE robustness augmentations on GSM8K.

Blockization Strategy We investigate the sensitivity of the model to different blockization strategies by varying the number of sentences per block. Table 7 shows results for 1, 2, and 3 sentences per block. Using more sentences per block requires more latent tokens to maintain reconstruction quality and significantly increases difficulty for the diffusion model. We find that **1 sentence per block** with 4 latent tokens offers the best balance between latent compactness and reasoning accuracy.

# Sentences	**# Latent Tokens**	**GSM8K**	**MATH**
1	4	**84.2**	**45.2**
2	8	78.4	39.6
3	12	72.0	36.1

Table 7: Ablation study on blockization strategy (sentences per block).

Method	MATH	GSM8K	Gaokao	DM	College	Olympia	TheoremQA	Avg.
				Diffusion Language Models — LLaDA 8B				
Base Model	–	–	–	–	–	–	–	–
CoT SFT	59.9	92.4	43.0	50.7	58.3	10.2	26.3	48.7
				Autoregressive Models — LLaMA 3.1 8B				
Sol-Only SFT	–	–	–	–	–	–	–	–
CoT SFT	49.8	89.0	37.9	52.0	54.9	12.9	25.0	45.9
iCoT	–	–	–	–	–	–	–	–
Pause Token	–	–	–	–	–	–	–	–
Coconut	39.3	74.3	29.3	36.9	42.9	6.3	14.9	34.8
Discrete Latent	47.3	88.6	39.7	49.5	53.7	17.8	28.5	46.4
LaDiR	**63.7**	**93.7**	**45.8**	**54.2**	**60.3**	**15.3**	**30.7**	**52.0**

Table 8: Pass@100 accuracy across in-domain and out-of-domain math benchmarks.

D Additional Results and Analysis

Pass@k Results on Math Reasoning

Intepretability In addition to achieving superior or competitive accuracy across each benchmark, as shown in Table 9, LaDiR also benefits from being more interpretable by nature compared to standard diffusion-based methods.

Block	Text
Question	*Billy sells DVDs. He has 8 customers on Tuesday. The first 3 buy one DVD each, the next 2 buy two DVDs each, the last 3 buy none. How many DVDs did Billy sell?*
Decode($Z^{(1)}$):	Billy's first 3 customers buy one DVD each, so that's $3 * 1 = \langle\langle 3 * 1 = 3\rangle\rangle$3 DVDs.
Decode($Z^{(2)}$):	His next 2 customers buy 2 DVDs each, so that's 2 × 2 = 4 DVDs.
Decode($Z^{(3)}$):	His last 3 customers don't buy any DVDs, so that's 0 DVDs sold.
Decode($Z^{(4)}$):	Therefore, Billy sold a total of 3 + 4 + 0 = 7 DVDs on Tuesday.
Answer	The answer is: 7.

Table 9: Example of interpretable continuous thought tokens: each latent block Z^i is able to be decoded to human-readable text through the VAE decoder. Each latent block is decoded individually, so the entire latent thought is represented by the block in isolation. This allows for clear interpretibility of each latent thought, while still allowing for a model to reason in a latent space.

Reasoning at Semantic Level Table 10 demonstrates that LaDiR refines its reasoning through semantic information rather than lexical connections. The pink segments trace how the model incrementally refines its intermediate reasoning steps—transforming rough or incomplete arithmetic expressions into logically consistent equations. For example, the model first produces an approximate relation such as $2 = 1 + 1$ and later reconstructs the correct multiplicative form $2 * 1 = 2$, eventually combining these into a complete causal chain that leads to $2 * 4 = 8$. Meanwhile, the green segments serve as stable anchors capturing the essential reasoning cues—phrases like "two times older than" and "four times as old"—which define the causal dependencies between Suzy, Ben, and Brandon. In contrast, the yellow segments correspond to non-reasoning or noisy parts (e.g., "Alice," "Instagram") that the model learns to "ignore" during early denoising, prioritizing semantically coherent and logical grounded reasoning over irrelevant lexical content. In contrast, the SFT baseline remains syntactically fluent but semantically brittle—producing text that appears correct on the surface yet fails to capture the true causal logic. These observations indicate that LaDiR's denoising process operates on the level of semantics and relational consistency, enabling reasoning that aligns with human-like conceptual understanding.

Consistency Across Long Reasoning Blocks As shown in Table 11, LaDiR maintains coherent and semantically consistent reasoning across a long sequence of blocks. Each block builds logically on the previous one—starting from variable definition (x *as silver coins*), to constraint formulation ($x + (x + 30) = 110$), simplification, and finally the correct numerical conclusion ($x = 40$, *gold*

Question	Brandon's iPhone is four times as old as Ben's iPhone. Ben's iPhone is two times older than Suzy's iPhone. If Suzy's iPhone is 1 year old, how old is Brandon's iPhone?	
SFT (Baseline)	Suzy's iPhone is 1 year old. Ben's iPhone would be $1 + 1 = 2$ years old. Since Brandon's iPhone is four times as old as Ben's, we compute $4 \times 1 = 4$. Therefore, Brandon's iPhone is 4 years old. The answer is: 4.	
LaDiR (Ours)	**Block 1**	**Block 2**
Decode($\hat{Z}_{t=1.0}$)	`10 and 10 10....`	`[garbled tokens]    &^* _`
Decode($\hat{Z}_{t=0.9}$)	If Alice is 1 year old, then Bob's age is 2 years older than Alice, which means Bob is 2 = 1 + 1 years old.	://@natechandra's Instagram is four times as old as Ben's Instagram, so Ben's Instagram is 1 * 4 = 4 years old.
Decode($\hat{Z}_{t=0.8}$)	If Suzy's age is 1 year old, then Ben's age is two times older than Suzy's age, which is 2 * 1 = 2 years old.	If Brandon's phone is four times as old as Ben's phone, then Brandon's phone is 2 * 4 = 8 years old. 1nbsp;nbsp;nbsp...
Decode($\hat{Z}_{t\leq 0.7}$)	If Suzy's iPhone is 1 year old, then Ben's iPhone is two times older than Suzy's iPhone, so Ben's iPhone is 2 * 1 = 2 years old.	If Brandon's iPhone is four times as old as Ben's iPhone, then Brandon's iPhone is 2 * 4 = 8 years old.
Answer	The answer is: 8.	

Table 10: An example of self-refinement during inference on the GSM8k dataset, showing how reasoning becomes progressively clearer as t decreases. Later denoising steps correct arithmetic errors while maintaining earlier structure, demonstrating semantic self-refinement. Pink segments highlight refined reasoning portions, yellow segments indicate non-reasoning or noisy parts that the model gradually corrects, and green segments denote key reasoning cues essential for correct logic.

$= 70$). Unlike the SFT baseline, which produces a single-step approximation that conflates intermediate relations, LaDiR preserves arithmetic and causal consistency throughout the reasoning chain, demonstrating stable multi-step inference even with a large number of reasoning blocks.

Question	Gretchen has 110 coins. There are 30 more gold coins than silver coins. How many gold coins does Gretchen have?
SFT (Baseline)	Let's assume Gretchen has 110 coins in total and 30 more gold than silver. Half of the coins plus 30 should be gold, so $110/2 + 30 = 85$ gold coins.
LaDiR (Ours)	
Decode($Z^{(1)}$)	Let's assume the number of silver coins Gretchen has is x silver coins.
Decode($Z^{(2)}$)	We also know that there are 30 more gold coins than silver coins, so the number of silver coins is $x + 30$ gold coins.
Decode($Z^{(3)}$)	The total number of coins Gretchen has is $x + (x + 30) = 110$.
Decode($Z^{(4)}$)	Combining like terms, we get $2x + 30 = 110$.
Decode($Z^{(5)}$)	Subtracting 30 from both sides, we get $2x = 80$.
Decode($Z^{(6)}$)	Dividing both sides by 2, we get $x = 40$.
Decode($Z^{(7)}$)	Therefore, Gretchen has $30 + 40 = 70$ gold coins.
Answer	The answer is: 70.

Table 11: An qualitative example in GSM8K illustrating the long reasoning blocks generated by our method compared to the baseline SFT.

E Experimental Details

E.1 Math Reasoning

Implementation Details The CoT data is segmented into thought-level blocks, where each block corresponds to a single sentence and is represented by 6 latent thought tokens. For VAE training, we set the to $\beta = 10^{-5}$, in which the encoder is finetuned from the backbone model while the decoder remains frozen. The flow-matching model is trained with the objective in Eq. 5, using $\lambda_{\text{FM}} = 5, \lambda_{\text{Ans}} = 1, \lambda_{\text{Spec}} = 2$. During inference, we initialize the Gaussian noise of scale 2, and apply diversity guidance with a maximum scale of 0.8.

Datasets We train on only the DART-MATH dataset, holding out all other benchmarks for evaluation only. Table 12 summarizes the datasets. While training is limited to mathematical reasoning problems, our evaluations also include out-of-domain tasks such as engineering and physics, providing both in-domain and out-of-domain benchmarks to assess the reasoning and generalization capabilities of LaDiR and the baselines.

Dataset	# Samples	Domain / Level	Notes
DART-MATH	585k	Mixed math (train)	Synthesized for reasoning, based on GSM8K/MATH
MATH	500	High school / competition	In-domain benchmark
GSM8K	1.3k	Grade school arithmetic	In-domain benchmark
College-Math	2.8k	College-level	Linear algebra, differential equations, etc.
DM-Math	1k	K–12 curriculum	Out-of-domain generalization
OlympiaBench-Math	675	Olympiad-level	Advanced competition problems
TheoremQA	800	STEM / theorem-driven	Math, physics, engineering
Fresh-Gaokao-Math-2023	30	Gaokao exam	Real-world test distribution

Table 12: Summary of datasets used in our experiments. We use DART-MATH (Tong et al., 2024b), MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), College-Math (Tang et al., 2024b), DeepMind-Math (Saxton et al., 2019), OlympiaBench-Math (He et al., 2024), TheoremQA (Chen et al., 2023), and Fresh-Gaokao-Math-2023 (Tang et al., 2024b).

E.2 Hyperparameters

Table 13 provides a complete summary of all hyperparameters used in our experiments, covering VAE pretraining, Stage-1 teacher-forcing training, Stage-2 rollout training, and inference settings.

Component	Hyperparameter	Value
VAE Pretraining	Latent dimension (d_z)	512
	# latent tokens per block	4
	KL weight β	1×10^{-5}
	Learning rate	2×10^{-5}
	Batch size	128
	# of Epochs	2
Stage-1 Teacher-Forcing	Flow-matching loss weight (λ_{FM})	5
	CE loss weight (λ_{Ans})	1
	Special-token loss weight (λ_{Spec})	1
	Learning rate	1×10^{-5}
	Batch size	64
	# of Epochs	20
Stage-2 Rollout Training	Learning rate	1×10^{-5}
	Batch size	12
	# of Epochs	20
Inference	Classifier-free guidance scale	4
	Answer token decoding temperature	0.7

Table 13: Complete hyperparameter settings for all training and inference stages.

F Additional Model Details

VAE Training Architecture Figure 7 is a more in-depth diagram of the training of the VAE. The encoder LLM first maps the input sequence of original text embeddings and learnable embeddings into hidden states. These hidden states are then projected through two linear layers to produce the mean and variance of the latent distribution, from which we sample the thought tokens via the reparameterization trick. The sampled latent tokens $\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_k$ are passed to the decoder LLM, which reconstructs the original text tokens under a teacher-forcing setup. This design enables the model to compress high-dimensional text into a smaller set of semantically meaningful latent variables, while still maintaining faithful reconstruction of the original reasoning process.

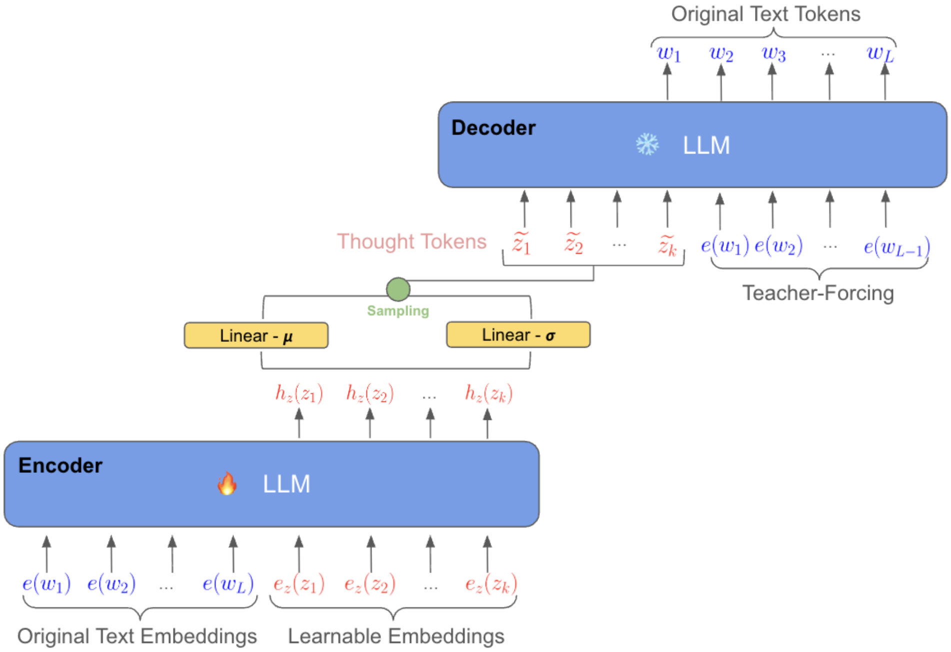

Original Text Tokens
w_1 w_2 w_3 ... w_L
Decoder
LLM
Thought Tokens
$\tilde{z}_1$ $\tilde{z}_2$ ... $\tilde{z}_k$
$e(w_1)$ $e(w_2)$ ... $e(w_{L-1})$
Teacher-Forcing
Sampling
Linear - μ
Linear - σ
$h_z(z_1)$ $h_z(z_2)$ ... $h_z(z_k)$
Encoder
LLM
$e(w_1)$ $e(w_2)$ ... $e(w_L)$
$e_z(z_1)$ $e_z(z_2)$ ... $e_z(z_k)$
Original Text Embeddings
Learnable Embeddings

Figure 7: Detailed architecture of the variational autoencoder for latent reasoning. The encoder is a finetuned LLM that takes both original text embeddings $e(w_i)$ and learnable embeddings $e_z(z_i)$, producing mean and variance vectors through linear projections of the last hidden state h. Latent thought tokens $\tilde{z}_i$ are then sampled from $\mathcal{N}(\mu, \sigma^2)$. The decoder is a frozen LLM that reconstructs the original CoT text under teacher forcing, conditioned on both the sampled thought tokens and the original text embeddings.

G Use of Large Language Models (LLMs)

Throughout this project, Large Language Models were utilized as coding tools and as grammar checkers to support the writing of the paper. They did **not** play a significant role in research ideation or writing to the extent of being listed as a contributor. LLMs were used strictly as a general purpose tool.