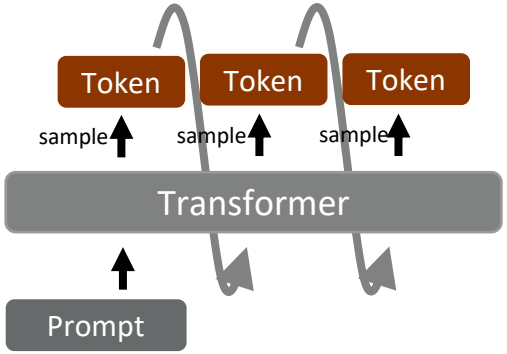
Coevolutionary Continuous Discrete Diffusion: Make Your Diffusion Language Model a Latent Reasoner

Cai Zhou, *MIT CSAIL*
caiz428@mit.edu
https://homepage.zhouc.ai

MIT CSAIL Computer Science & Artificial Intelligence Laboratory

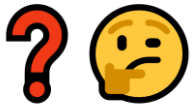
Motivations: Rethinking Current Paradigms of LLM Reasoning



Auto-Regressive

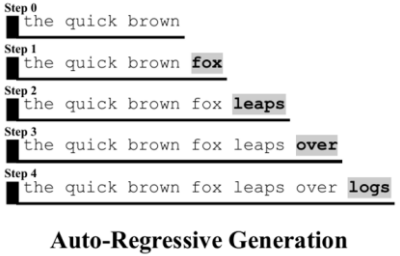
Discrete

- Not good at explicit searching, planning, and self-correction over iterative forward passes.
- External tools or workflows are utilized to tackle this, such as CoT, test-time scaling, agent, etc.
- Can we enable LLMs to have these capabilities built-in internally, using more efficient and expressive way than CoT?



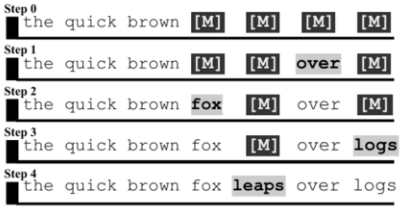
Background

- Autoregressive LLM
  - Empirically successful
  - Fundamental limitations in some hard / structured tasks



Auto-Regressive Generation

- Diffusion language model
  - Mostly discrete diffusion (mainly masked diffusion)
  - Capable of any-order generation



Any-Order Generation

Why Diffusion Language Models?

- Bidirectional attention?
- Any-order generation?
- Self-correction?
- Multi-token prediction?
- …

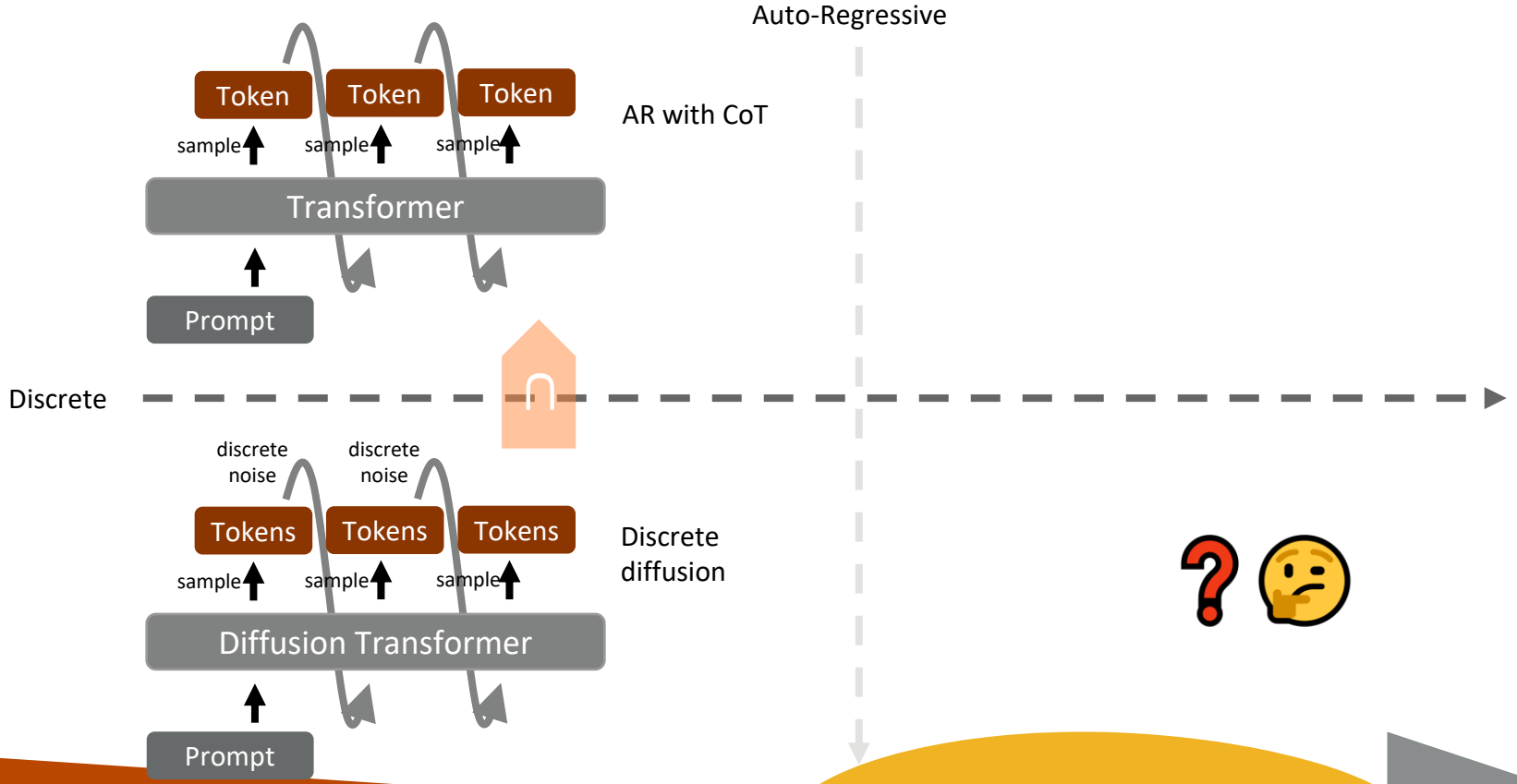
In this talk: expressivity (function class and computation complexity theory)

Why Not Masked Discrete Diffusion Models?

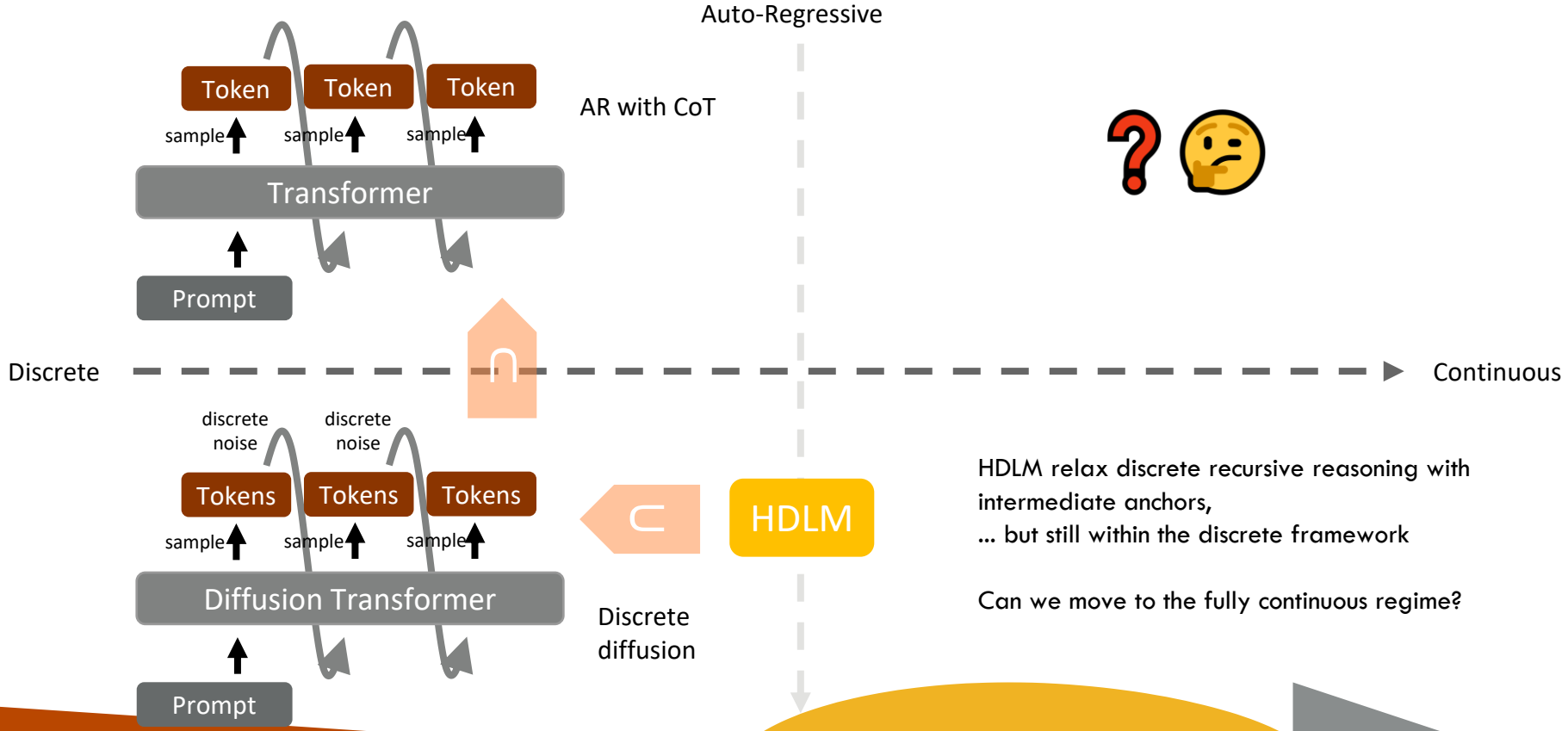
- Limited expressivity due to finite discrete states
- Underusing historical information
  - [M] not unmasked: no token-level information gain
  - [M] unmasked: cannot be corrected
- “Randomized BERT” (SDE only)

In this talk: new diffusion modeling paradigms (hierarchical and continuous)

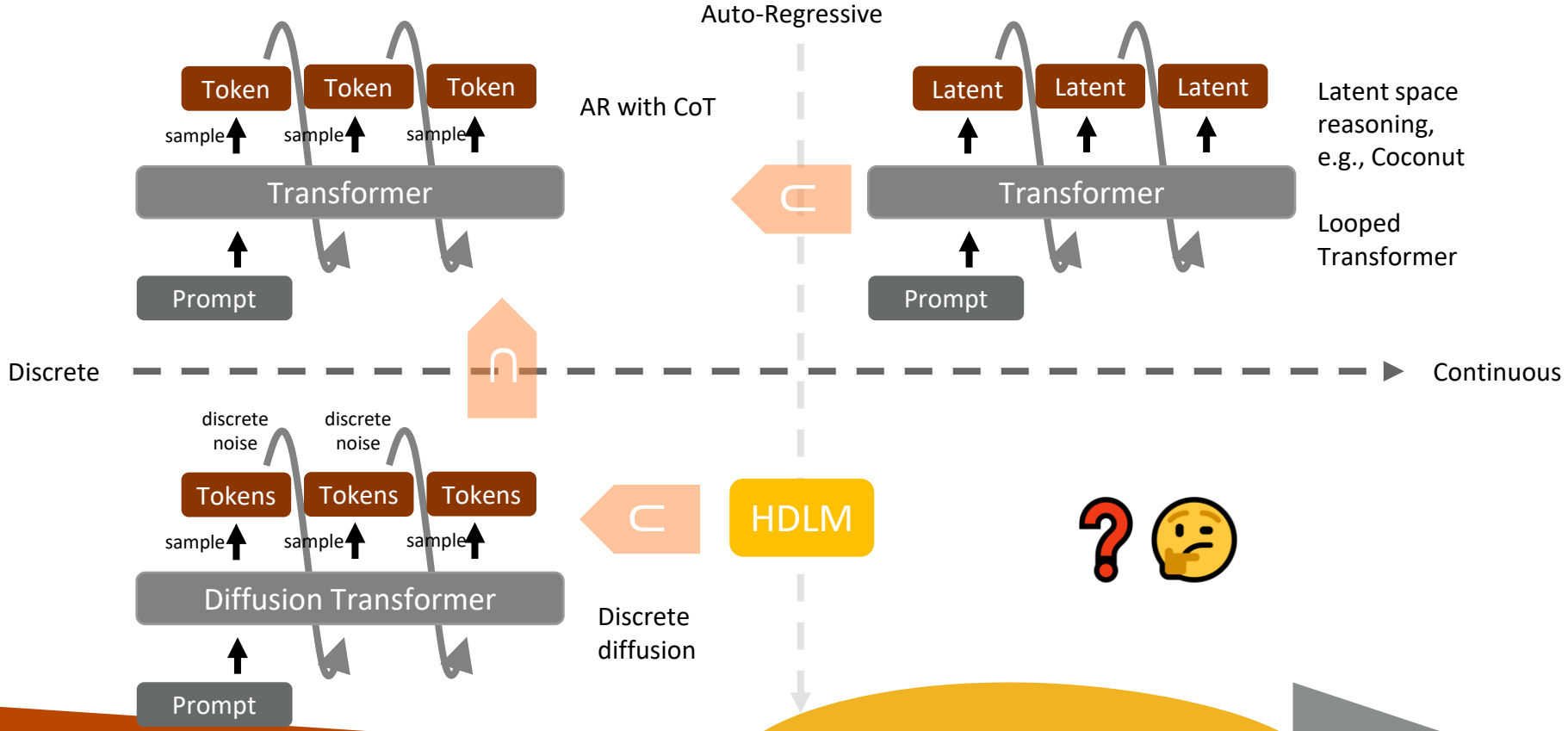
Rethinking Current Paradigms of LLM Reasoning



Rethinking Current Paradigms of LLM Reasoning



Rethinking Current Paradigms of LLM Reasoning



Coevolutionary Continuous Discrete Diffusion: Make Your Diffusion Language Model a Latent Reasoner

Cai Zhou, Chenxiao Yang, Yi Hu, Chenyu Wang, Chubin Zhang,

Muhan Zhang, Lester Mackey, Tommi Jaakkola, Stephen Bates, Dinghuai Zhang

MIT & Microsoft & TTIC & PKU & THU

ICML 2026

https://arxiv.org/abs/2510.03206

Preliminary

- CTMC and discrete diffusion

$$\dot{q}_t = G_t\, q_t, \quad P_{s\to t} = \mathcal{T}\exp\Big(\int_s^t G_\tau\, d\tau\Big), \quad q_t = P_{0\to t} q_0.$$

$$q_t(x_t|x_0) = \mathrm{Cat}(\eta_t x_0 + (1-\eta_t)\pi_t)$$

- ODE/SDE and continuous diffusion

$$dz_t \;=\; a_t(z_t)\, dt \;+\; g_t\, dW_t$$

$$z_t \;=\; \alpha_t\, z_0 + \sigma_t\, \epsilon_t, \;\; \epsilon_t \sim \mathcal{N}(0, I), \;\; \alpha_t = \exp\Big(-\tfrac{1}{2}\int_0^t \beta_\tau\, d\tau\Big), \;\; \sigma_t = \sqrt{1-\alpha_t^2}.$$

- Looped transformers:

$$h_{k+1} \;=\; \Phi_\theta\big(h_k\big), \qquad k = 0, \ldots, T-1.$$

CCDD: Theoretical Expressivity

- For fair comparison:
  - Same architectures and parameter counts
  - Up to slight difference in the first encoding layer and the last decoding layer
- Measure the computation complexity required to solve certain problem
  - Transformers with CoT lie in TC0, looped transformers lie in TC1
  - Complexity for graph reachability problems or formal language recognition with size n: $O(n)$ for CoT, $log(n)$ for LT [1]
  - Scaling the depth (LT) is more efficient than scaling the context length (CoT)

[1] William Merrill and Ashish Sabharwal. A little depth goes a long way: The expressive power of log-depth transformers. arXiv preprint arXiv:2503.03961, 2025.

CCDD: Theoretical Expressivity

- ★ CDM > DDM

Definition 1 (Trajectory families and embedded discrete family)**.** *Define the trajectory family of continuous diffusion* $\mathsf{F}_{\text{cont}}(\theta)$*, discrete diffusion* $F_{\text{disc}}(\theta)$*, and the* embedded discrete family $\widetilde{\mathsf{F}}_{\text{disc}}(\theta) \subset \mathcal{P}(\mathcal{Z})$ *(the pushforward by the fixed encoder* $\mathcal{E}$*) as follows:*

$$\mathsf{F}_{\text{cont}}(\theta) := \left\{\{p_t(z_t)\}_{t\in[0,1]}\right\},\ \mathsf{F}_{\text{disc}}(\theta) := \left\{\{p_t(x_t)\}_{t\in[0,1]}\right\},\ \widetilde{\mathsf{F}}_{\text{disc}}(\theta) := \mathcal{E}_\sharp \mathsf{F}_{\text{disc}}(\theta) \tag{11}$$

Theorem 1 (Strict trajectory-level gap)**.** *At any fixed* $t \in [0,1]$*, we have the following strict inclusion*

$$\widetilde{\mathsf{F}}_{\text{disc}}(\theta) \subsetneq \mathsf{F}_{\text{cont}}(\theta) \subseteq \mathcal{P}(\mathcal{Z}) \tag{12}$$

Proof Sketch: DDM is always finitely supported and discards full logits in sampling

- ★ CDM > LT

CCDD: Theoretical Expressivity

- ★ CDM > DDM
- ★ CDM > LT

Proposition 2 (Continuous diffusion sampler can simulate looped rollouts)**.** *Fix any looped transformer Φ_θ and any roll out times $T \in \mathbb{N}$, there exists a continuous diffusion sampler for the reverse PF–ODE by the explicit Euler method with step size $1/T$ that exactly reproduces the looped rollout.*

Proof Sketch: Construct the ODE Euler steps as LT iterations.

CDM is more expressive thanks to timestep embeddings, stochastic injected noises, and flexible sampling procedures (test-time scaling).

CCDD: Practical Trainability

- ★ CDM > LT

  Remark: CDM provides intermediate supervision on the SDE/ODE path;

  LT typically only supervises the final rollout states

- ★ DDM > CDM

Table 1: Comparison between generation space of continuous diffusion.

	Simplex Δ^{V-1}	**Token-wise** $\mathbb{R}^d$	**Contextualized** $\mathbb{R}^d$
Dimensionality	$V-1$ (high)	$d \leq V$ (often $\ll V$)	$d \leq V$ (often $\ll V$)
Geometry	Constrained manifold	Euclidean; codebook cells	Euclidean; contextual manifold
Target smoothness	Low (near vertices)	Atomic, non-smooth	Higher (good embedding models)
Calibration	Natural	Requires decoder	Requires decoder (context)
Expressivity (terminal)	Baseline	$\not\succ$ simplex (Prop. 17)	$\geq$ simplex if decoder strong
Decoding ambiguity	Low	Medium (NN/energy)	High if not sufficient
Optimization	Hard (constraints)	Boundary brittle	Complex but smoother targets

CCDD: Practical Trainability

- ★ CDM > LT
- ★ DDM > CDM
  - Probability simplex: high dimension
  - Token-wise embedding: low-quality representation; less expressive
  - Contextualized embedding: combinatorial decoding complexity

Large decision space makes optimization and sampling harder!

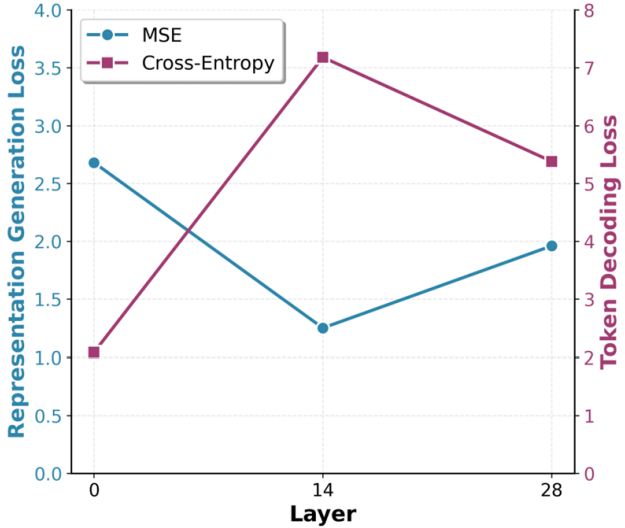
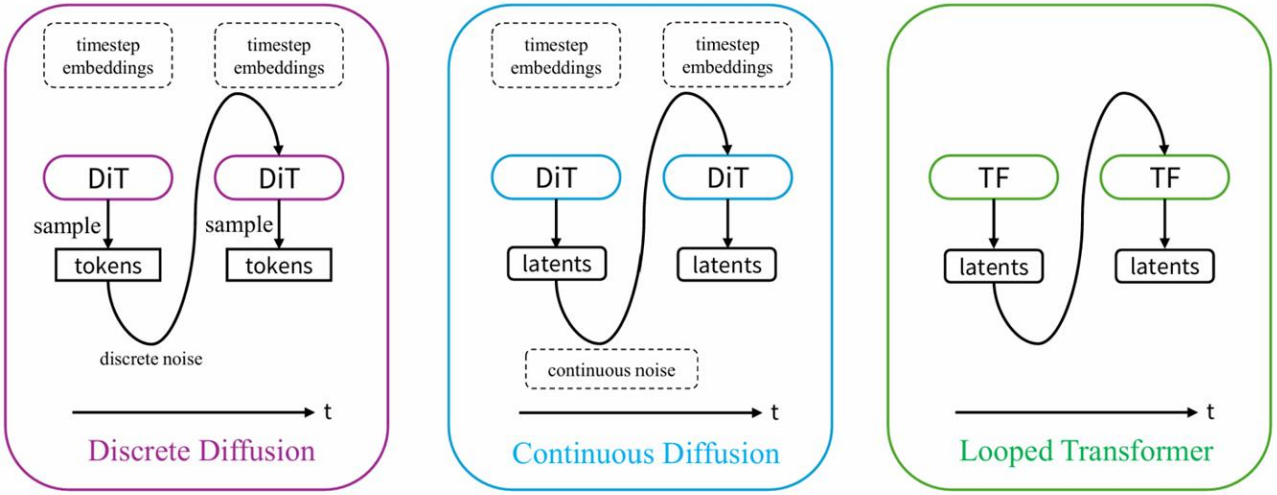


Figure 2: Comparison of validation losses when using representations from different layers of Qwen3-Embedding-0.6B as the latent spaces for CDMs.

Coevolutionary Continuous Discrete Diffusion



Theoretical Expressivity (Section 3.1): CDM $\succcurlyeq$ DDM CDM $\succcurlyeq$ LT

Empirical Trainability (Section 3.2): DDM $\succcurlyeq$ CDM $\succcurlyeq$ LT

Figure 1: Comparison of theoretical expressiveness and practical trainability of: discrete diffusion (left), continuous diffusion with optional continuous noise (middle), and looped transformer (right).

Coevolutionary Continuous Discrete Diffusion

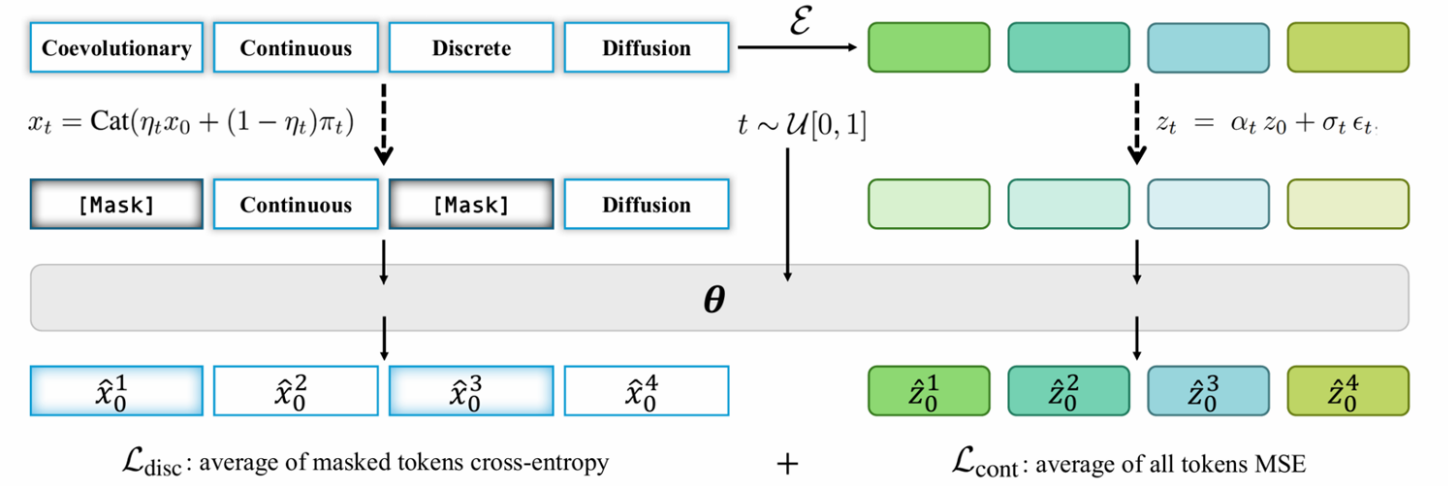


Figure 3: Framework of Coevolutionary Continuous Discrete Diffusion.

Remark: factorization is an efficient and sufficient parameterization that indeed captures the cross-modal dependency!

Coevolutionary Continuous Discrete Diffusion

- Joint diffusion process over the union of the discrete and continuous space

$$(x_t, z_t) \in \mathcal{X} \times \mathcal{Z}$$

- Forward process:

$$q_t(x_t, z_t \mid x_0, z_0) = q_t^{\text{disc}}(x_t \mid x_0)\, q_t^{\text{cont}}(z_t \mid z_0).$$

$$z_t \sim \mathcal{N}(\alpha_t z_0, \sigma_t^2 I), \quad x_t \sim \text{Cat}(\eta_t x_0 + (1-\eta_t)\pi_t).$$

- Reverse process:

$$p_\theta(x_s, z_s \mid x_t, z_t) \;=\; p_\theta^{\text{disc}}(x_s \mid x_t, z_t)\; p_\theta^{\text{cont}}(z_s \mid x_t, z_t).$$

- Training loss:

$$\mathcal{L}_{\text{CCDD}} = \gamma_{\text{cont}} \cdot \mathcal{L}_{\text{cont}} + \gamma_{\text{disc}} \cdot \mathcal{L}_{\text{disc}}$$

CCDD: Architecture

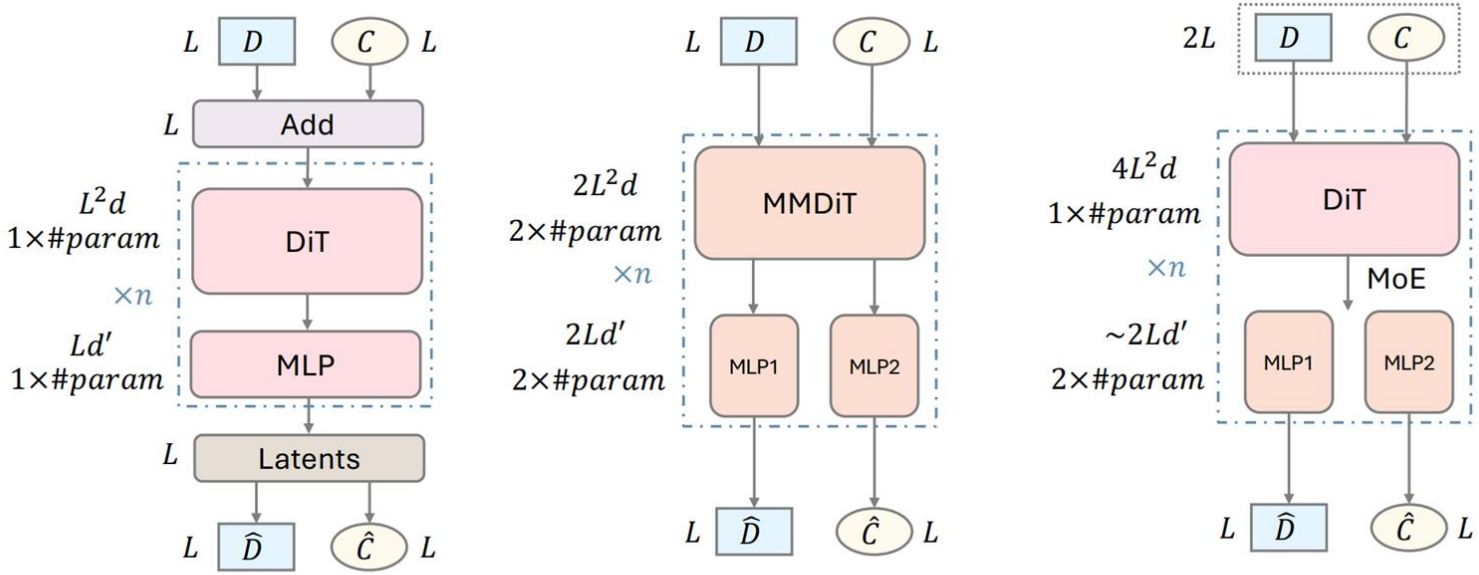


Figure 4: Comparison of different denoising network architectures for CCDD.

CCDD: Implementation

- Selection of representation space: Qwen3-Embedding.
  - Representation alignment for acceleration: REPA [1], REED [2]
  - Representation guidance: REED [2], RCG [3]
- Asynchronous noise schedule

[1] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. ICLR 2025.
[2] Chenyu Wang*, **Cai Zhou***, Sharut Gupta, Zongyu Lin, Stefanie Jegelka, Stephen Bates, and Tommi Jaakkola. Learning diffusion models with flexible representation guidance. NeurIPS 2025.
[3] Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. NeurIPS 2024.

CCDD: Experiments

- Reduces over 25% valid. ppl. w/ same #params. and almost 0 computation overheads
- Further better performance w/ more #params.

Table 2. Validation perplexity on LM1B. We use Qwen3-Embedding-0.6B as the continuous generation space for CCDD, and reimplement the baselines with the same Qwen-2 tokenizer. CCDD-MDiT with the same number of parameters of FLOPs significantly outperforms the discrete-only MDLM baseline, reducing over 25% perplexity. CCDD-MoEDiT and MMDiT further improve the performance.

Model	Train. toks.	# params.	Validation PPL (↓)
MDLM (Sahoo et al., 2024) (reimpl.)	33B	92.1M	$\leq$ 39.17
CCDD-MDiT w/ Qwen3 (ours)	33B	92.1M	$\leq$ 29.22
CCDD-MoEDiT w/ Qwen3 (ours)	33B	104.0M	$\leq$ 28.50
CCDD-MMDiT w/ Qwen3 (ours)	33B	216.2M	$\leq$ **25.76**

CCDD: Experiments

Table 3. Validation perplexity on OWT with Qwen-2 tokenizer. CCDD is trained with Qwen3-Embedding-0.6B embeddings.

Model	Train. toks.	# params.	Validation PPL (↓)
MDLM (reimpl.)	131B	92.1M	$\leq$ 33.78
CCDD-MDiT w/Qwen3 (ours)	131B	92.1M	$\leq$ 29.18
CCDD-MoEDiT w/Qwen3 (ours)	131B	104.0M	$\leq$ **21.90**
CCDD-MMDiT w/Qwen3 (ours)	131B	124.0M	$\leq$ 26.59

Table 4. Validation perplexity on OWT with GPT-2 tokenizer. CCDD is trained with the simple RoBERTa-base or Qwen3-Embedding-0.6B embeddings, but still outperforms discrete baselines.

Model	Train. toks.	# params.	Validation PPL (↓)
GPT2 (Radford et al., 2019)†	unk.	117M	23.40
Llama110M (retrain.)†	262B	110M	16.11
SEDD (Lou et al., 2023)†	262B	92.1M	$\leq$ 24.10
MDLM (Sahoo et al., 2024) (reimpl.)	131B	92.1M	$\leq$ 27.39
GIDD+ (von Rütte et al., 2025) (reimpl.)	131B	92.1M	$\leq$ 25.82
CCDD-MoEDiT w/RoBERTa (ours)	131B	104.0M	$\leq$ 24.56

CCDD: Experiments

- Classifier-free guidance:
  - Training: randomly zero-in & zero-out continuous component
  - Inference: combine discrete-only logits (unconditional) and joint logits (conditional) using CFG scale w:

$$\text{logits} = w \cdot \text{logits}_c + (1 - w) \cdot \text{logits}_\phi$$

Table 5. Generative NLLs of CCDD with Qwen3-Embedding pretraining on OWT with Qwen-2 tokenizer using CFG.

Model	w	Gen. NLL (↓)
MDLM	-	9.19
CCDD-MoEDiT	0.0	9.06
CCDD-MoEDiT	1.0	8.38
CCDD-MoEDiT	1.5	**8.25**

CCDD: Experiments

- Few-step generation:
  - Lower generative ppl with 8 steps than baseline with 256 steps
  - Capability of global latent planning and modeling joint distribution!

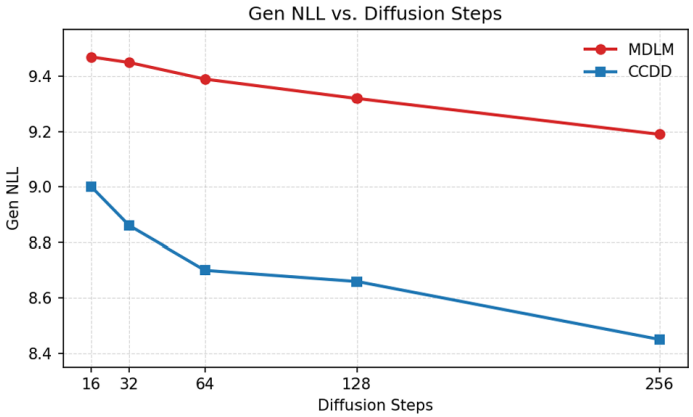


Figure 4. Few-step generative NLL with Qwen-2 tokenizer.

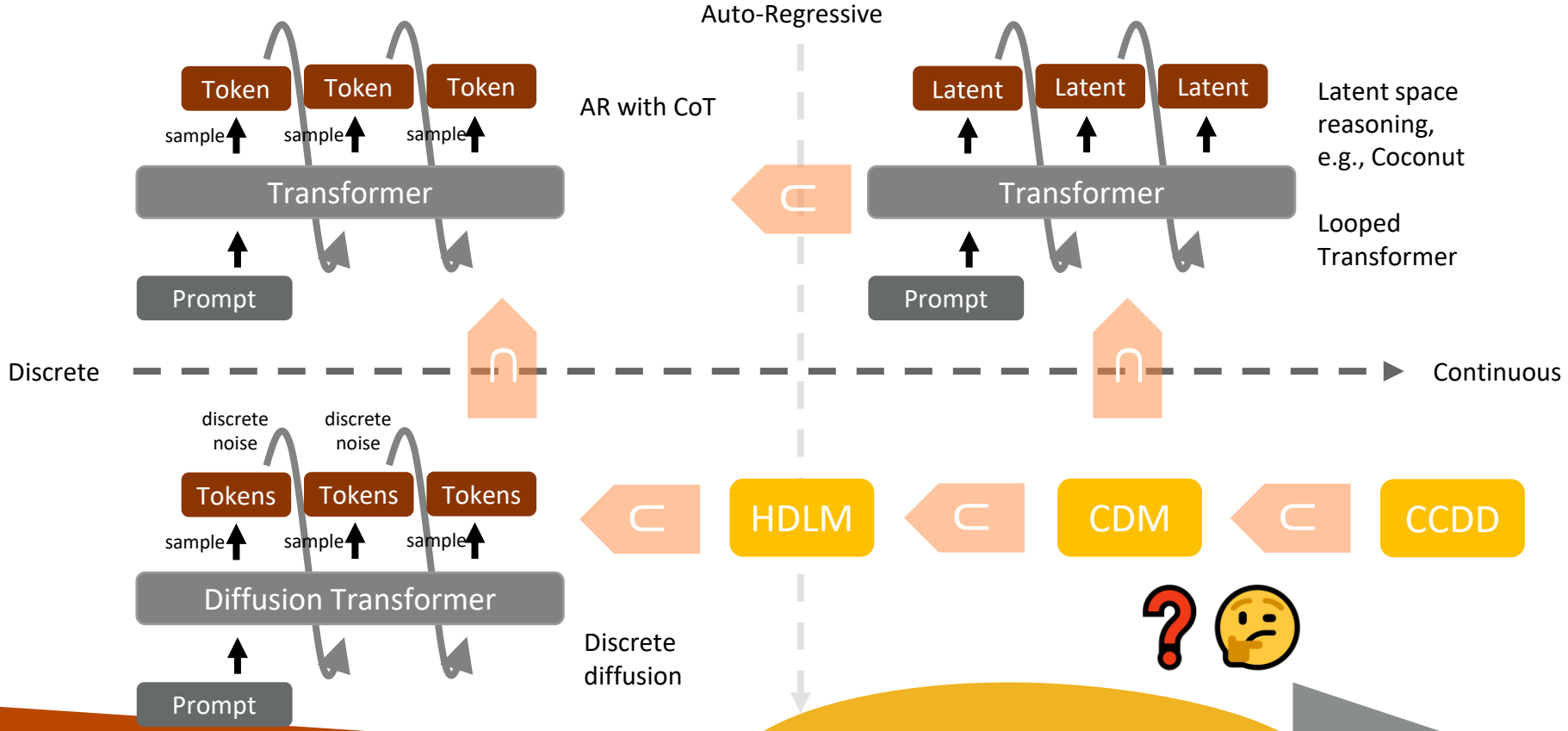
Axes of Reasoning Model Scaling

Table 6: Test accuracy on complex reasoning datasets: Sudoku, 3-SAT, and Countdown.

	Model	Size	Depth T	Sudoku	3-SAT	Countdown
CoT: Scaling in length	GPT2 scratch	6M	Seq. Length	16.2	73.1	31.9
	GPT2 scratch	303M	Seq. Length	19.4	-	41.3
	Llama	7B	Seq. Length	27.1	-	41.1
	Llama	13B	Seq. Length	33.8	-	51.1
MDM/LT: Scaling in depth	MDM	6M	2	88.6	39.2	-
	MDM	6M	20	99.9	87.0	52.0
	LT	6M	2	100.0	91.3	60.6
	LT	6M	3	100.0	-	68.2
	CCDD (ours)	6M	2	100.0	**91.9**	67.8
	CCDD (ours)	6M	3	100.0	-	**73.7**

- Interleave between these two axes extends the boundary of reasoning

Summary: Rethinking Current Paradigms of LLM Reasoning



Thank you!

MIT CSAIL
Computer Science & Artificial Intelligence Laboratory