# Jump Your Steps: Optimizing Sampling Schedule of Discrete Diffusion Models

Yonghyun Park[1], Chieh-Hsin Lai[1], Satoshi Hayakawa[2], Yuhta Takida[1], Yuki Mitsufuji[1,2]
[1]SONY AI, [2]Sony Group Corporation

Poster presentation @ ICLR25

# Discrete Diffusion Models (DDMs)

- Recently, DDMs show promising results on in various fields:
  - Image
  - Language
  - Gene, Protein

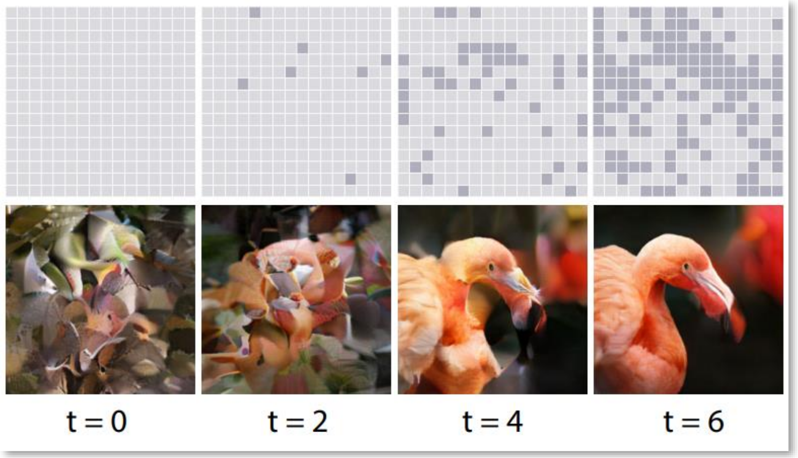
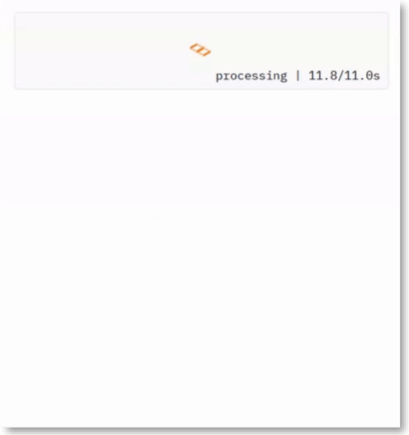
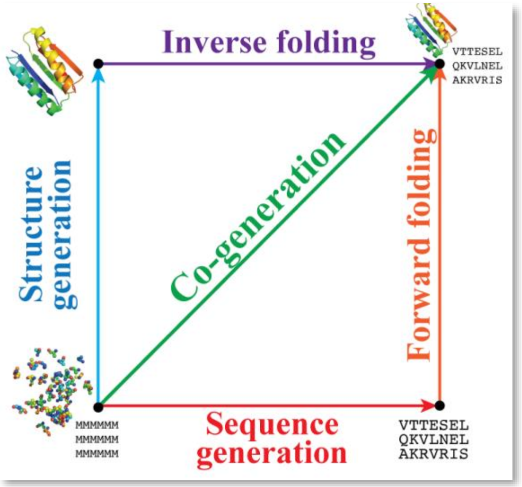



Image Generation
(MaskGiT)



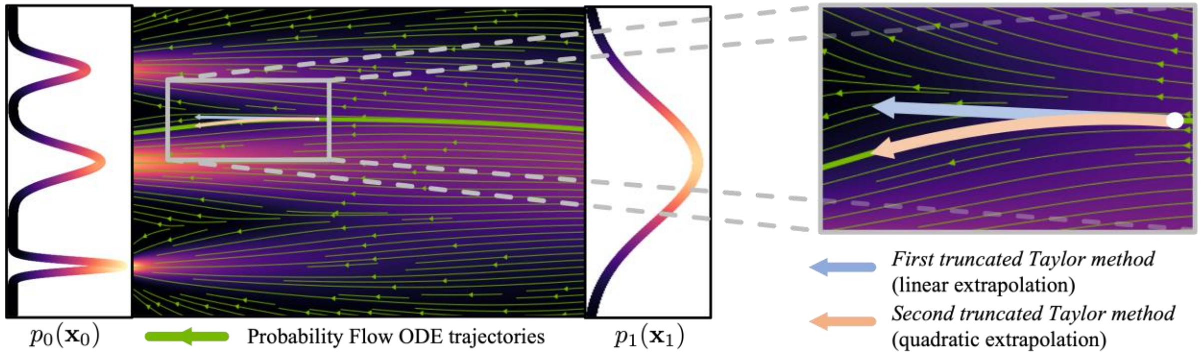

Language Generation
(LLaDA)



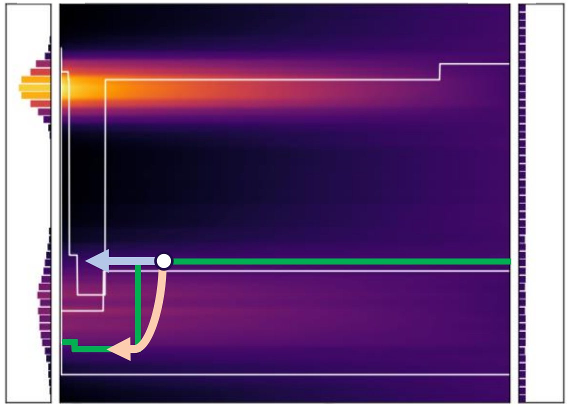

Gene / Protein Generation
(DFM)

# Challenges with DDMs

- **Slow sampling:**
  - DMs require multiple NFEs for generation.
- **Continuous DMs**
  - Solution: high-order ODE / SDE solver.
  - There are many previous work for fast and reliable differential equation solver.
- **Discrete DMs**
  - Solution: high-order Jump process solver?
  - There are little work for efficient Markov jump process solver.



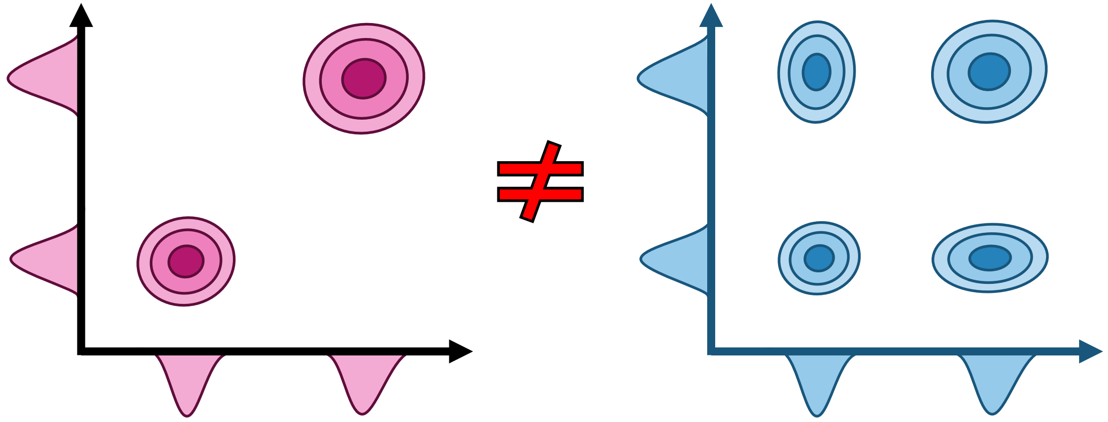

**GENIE: Higher-Order Denoising Diffusion Solvers**



**High-order Markov Jump process solvers are difficult to develop.**

# Challenges with DDMs

- **Faster sampling for DDMs: Parallel sampling**
  - Sampling multiple tokens within single step
    - $k$-Gillespie: Sample k tokens simultaneously.
    - $\tau$-leaping: Sample all tokens simultaneously within a given time frame.
- **Compounding Decoding Error (CDE)**
  - Generally, parallel sampling leads to error.
    - Parallel sampling: $p(x_1)p(x_2)$
    - Sequential sampling: $p(x_1)p(x_2|x_1)$
  - CDE $\coloneqq D_{KL}(p(x_1, x_2)||p(x_1)p(x_2))$



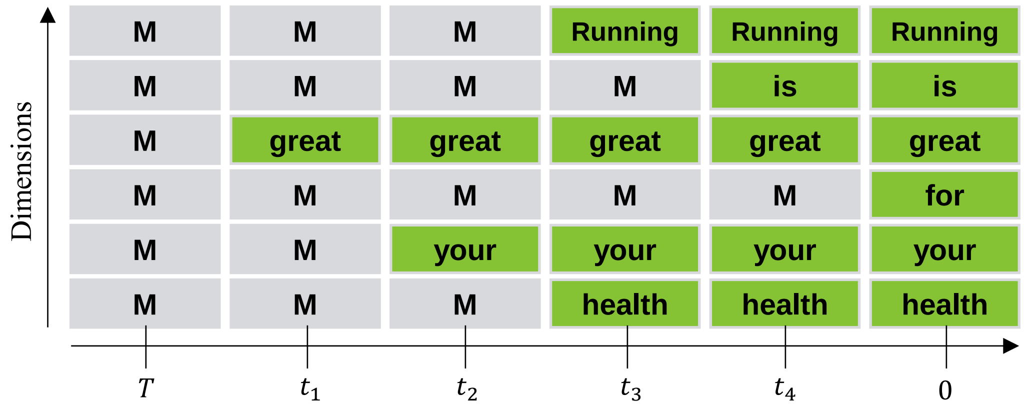
Joint distribution
(Sequential sampling, $p(x_1, x_2)$)

Product of marginal distribution
(Parallel sampling, $p(x_1)p(x_2)$)

# Main Idea: Motivating Example

- CDE depends on timesteps

$\mathit{Discrete\ sequence\ trajectory}:\ (X_t)_{t\in[0,T]}$

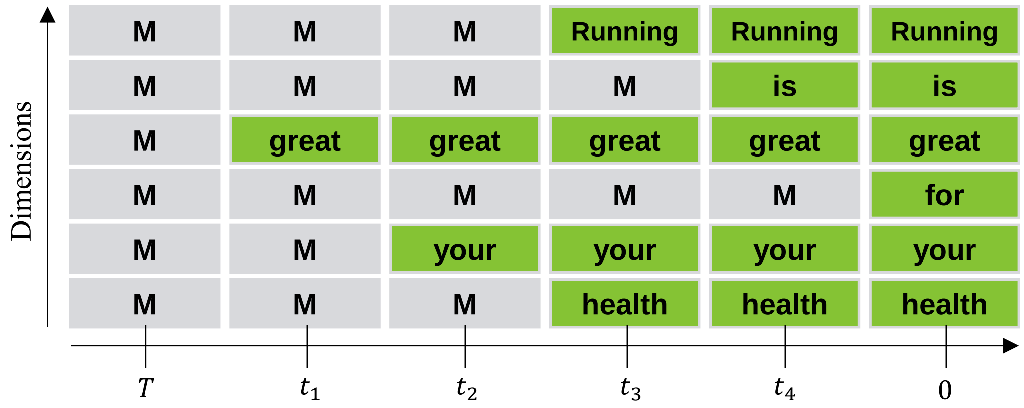



Ground-truth sampling trajectory

# Main Idea: Motivating Example

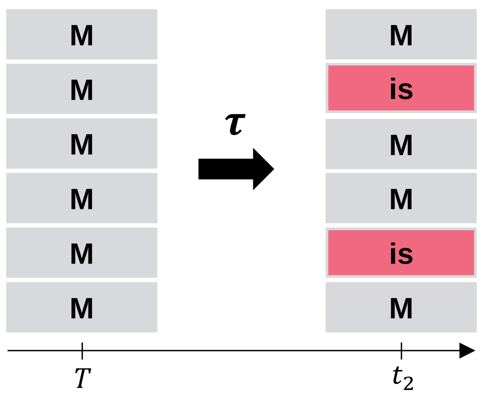
- CDE depends on timesteps

*Discrete sequence trajectory*: $(X_t)_{t \in [0,T]}$




Ground-truth sampling trajectory

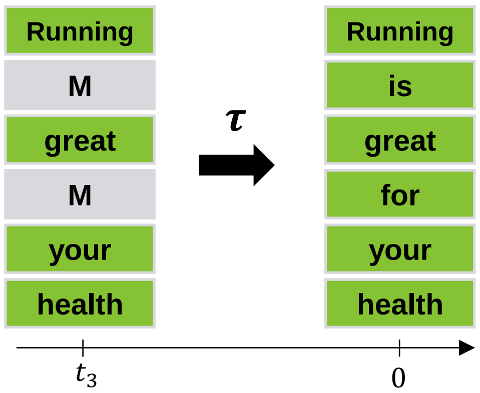
**$t \approx T$**




## High CDE

Parallel sampling hurts performance.

**$t \ll T$**



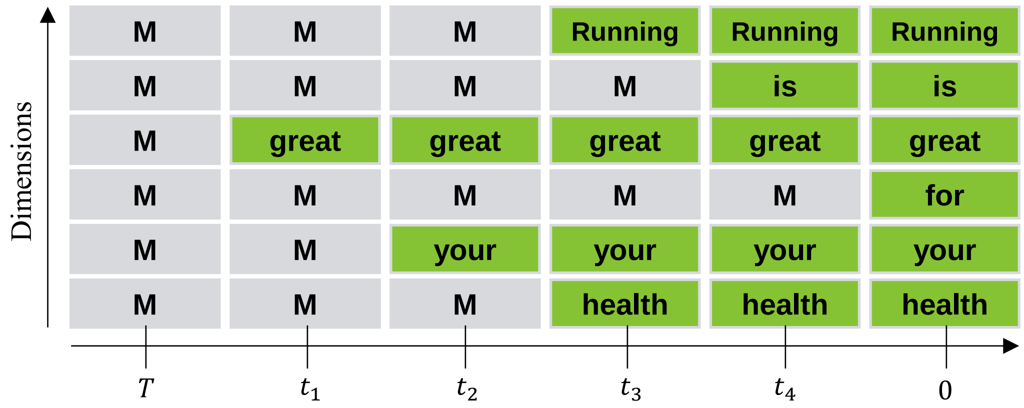

## Low CDE

Minimal impact from parallel generation.

# Main Idea: Jump Your steps

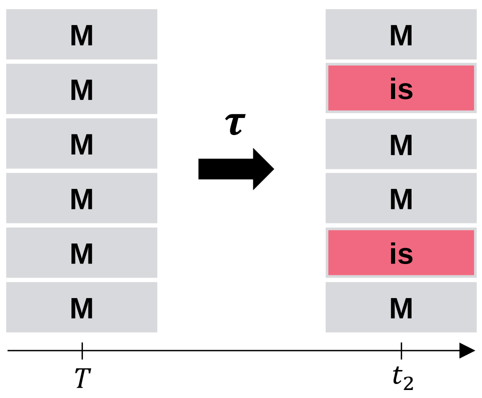
- CDE depends on timesteps

*Discrete sequence trajectory*: $(X_t)_{t \in [0,T]}$

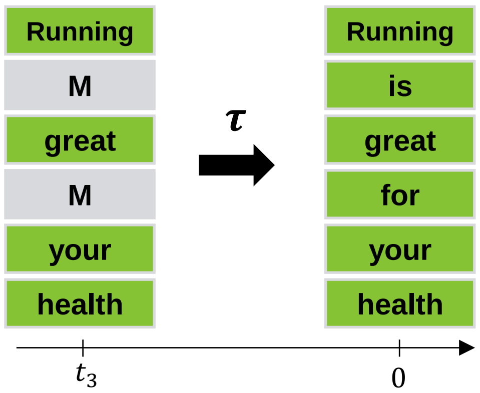



Ground-truth sampling trajectory

$t \approx T$




$t \ll T$




## High CDE

Parallel sampling hurts performance.

Use **small step size** when decoding error is large

## Low CDE

Minimal impact from parallel generation.

Use **large step size** when decoding error is small

**Optimizing Sampling Schedule of DDMs**

# Theory

$$\boxed{\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_0 \| \mathbb{Q}_0^{t_0 \to t_1 \to \cdots \to 0})} \overset{\text{Theorem 3.1}}{\leq} \sum_{i=0}^{N-1} \mathcal{D}_{\mathrm{KL}}(\mathbb{P}_{t_{i+1}} \| \mathbb{Q}_{t_{i+1}}^{t_i \to t_{i+1}}) \overset{\text{Eq. (10)}}{\leq} \mathcal{D}_{\mathrm{KL}}(\mathbb{P}_{\mathrm{paths}} \| \mathbb{Q}_{\mathrm{paths}}^{t_0 \to t_1 \to \cdots \to 0})$$

Sampling Quality

Eqs. (3, 5)

Theorem 3.2

$$\boxed{\sum_{i=0}^{N-1} \mathcal{E}_{\mathrm{CDE}}(t_i \to t_{i+1})}$$

$$\mathrm{KLUB}(\mathbb{P}_0 \| \mathbb{Q}_0^{t_0 \to t_1 \to \cdots \to 0})$$

Motivation: Minimize CDE

**Q:** Why should we minimize CDE?
**A:** It improves the sampling quality.

# Theory

$$\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_0 \| \mathbb{Q}_0^{t_0 \to t_1 \to \cdots \to 0}) \overset{\text{Theorem 3.1}}{\leq} \underbrace{\sum_{i=0}^{N-1} \mathcal{D}_{\mathrm{KL}}(\mathbb{P}_{t_{i+1}} \| \mathbb{Q}_{t_{i+1}}^{t_i \to t_{i+1}})}_{\text{Eqs. (3, 5)}} \overset{\text{Eq. (10)}}{\leq} \underbrace{\mathcal{D}_{\mathrm{KL}}(\mathbb{P}_{\text{paths}} \| \mathbb{Q}_{\text{paths}}^{t_0 \to t_1 \to \cdots \to 0})}_{\text{Theorem 3.2}}$$

$$\sum_{i=0}^{N-1} \mathcal{E}_{\mathrm{CDE}}(t_i \to t_{i+1})$$

**Motivation: Minimize CDE**

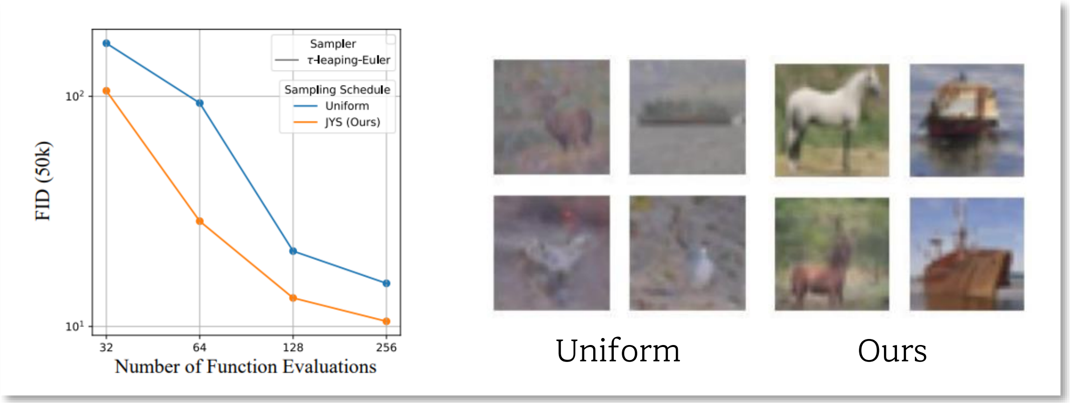
$$\mathrm{KLUB}(\mathbb{P}_0 \| \mathbb{Q}_0^{t_0 \to t_1 \to \cdots \to 0})$$

**Trackable Upper-bound**

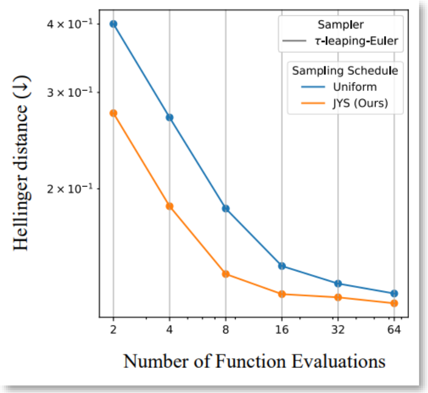
**Q:** How could we minimize CDE?
**A:** Using trackable upper-bound.

# Results

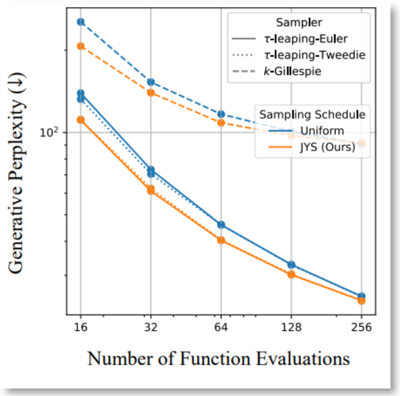
- **Improved Sampling quality:**
  - JYS sampling scheduler achieves performance improvement regardless of data domain or model.



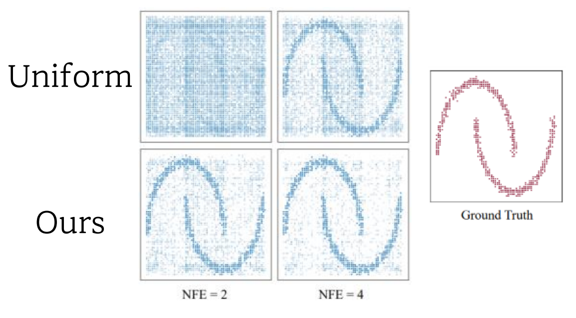

CIFAR10



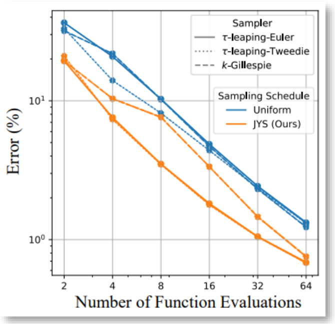

Piano




Language (GPT-2 scale)




2D Dataset




CountDown

Thank you