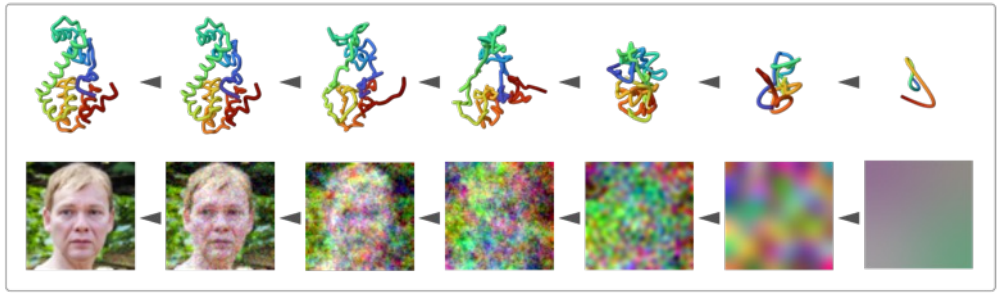
ERG 2026

# Exact RG for generative AI

Yuto Ashida (UTokyo)

Kanta Masuki

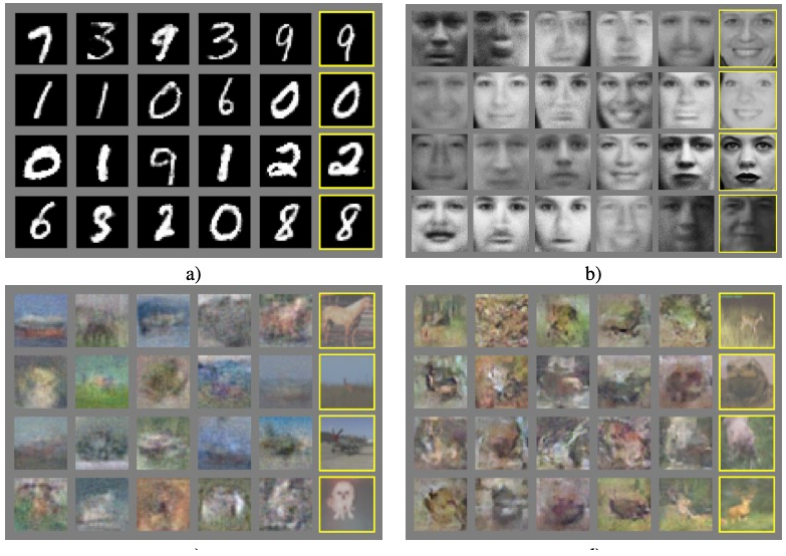


Based on arXiv: 2501.09064 and 2608.23696

# What is a generative model ?

Generative models: machine-learning models that generate images, audio, text …

| | |
|---|---|
| GANs | I. J. Goodfellow et al. 2014. |
| **Diffusion models** | J. Ho et al. 2019, Y. Song et al. 2019. |



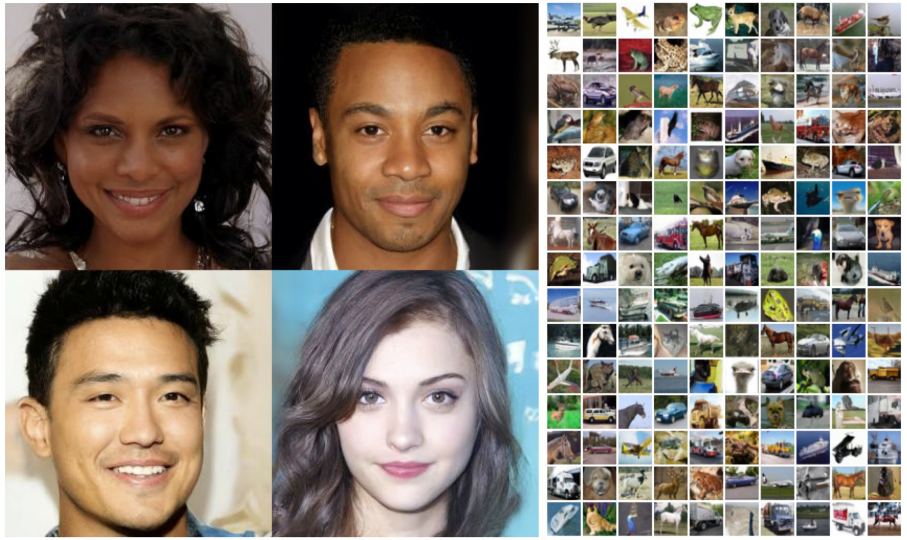



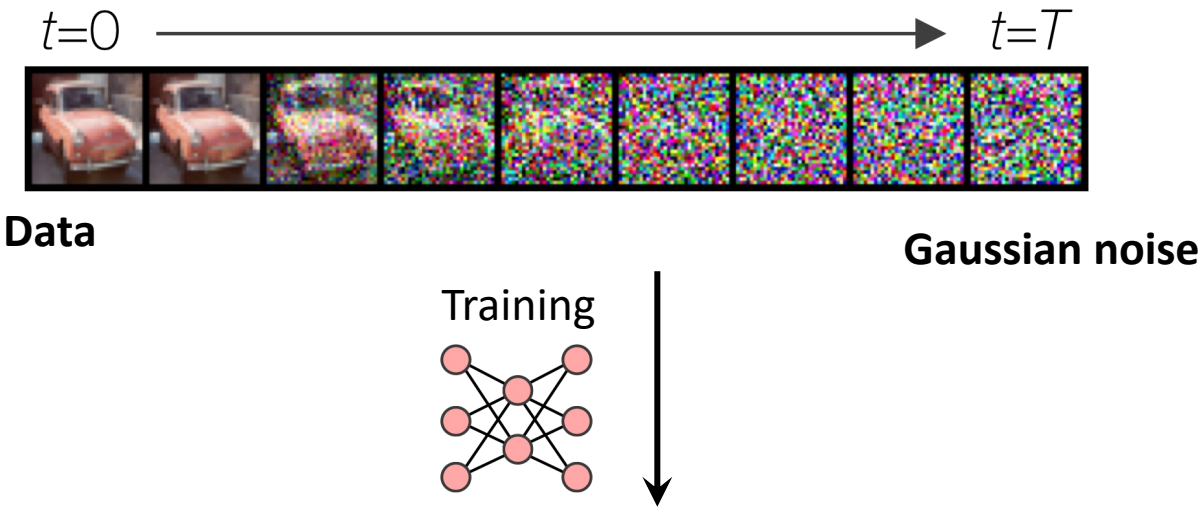
Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

**Diffusion models lie at the heart of many modern generative models (ChatGPT, image AI, …)**

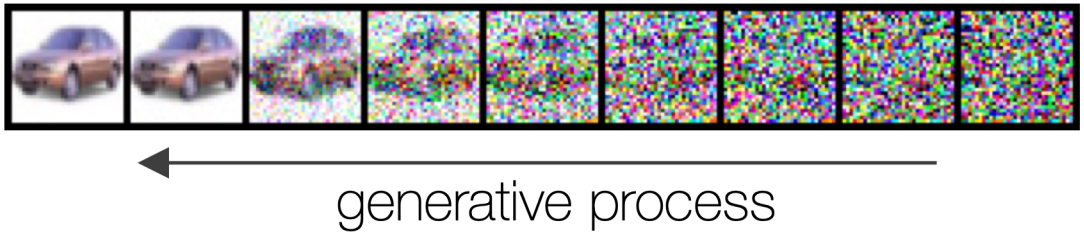
# Generative diffusion model

J. Sohl-Dickstein et al., 2015

**Diffusion** = Add noise to data gradually, converting data into Gaussian noise




Generate a sample by **reversing** the diffusion process
= eliminate noise iteratively, converting Gaussian noise into data.



# Diffusion process

**Diffusion:**

$$d\boldsymbol{x} = \underbrace{-\boldsymbol{x}\, dt}_{\text{drift}} + \underbrace{\sqrt{2} d\boldsymbol{w}_t}_{\text{noise}}.$$

$d\boldsymbol{w}_t$: Wiener process

$\mathbb{E}[d\boldsymbol{w}_t d\boldsymbol{w}_{t'}^T] = I\, dt$

$$\partial_t p_t = \nabla \cdot (\boldsymbol{x} p_t) + \nabla^2 p_t$$

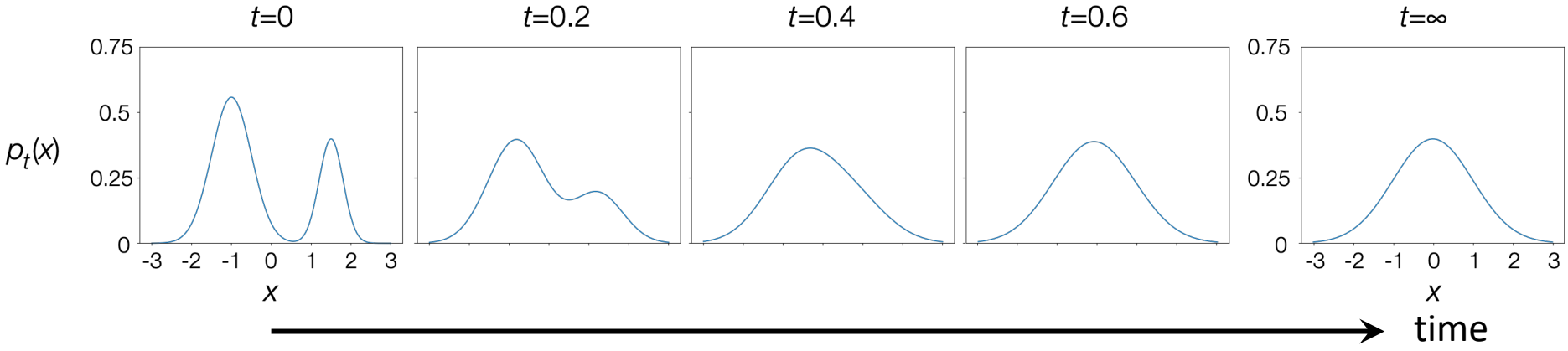
$p(\boldsymbol{x})$: prob. desntiy

**Gaussian kernel:**

$$p_t(x_t) = \int dx_0\, p(x_t|x_0) p_{t=0}(x_0)$$

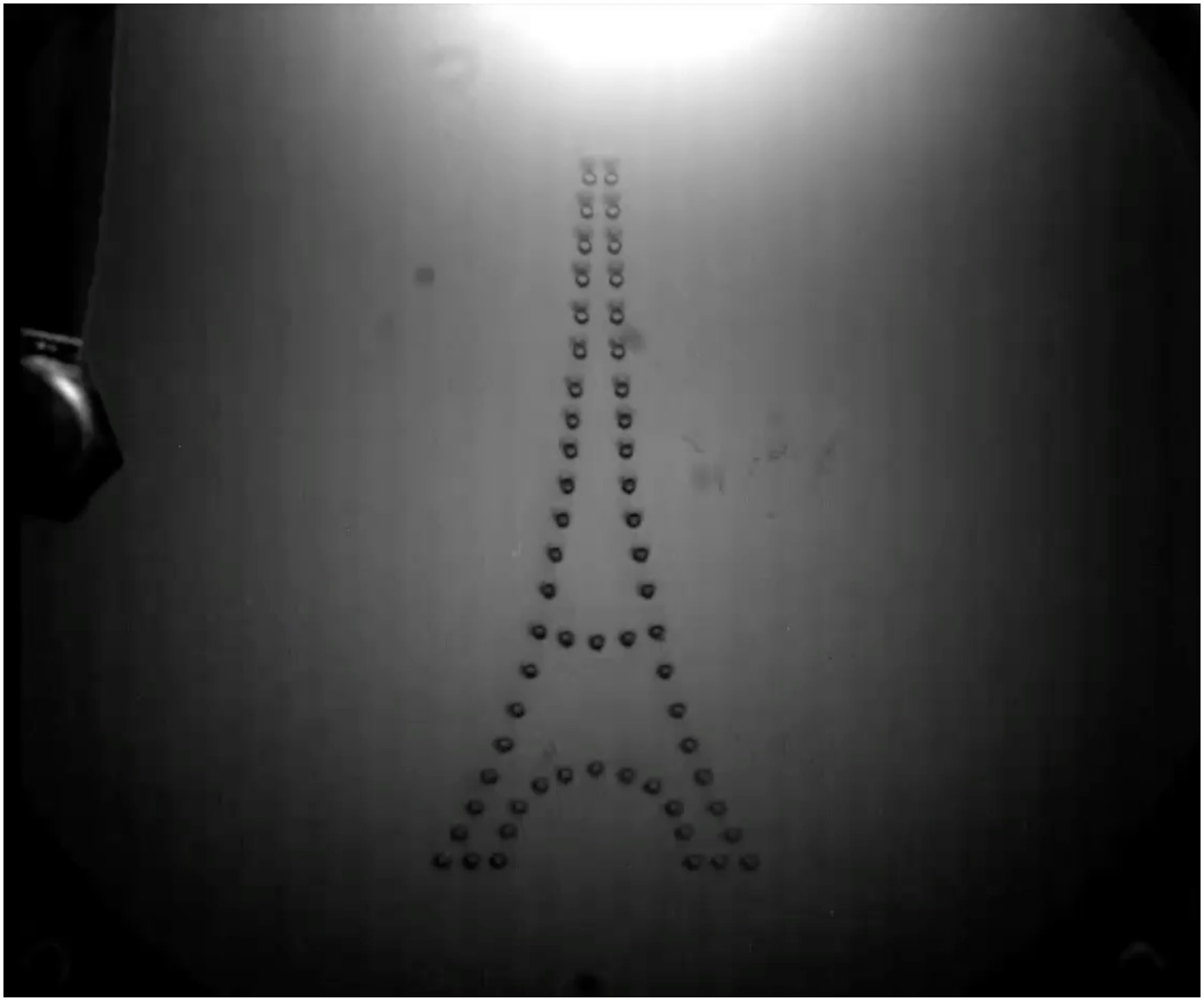
$$p(x_t|x_0) = \mathcal{N}(x_t; e^{-t}x_0, 1 - e^{-2t}) \to \mathcal{N}(x_t; 0, I) \quad t \to \infty$$

Any distribution converges to $\mathcal{N}(0, I)$ in $t \to \infty$ : (appears to be) irreversible




However, the **entire** history of $p_t$ contains much more information than the endpoint.

## Irreversible process can be "reversed" with precise knowledge of the prob density $p_t(x)$



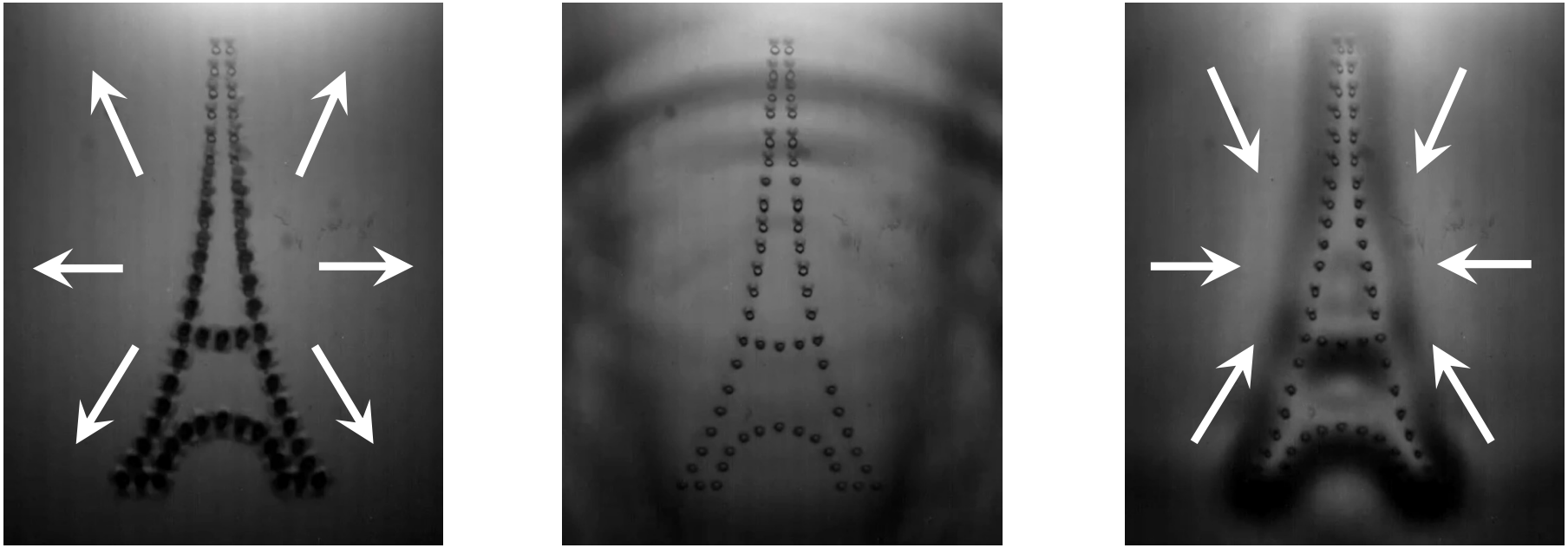
https://www.youtube.com/watch?v=ndq55y69hH8&list=WL&index=7

## Irreversible process can be "reversed" with precise knowledge of the prob density $p_t(x)$

Forward

$t = 0$ ⟶ $t = T$ ⟶ $t \simeq 0$

"Reverse"

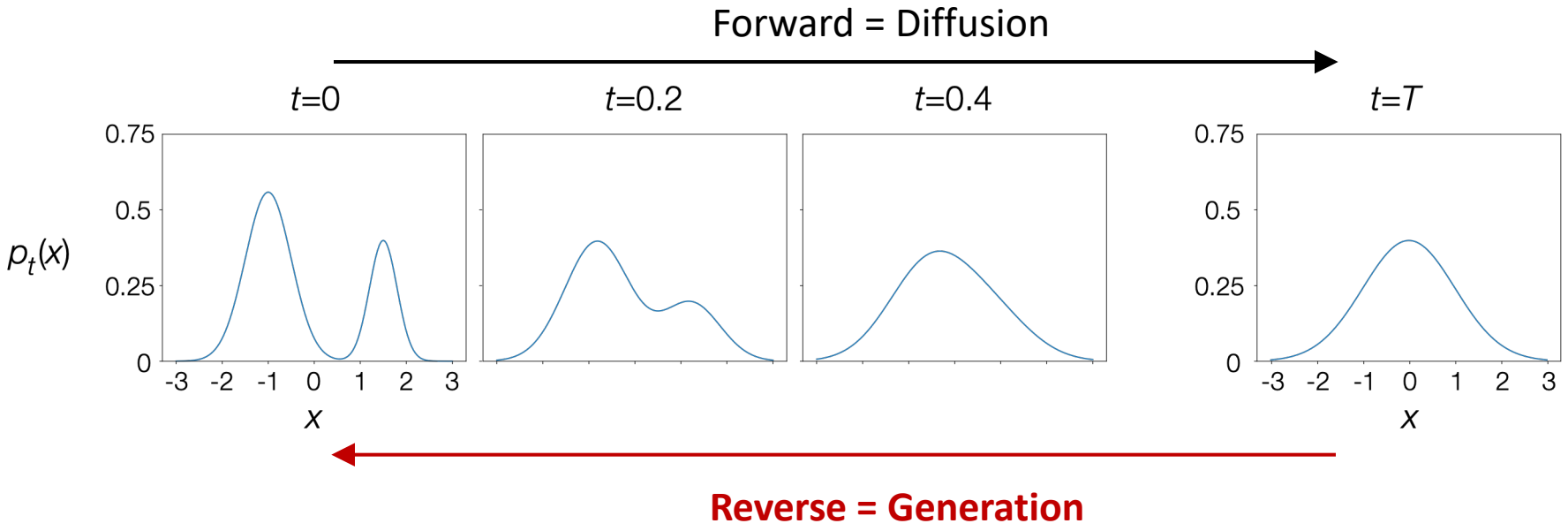


**Apply fine-tuned external perturbation**

**Lesson**: if we have the detailed knowledge of the dynamics,
we can reverse the seemingly irreversible evolution.

https://www.youtube.com/watch?v=ndq55y69hH8&list=WL&index=7

# Diffusion model: "reverse" diffusion via knowledge of $p_t(x)$




**score** : a fine-tuned external "force"

**Reverse evolution**

$$t' = T - t$$

$$d\boldsymbol{x} = \boldsymbol{x}\, dt' + 2\nabla_x \ln p_{t'}(\boldsymbol{x})\, dt' + \sqrt{2} d\boldsymbol{w}_t$$

$$\partial_{t'} p_{t'}(\boldsymbol{x}) = -\partial_t p_t(\boldsymbol{x}) \Big|_{t=T-t'}$$

**Diffusion model:**

**[Training]** Approximate the score with NN and learn it from data.

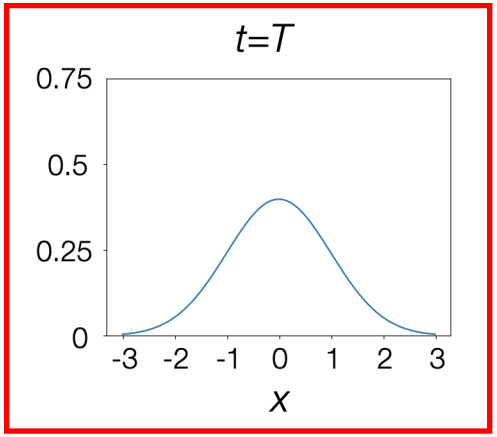
**[Sampling]** Use the reverse process to update $\boldsymbol{x}_t$ from $t = T$ to $t = 0$.

# Diffusion model: "reverse" diffusion via knowledge of $p_t(x)$

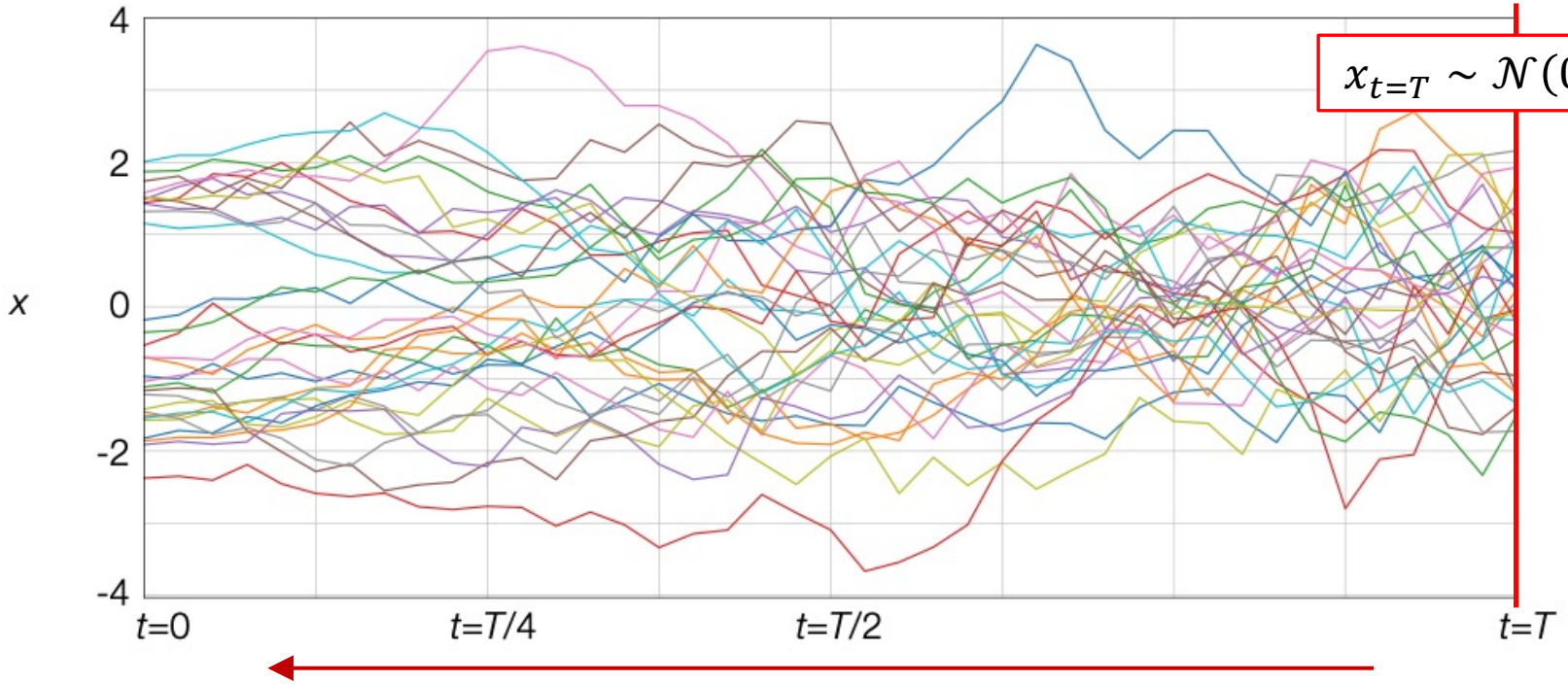
**Reverse = Generation**

$$d\boldsymbol{x} = \boldsymbol{x}\, dt' + 2\nabla_x \ln p_{t'}(\boldsymbol{x})\, dt' + \sqrt{2} d\boldsymbol{w}_t$$

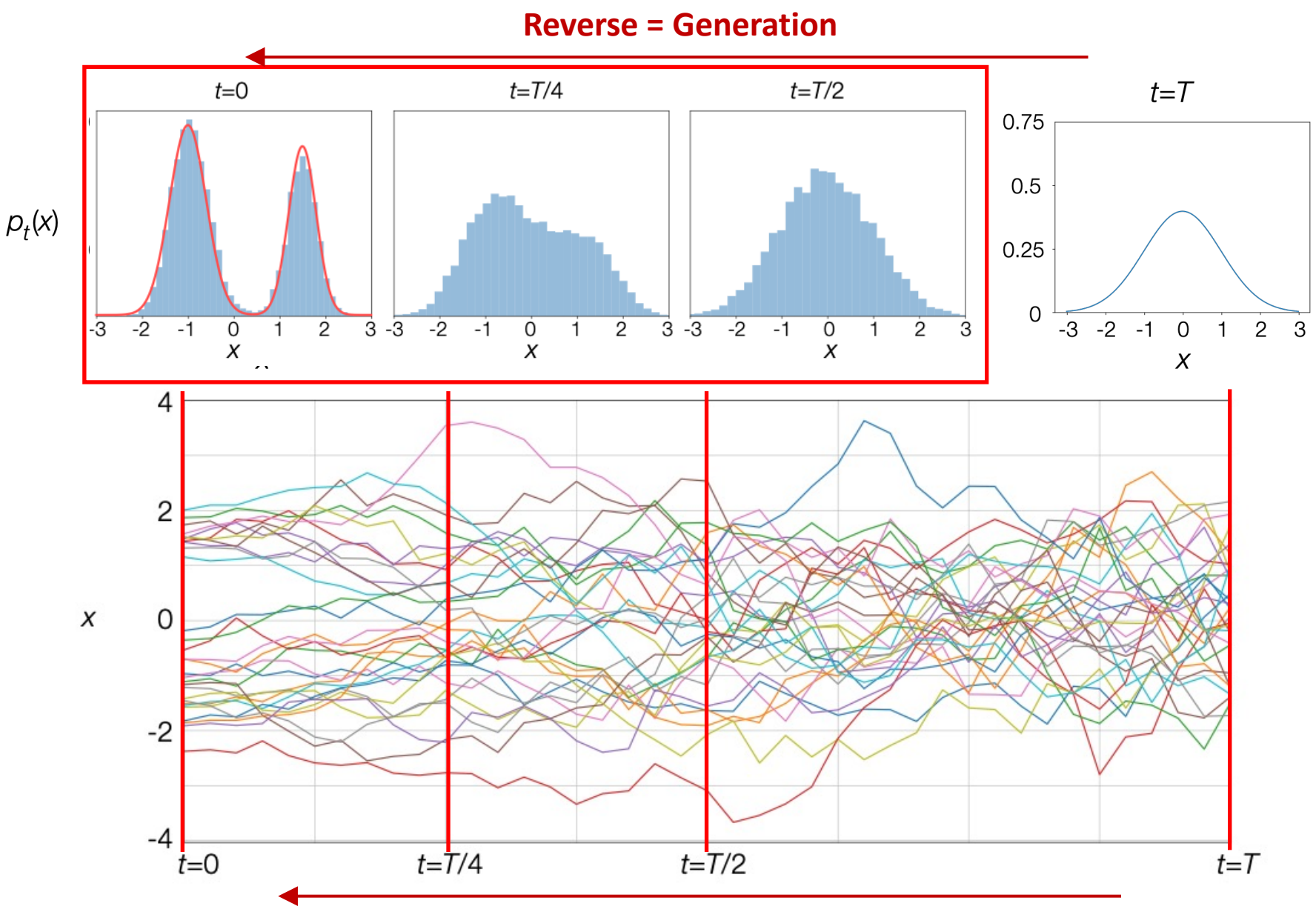
$$t' = T - t$$




$x_{t=T} \sim \mathcal{N}(0,1)$



# Diffusion model: "reverse" diffusion via knowledge of $p_t(x)$



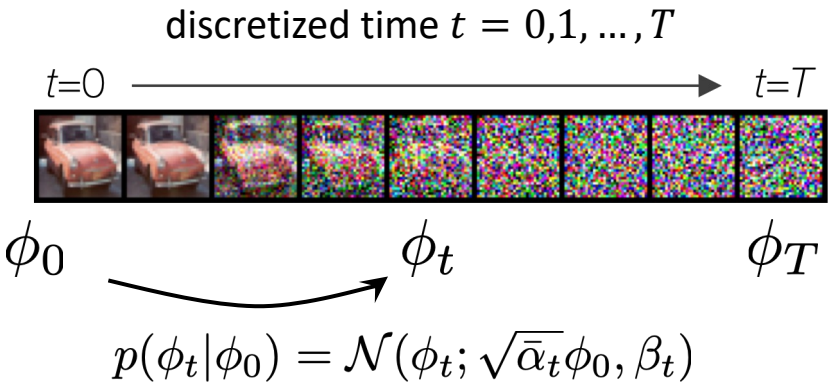
# Generative diffusion model

**Forward process (diffusion):**

$$\phi_t = \sqrt{\bar{\alpha}_t}\phi_0 + \sqrt{\bar{\beta}_t}\epsilon$$

$\bar{\alpha}_t, \bar{\beta}_t$ : noise schedule $\quad \bar{\alpha}_t + \bar{\beta}_t = 1$

$\epsilon \sim \mathcal{N}(0, I)$ : **white** noise

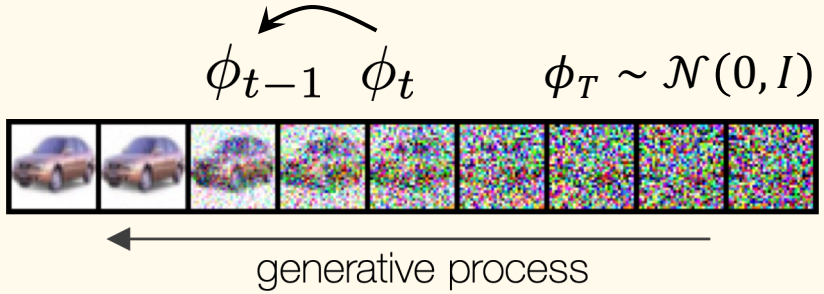



**Reverse process (generation):**

score

$$\phi_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left[\phi_t - \beta_t\, s_\theta(\phi_t, t)\right] + \sqrt{\beta_t}\epsilon$$

$$\alpha_t = \bar{\alpha}_t/\bar{\alpha}_{t-1} \qquad \alpha_t + \beta_t = 1$$




✗ Highly **sensitive** to a choice of the noise schedule $\alpha_t, \beta_t$, which needs to be **heuristically** optimized due to the lack of guiding principles.

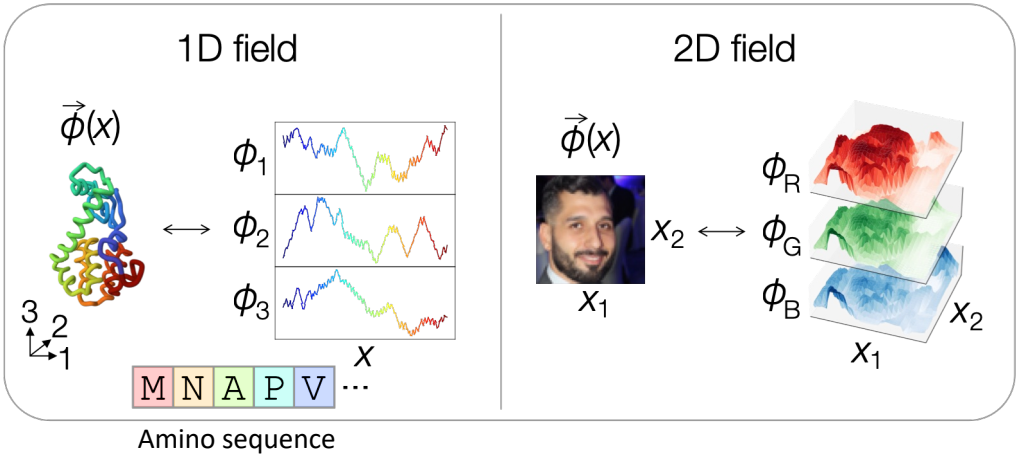
Ho+ '19; Chen '23, Hoogeboom+ '23

✗ Destroys features at **all** length scales **simultaneously** (ignores scale structure behind data)

**Use RG to overcome these limitations ?**

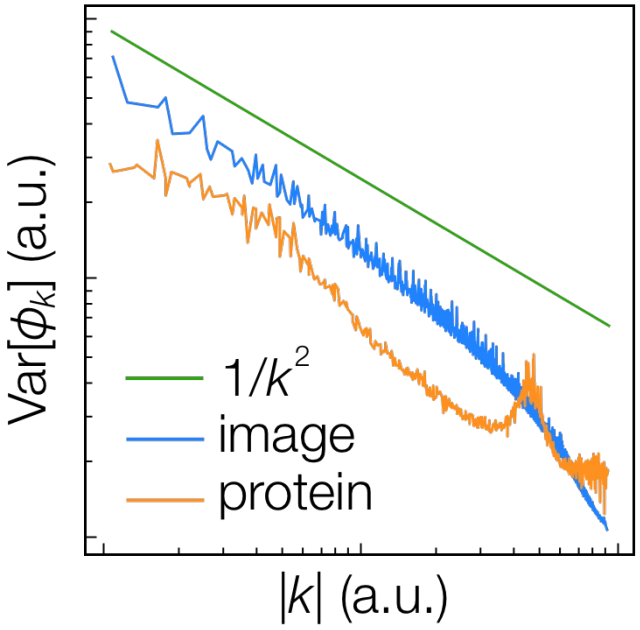
# Why RG? Multiscale Structure Across Diverse Data

Many types of natural data typically have multiscale spatial structures




Protein ：
Vector field $\phi(x) \in \mathbb{R}^3$ on 1D lattice
(x: position of an amino acid; $\phi$ : displacements)

Image ：
Vector field $\phi(x) \in \mathbb{R}^3$ on 2D lattice
(x: pixel position; $\phi$ : RGB values )




$$\mathrm{Var}[\phi_k] \sim k^{-2}$$ $\phi_k$ : Fourier modes

Effective "action" of data distribution :

$$p_\Lambda(\phi) \propto e^{-H_\Lambda(\phi)}$$

$$H_\Lambda(\phi) = \int_{|k|<\Lambda} \frac{d^d k}{(2\pi)^d}\, k^2 |\phi_k|^2 + V_\Lambda(\phi)$$

# Coarse graining of data distribution by Exact RG

Prob. density $p_\Lambda(\phi) \propto e^{-H_\Lambda(\phi)}$

$$H_\Lambda(\phi) = \int_{|k|<\Lambda_{\mathrm{UV}}} \frac{d^d k}{(2\pi)^d} K_\Lambda^{-1}(k) k^2 |\phi_k|^2 + V_\Lambda(\phi)$$

eliminate $\Lambda < k < \Lambda_{UV}$ ↓

constraint $\langle \phi_{k_1} \cdots \phi_{k_n} \rangle_{p_{\Lambda_{\mathrm{UV}}}} = \langle \phi_{k_1} \cdots \phi_{k_n} \rangle_{p_\Lambda} \ (|k_1|, \ldots, |k_n| < \Lambda)$

Polchinski eq. : $$\partial_\Lambda V_\Lambda = -\frac{1}{2} \int_k \frac{d^d k}{(2\pi)^d} \, k^{-2} \partial_\Lambda K_\Lambda(k) \left( \frac{\delta^2 V_\Lambda}{\delta\phi_k \delta\phi_{-k}} - \frac{\delta V_\Lambda}{\delta\phi_k} \frac{\delta V_\Lambda}{\delta\phi_{-k}} \right)$$

Legendre dual to Wetterich eq./ (subclass of) Wegner-Morris eq.

↓ "time" = log RG scale $t = \ln \frac{\Lambda_{UV}}{\Lambda_t}$

Polchinski (1984); Cotler et al. (2023)

Flow eq of $p_t \equiv p_{\Lambda_t}$ :

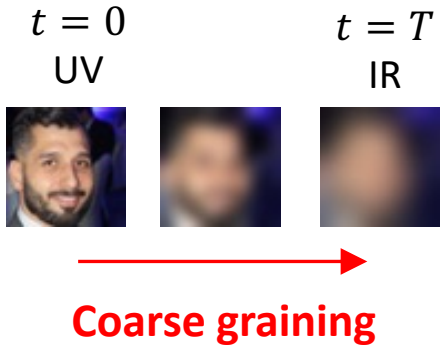
$$\partial_t p_t(\phi) = -\frac{1}{2} \int \frac{d^d k}{(2\pi)^d} \left[ \underbrace{k^{-2} \partial_t K_t(k) \frac{\delta^2 p_t}{\delta\phi_k \delta\phi_{-k}}}_{\text{diffusion}} + \underbrace{\frac{\delta}{\delta\phi_k} \left( \frac{\partial_t K_t(k)}{K_t(k)} \phi_k p_t \right)}_{\text{drift}} \right]$$

→ rigorously describes the change of $p_t$ under **coarse graining**

# RGDM: Renormalization Group-based Diffusion Model

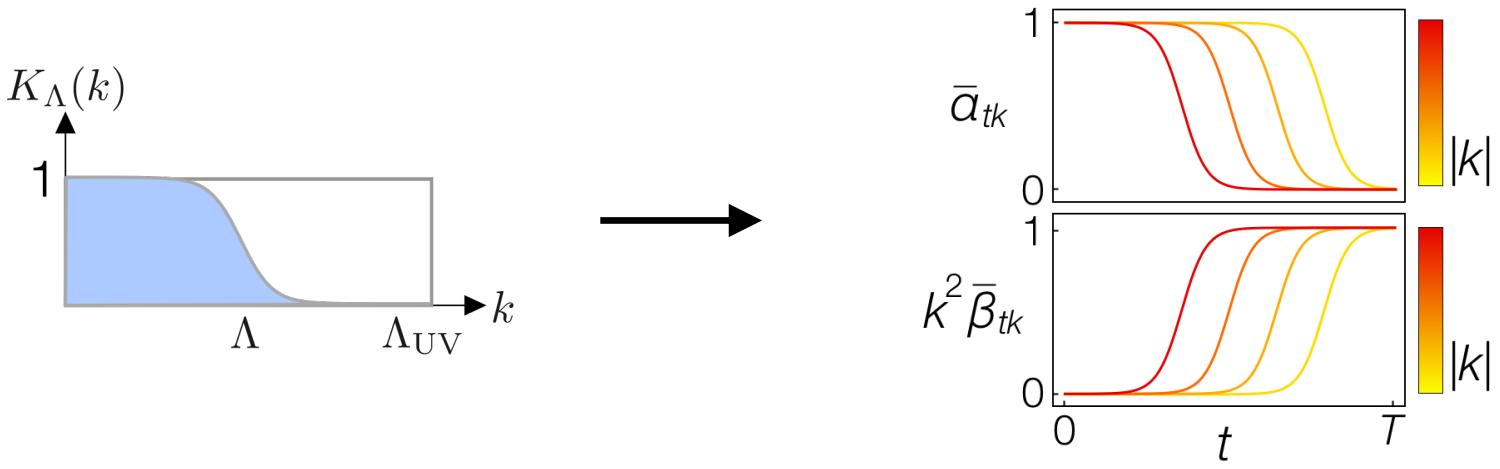
## Forward process (iterative coarse graining)

$$\phi_{tk} = \sqrt{\bar{\alpha}_{tk}}\phi_{0k} + \sqrt{\bar{\beta}_{tk}}\,\epsilon,$$
$$\bar{\alpha}_{tk} = K_t(k),\ \ \bar{\beta}_{tk} = k^{-2}(1-\bar{\alpha}_{tk}),\ \ \epsilon \sim \mathcal{N}(0,1)$$




✓ ***k*-dependent** noise schedule $\alpha_{tk}, \beta_{tk}$ → destroys high $k$ modes first (**RG ordering**)

✓ Noise schedules are **unambiguously** fixed once $K_\Lambda$ is specified
→ giving a guiding principle: **removing** the need for **heuristic** tuning

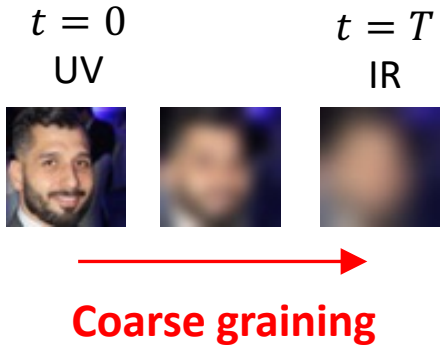



typical choice : $K_t(k) = \frac{r(k^2/\Lambda_t^2)}{1+r(k^2/\Lambda_t^2)}$ with $r(x) = x^{-a}$ or $(e^x-1)^{-1}$ cf. Litim 2000, 2001.

# RGDM: Renormalization Group-based Diffusion Model

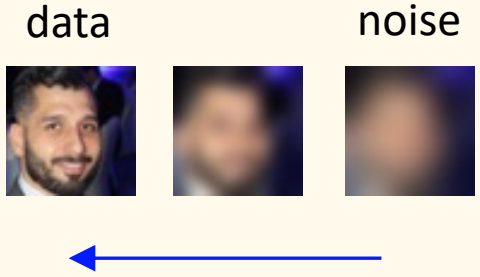
## Forward process (iterative coarse graining)

$$\phi_{tk} = \sqrt{\bar{\alpha}_{tk}}\phi_{0k} + \sqrt{\bar{\beta}_{tk}}\,\epsilon,$$
$$\bar{\alpha}_{tk} = K_t(k),\ \ \bar{\beta}_{tk} = k^{-2}(1-\bar{\alpha}_{tk}),\ \ \epsilon \sim \mathcal{N}(0,1)$$




## Reverse process (inverse RG flow)

$$\phi_{t-1k} = \frac{1}{\sqrt{\alpha_{tk}}}\left(\phi_{tk} - \frac{\beta_{tk}}{\bar{\beta}_{tk}}\xi_{\theta k}(\phi_t, t)\right) + \sqrt{\beta_{tk}}\epsilon_k$$
$$\alpha_{tk} = \bar{\alpha}_{tk}/\bar{\alpha}_{t-1k},\ \ \beta_{tk} = k^{-2}(1-\alpha_{tk})$$

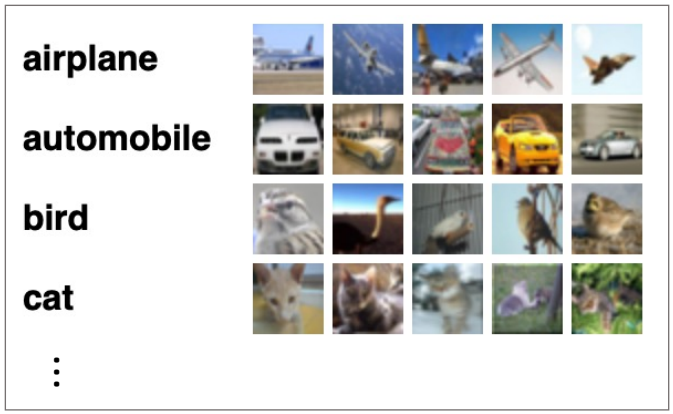



- ✓ Use NN to express $k$-dependent score $\xi_{\theta k}$, train it from data
- ✓ Generate data in a **coarse-to-fine manner (low-to-high $k$)**

Masuki and YA, arXiv:2501.09064

# Application 1 : image generation

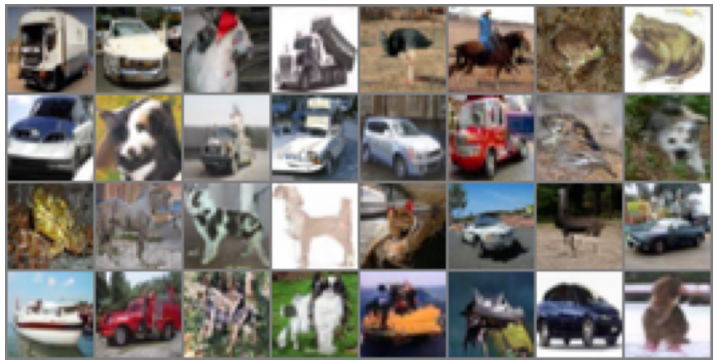
CIFAR-10 (32x32)




FFHQ (64x64)

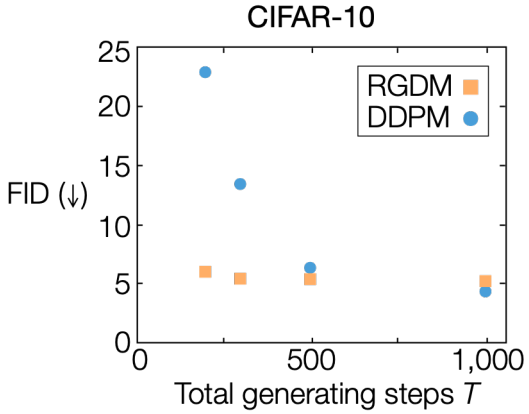
# Application 1 : image generation

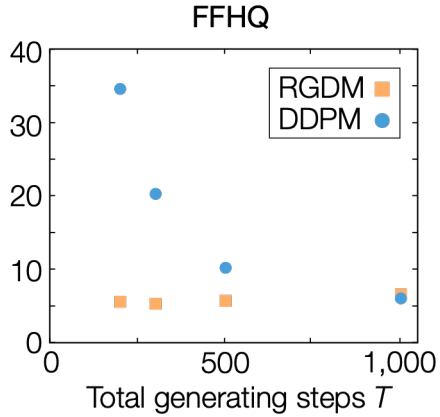
## Images generated by RGDM



**Frechét interception distance = FID** ~ indicator for quantifying 'distance' between $p_{\text{data}}$ and $p_{\text{model}}$

## Comparisons to DDPM

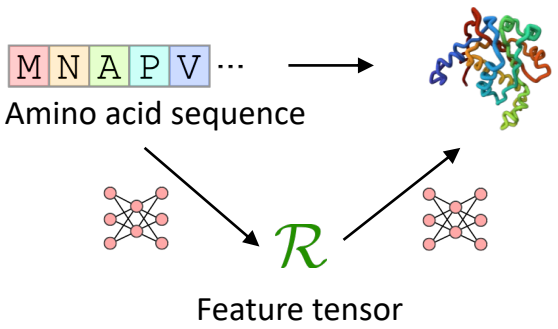



generate $5\times10^4$ images

RGDM can generate a high-quality data at fewer steps *T* than DDPM
-> Better sample efficiency

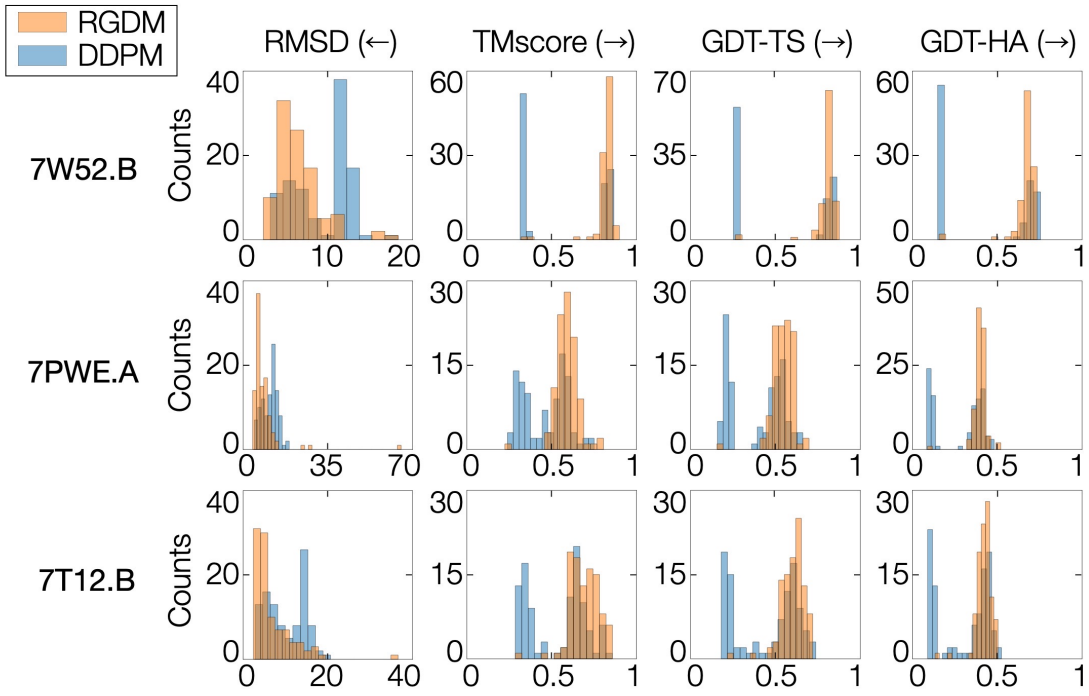
# Application 2 : Protein structure prediction

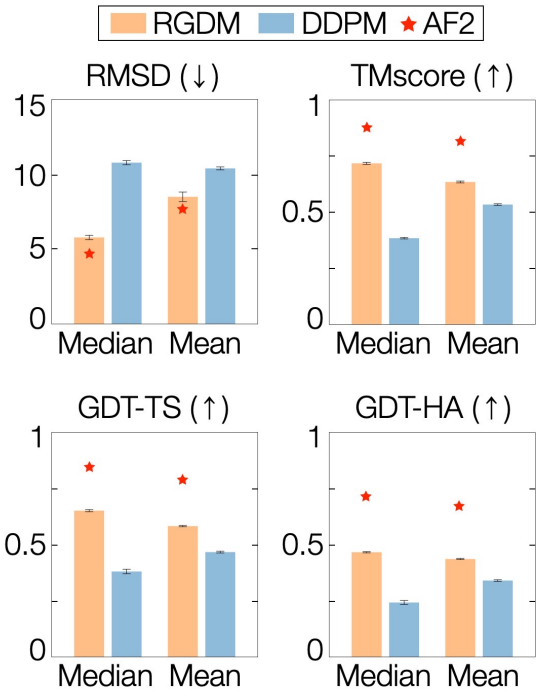



Protein : a chain composed of 20 types of amino acids

**Structure prediction**

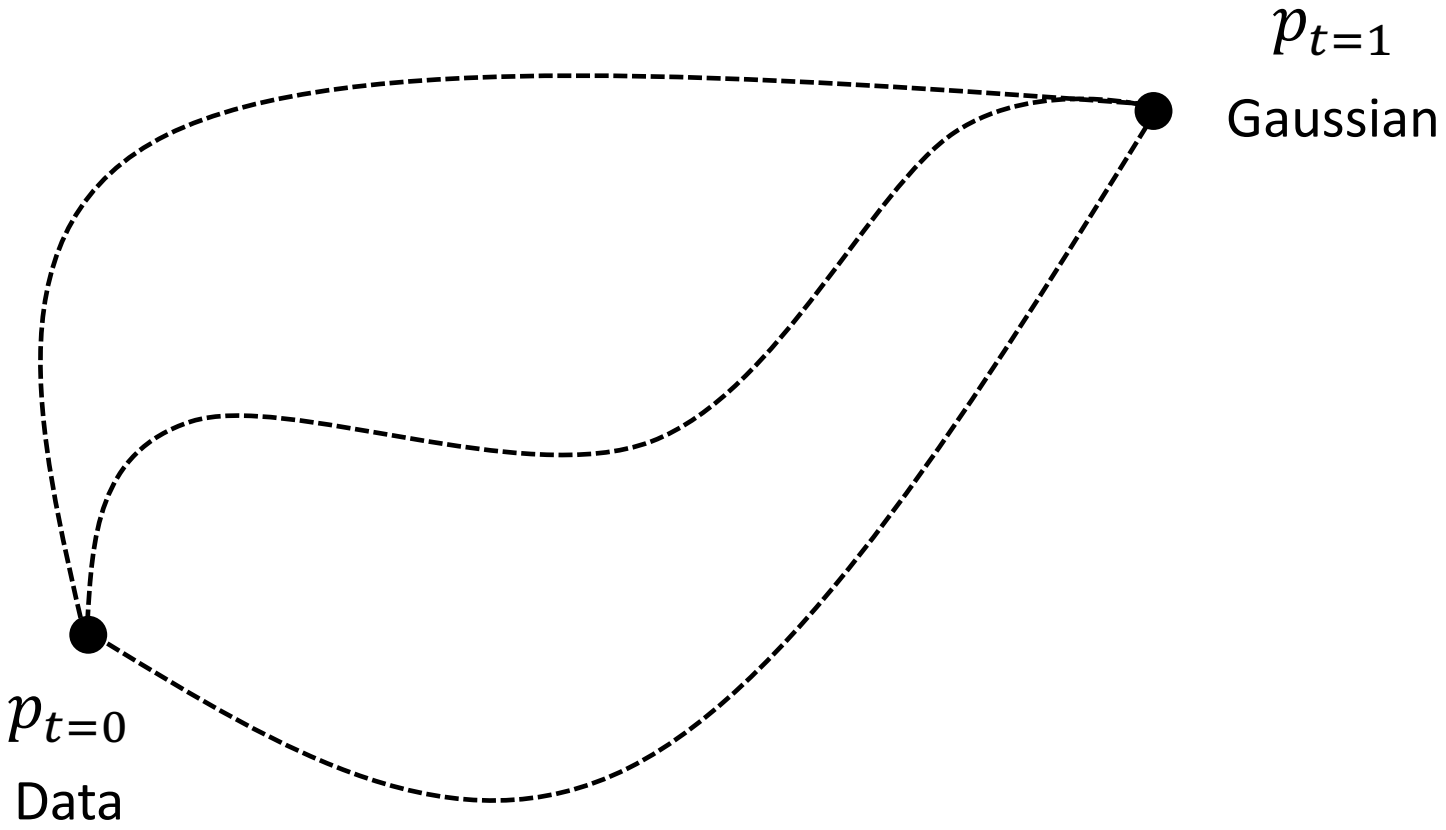
= inferring molecular structure from a given sequence



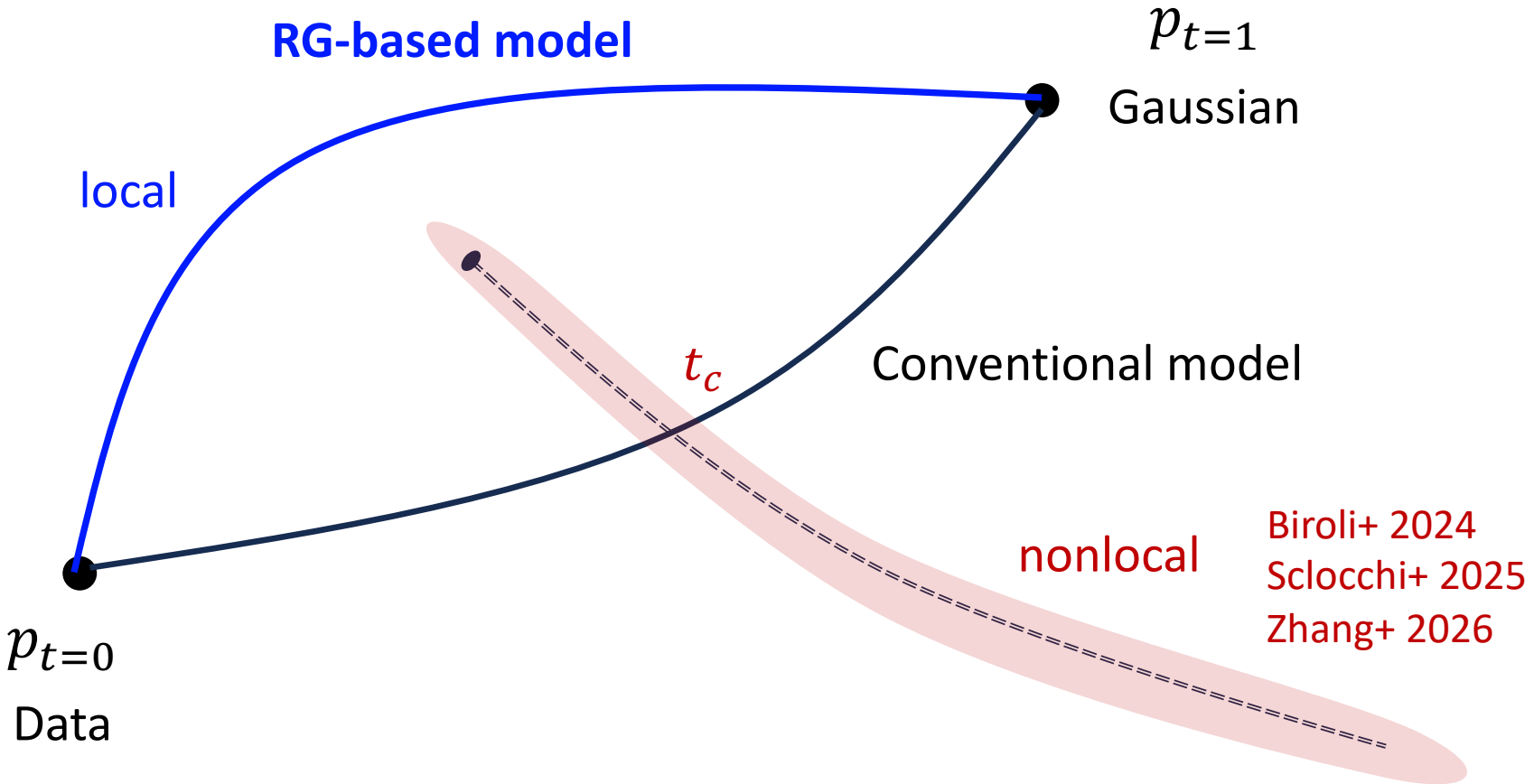

RGDM performs better in terms of stability and accuracy than DDPM

# Why RG generative model better ? : Locality theorem



# Why RG generative model better ? : Locality theorem

Choose a probability path on which the effective `action' $S \sim -\ln p_t$ remains **local**.




Masuki and YA arXiv:2608.23696

# Why RG generative model better ? : Locality theorem

Local information is enough to recover data distribution in RG model.

**Locality theorem (informal) :**

For a physical target distribution $p_{\mathrm{data}}$ and for any error tolerance $\epsilon > 0$, the RG probability path can be approximated by a local denoiser acting on a region of size $\ell_t$ that scales as

$$\ell_t = O\left(\Lambda_t^{-1} \ln \frac{L}{\epsilon}\right)$$

where $L$ is size of a sample, $\Lambda_t$ is a running momentum cutoff, and the 2-Wasserstein distance is bounded as

$$W_2\left(p_{\mathrm{data}}, p_{\mathrm{data}}^{\mathrm{loc}}\right) \leq \epsilon$$

Key ideas of the proof:

(quasi)locality of the UV action and the ERG flows

→ bound on local approximation error of denoiser

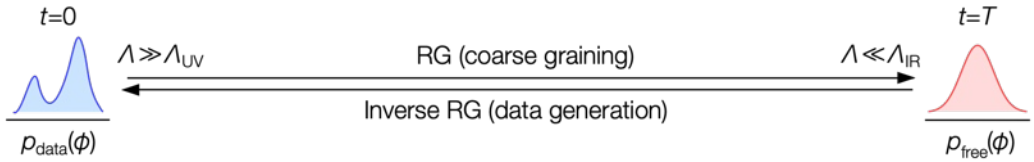
→ bound on local approximation error of reconstructed distribution $p_{\mathrm{data}}^{\mathrm{loc}}$

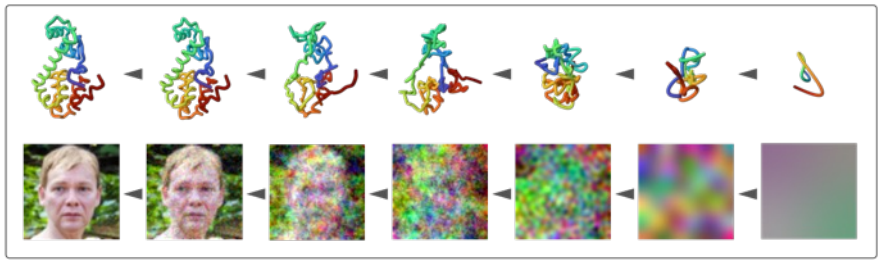
Masuki and YA arXiv:2608.23696

# Summary

- Proposed a generative model based on the exact RG.
- The model generates data in coarse-to-fine manner by reversing the RG flows.
- Numerical experiments suggest the possibility of improving robustness and sample efficiency through the RG.

All the codes available in Github; please try!



Masuki and YA,
arXiv:2501.09064 and 2608.23696