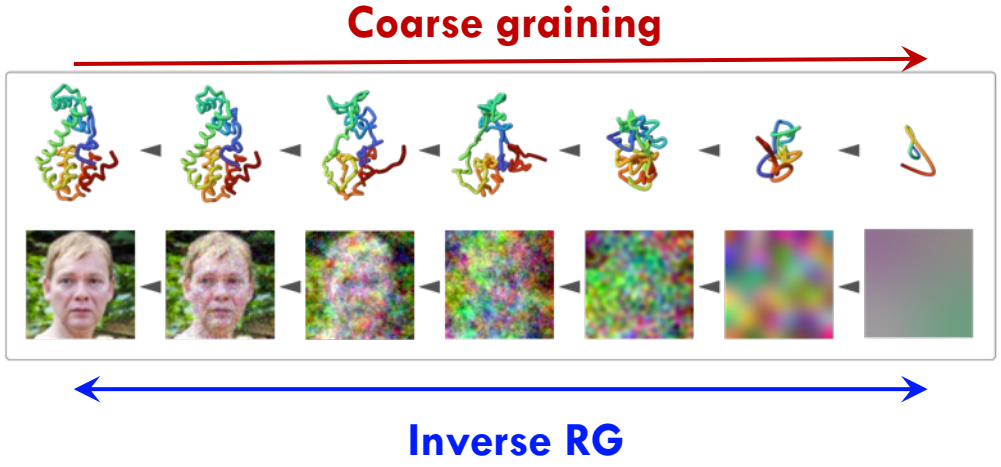
ECT* workshop

Generative models via inverse RG

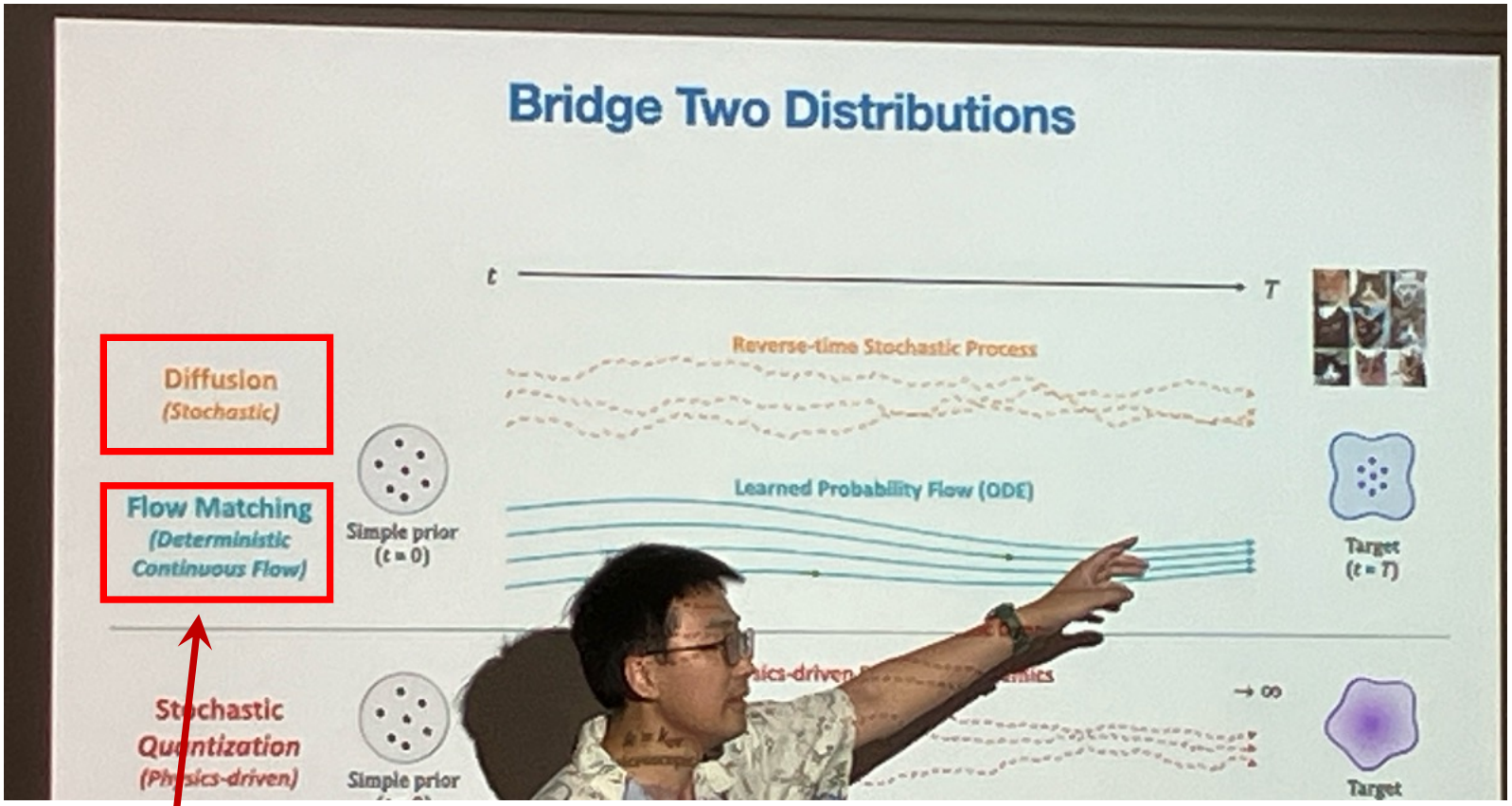
Yuto Ashida (UTokyo)



Kanta Masuki



Based on arXiv 2501.09064 and 2608.23696



RGDM

2501.09064

Generative diffusion model with inverse renormalization group flows

Kanta Masuki[1,*] and Yuto Ashida[1,2,†]

[1]*Department of Physics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan*
[2]*Institute for Physics of Intelligence, The University of Tokyo, 7-3-1 Hongo, Tokyo 113-0033, Japan*

RGFM

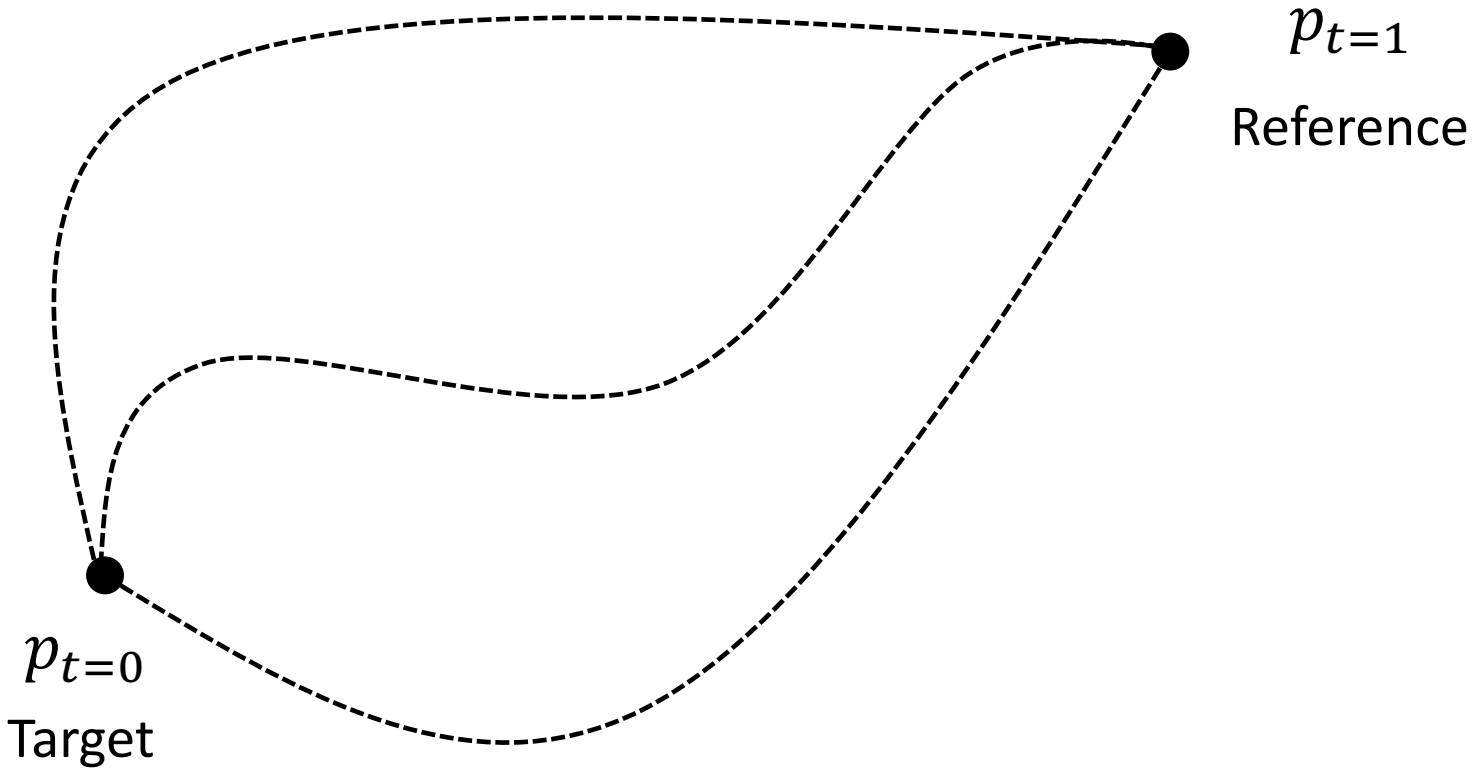
2608.23696

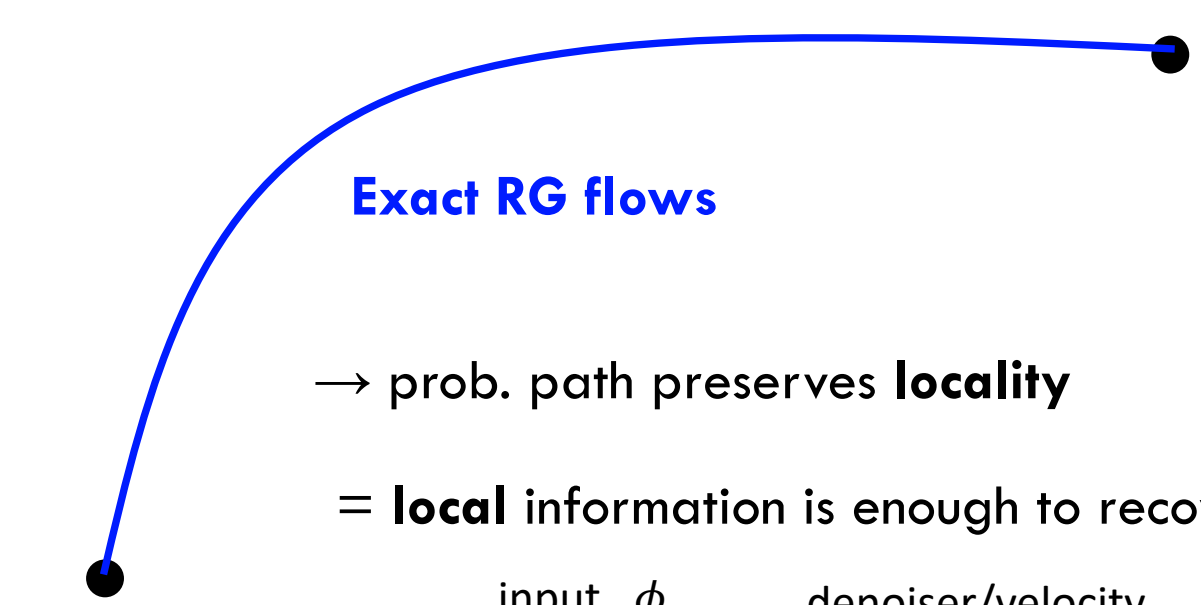
Renormalization Group Flow Matching for Scalable Local Generative Modeling

Kanta Masuki[1,*] and Yuto Ashida[1,2,†]

[1]*Department of Physics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan*
[2]*Institute for Physics of Intelligence, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan*

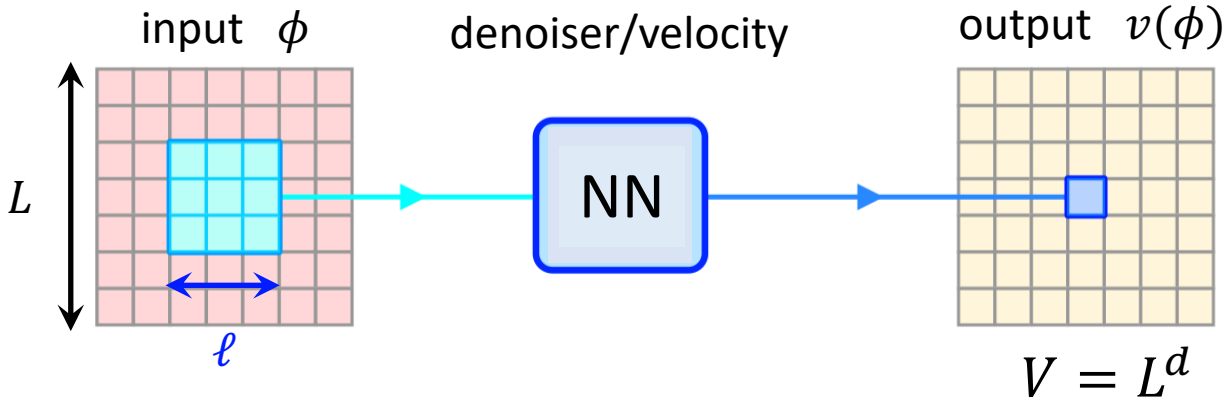
(Dated: August 26, 2026)


$p_{t=1}$
Reference
$p_{t=0}$
Target



→ prob. path preserves **locality**

= **local** information is enough to recover target distribution



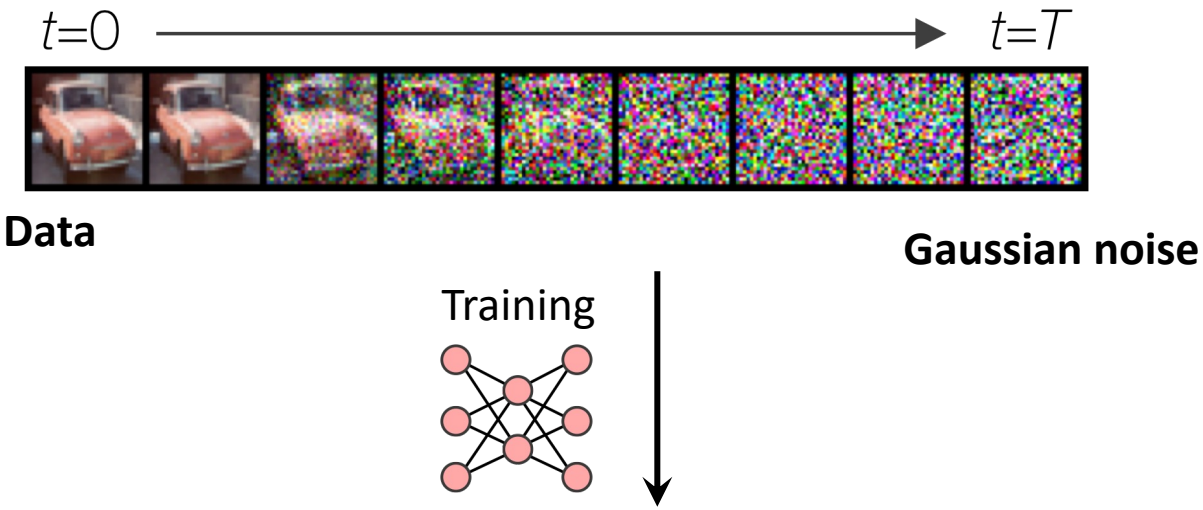
memory costs:
- $\sim V^2 = L^{2d}$ (global)
- $\sim \ell^d V = \ell^d L^d$ (local)

→ **Computationally efficient, scalable sampling by RG**

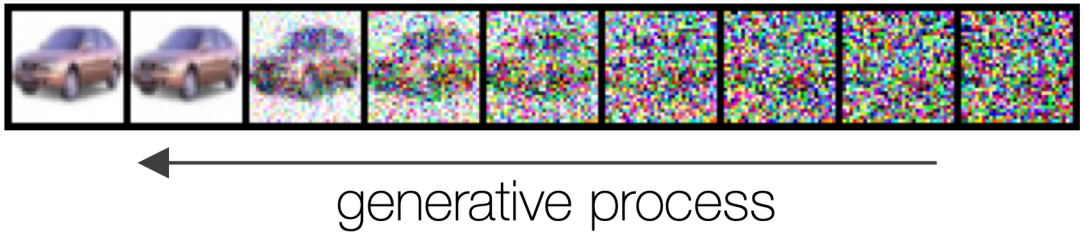
Generative diffusion model

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

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

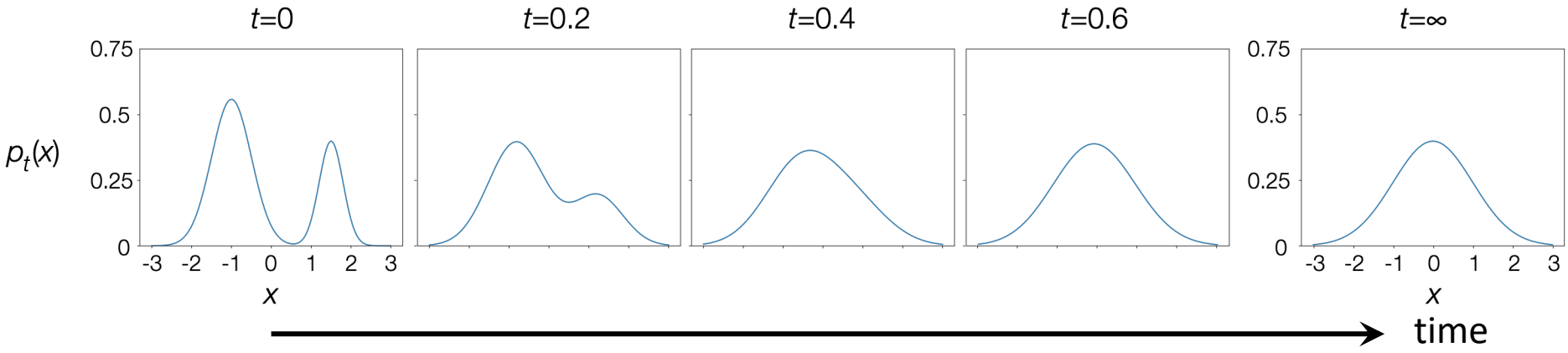
$$\partial_t p_t = \nabla \cdot (\boldsymbol{x} p_t) + \nabla^2 p_t$$

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

Gaussian kernel:

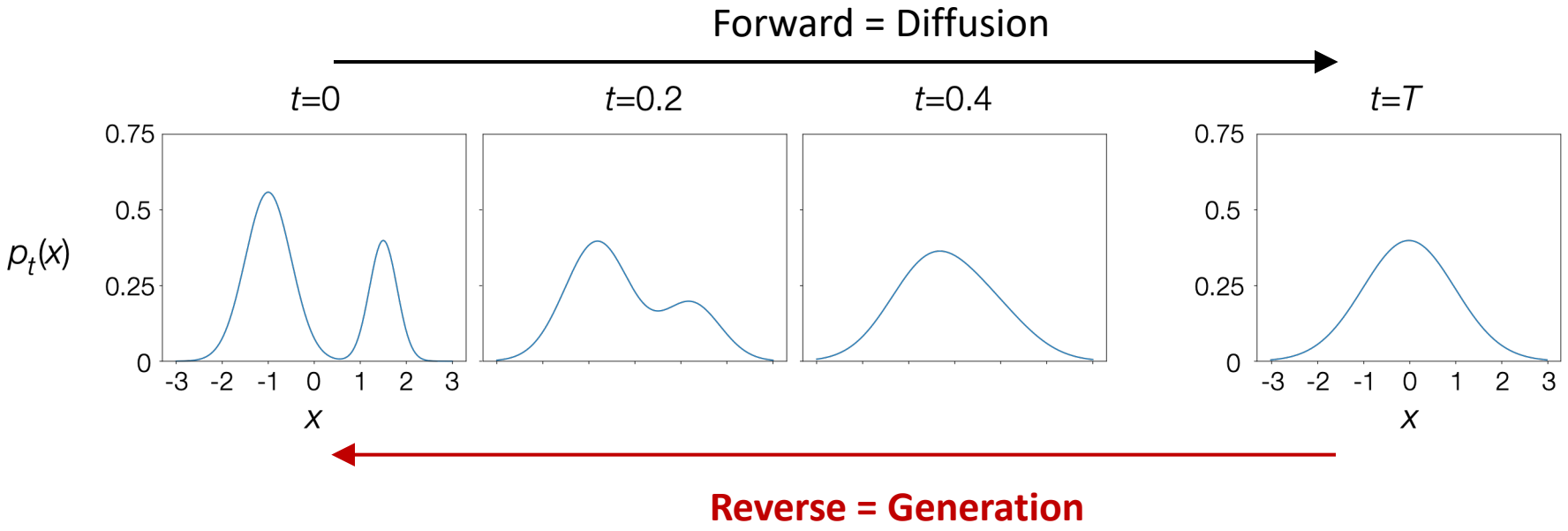
$$p_t(x_t) = \int dx_0\, p(x_t|x_0) p_{t=0}(x_0)$$

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



The **entire** history of p_t contains much more information than the endpoints.

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



Reverse evolution

$$t' = T - t$$

score : a fine-tuned external "force"

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

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

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 $t = 0$.

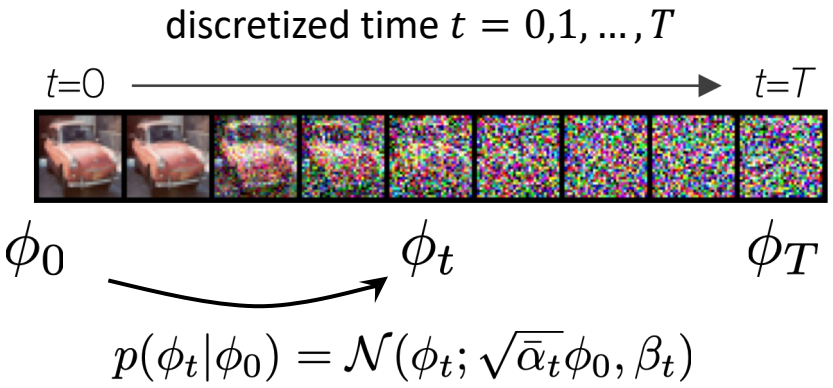
Generative diffusion model

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

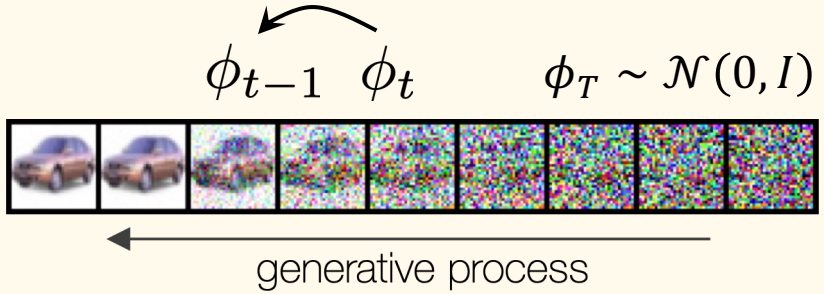


Reverse process (generation):

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

score

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



✗ Highly **sensitive** to a choice of the noise schedule α_t, β_t, which needs to be **heuristically** optimized depending on each data set.

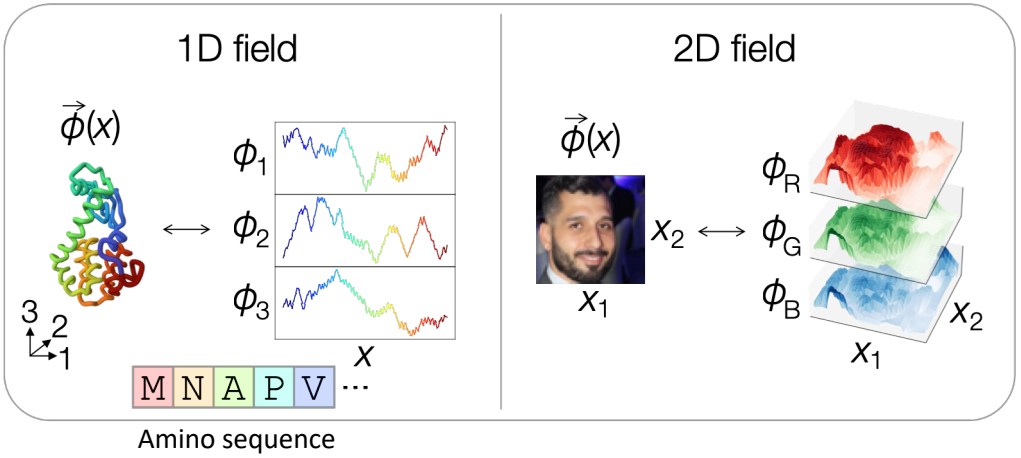
Ho+ '19; Chen '23, Hoogeboom+ '23

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

Use RG to overcome these limitations ?

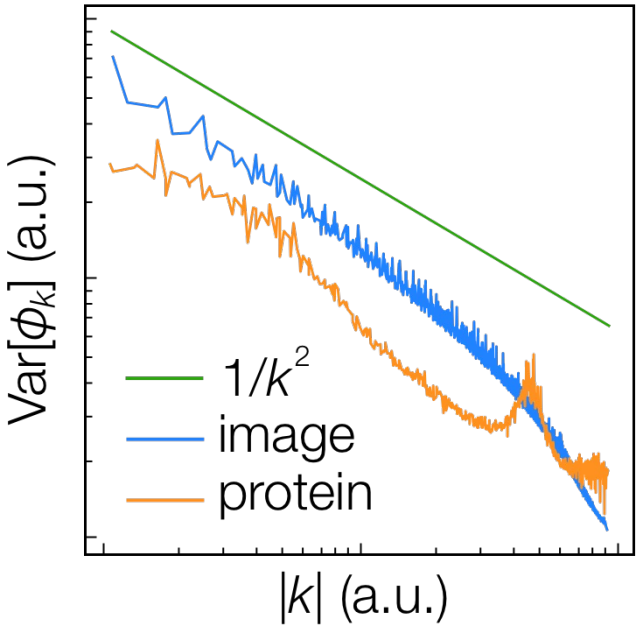
Why RG ? Multiscale Structure Across Diverse Data

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; ϕ : displacements)

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



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

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

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

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

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

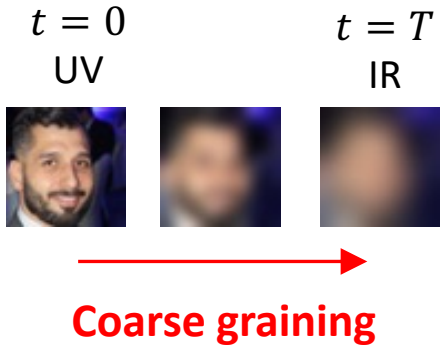
$$\partial_t p_t(\phi) = -\frac{1}{2} \int \frac{d^d k}{(2\pi)^d} \left[\underbrace{k^{-2} \partial_t K_t(k) \frac{\delta^2 p_t}{\delta\phi_k \delta\phi_{-k}}}_{\text{diffusion}} + \underbrace{\frac{\delta}{\delta\phi_k} \left(\frac{\partial_t K_t(k)}{K_t(k)} \phi_k p_t \right)}_{\text{drift}} \right]$$

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

RGDM: Renormalization Group-based Diffusion Model

Forward process (iterative coarse graining)

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



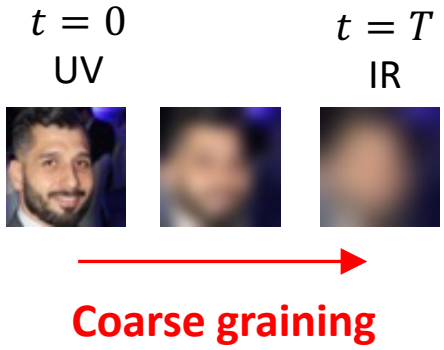
Probability path (**ERG flow eq.**):

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

RGDM: Renormalization Group-based Diffusion Model

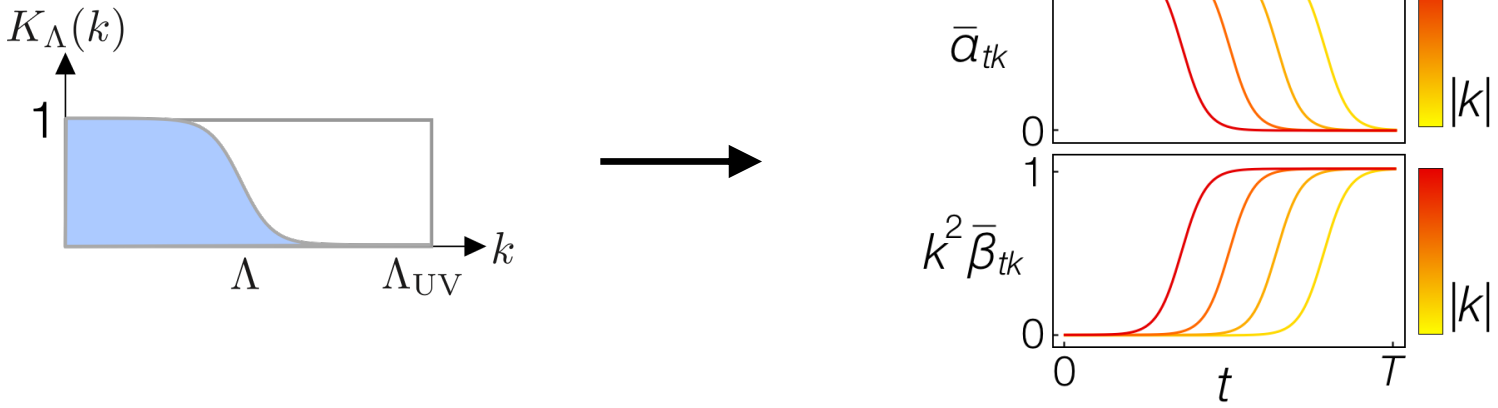
Forward process (iterative coarse graining)

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



✓ ***k*-dependent** noise schedule α_{tk}, β_{tk} → destroys high k modes first (**RG ordering**)

✓ Noise schedules are **unambiguously** fixed once K_Λ is specified
→ giving a guiding principle: (partially) **removing** the need for **heuristic** tuning

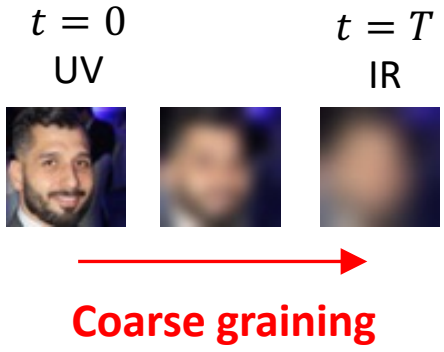


typical choice : $K_t(k) = \frac{r(k^2/\Lambda_t^2)}{1+r(k^2/\Lambda_t^2)}$ with $r(x) = x^{-a}$ or $(e^x-1)^{-1}$ cf. Litim 2000, 2001.

RGDM: Renormalization Group-based Diffusion Model

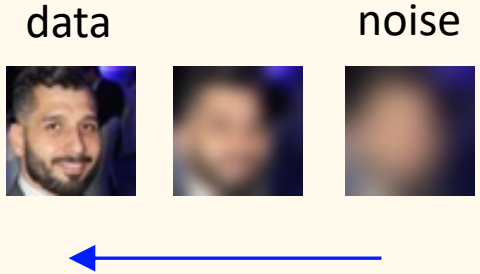
Forward process (iterative coarse graining)

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



Reverse process (inverse RG flow)

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



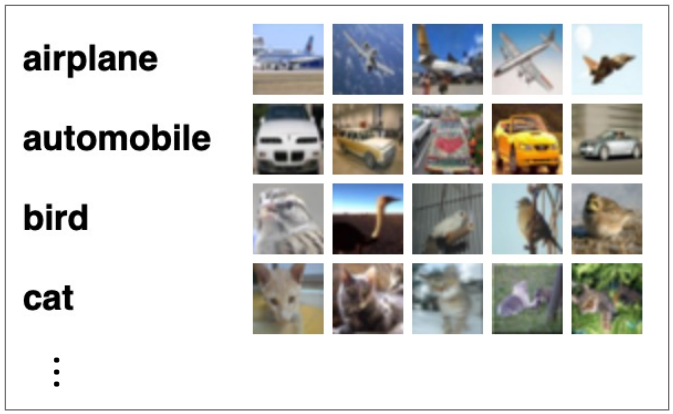
- ✓ Use NN to express k-dependent score $\xi_{\theta k}$, train it from data
- ✓ Generate data in a **coarse-to-fine manner (low-to-high k)**

Masuki and YA, arXiv:2501.09064

Related ideas: talks by Mathis, Anna, Elia …

Application 1 : image generation

CIFAR-10 (32x32)

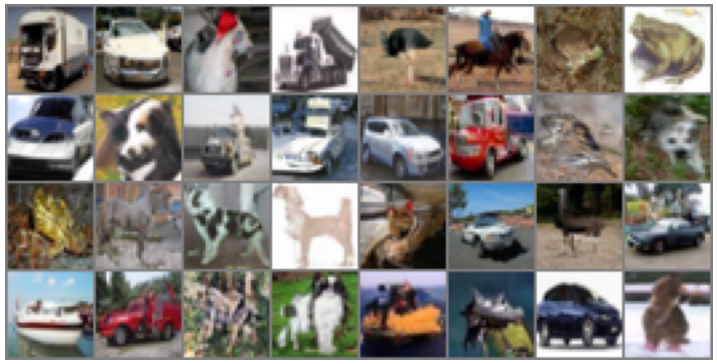


FFHQ (64x64)



Application 1 : image generation

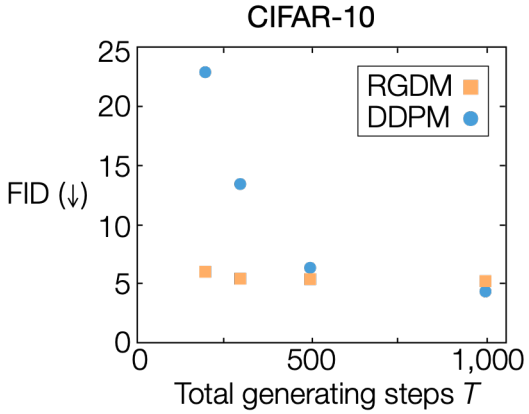
Images generated by RGDM

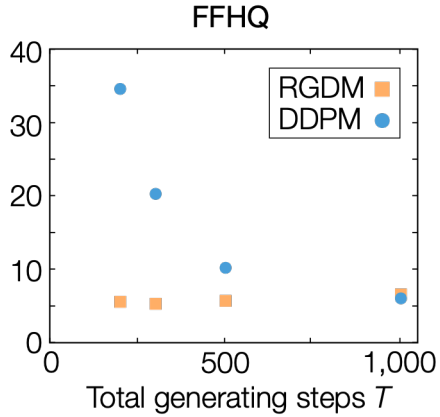




Frechét interception distance = FID ~ indicator for quantifying 'distance' between p_{data} and p_{model}

Comparisons to DDPM

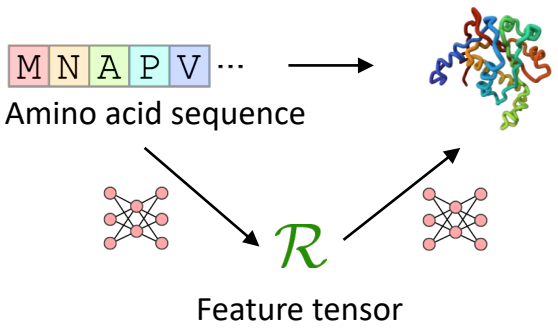




generate 5×10^4 images

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

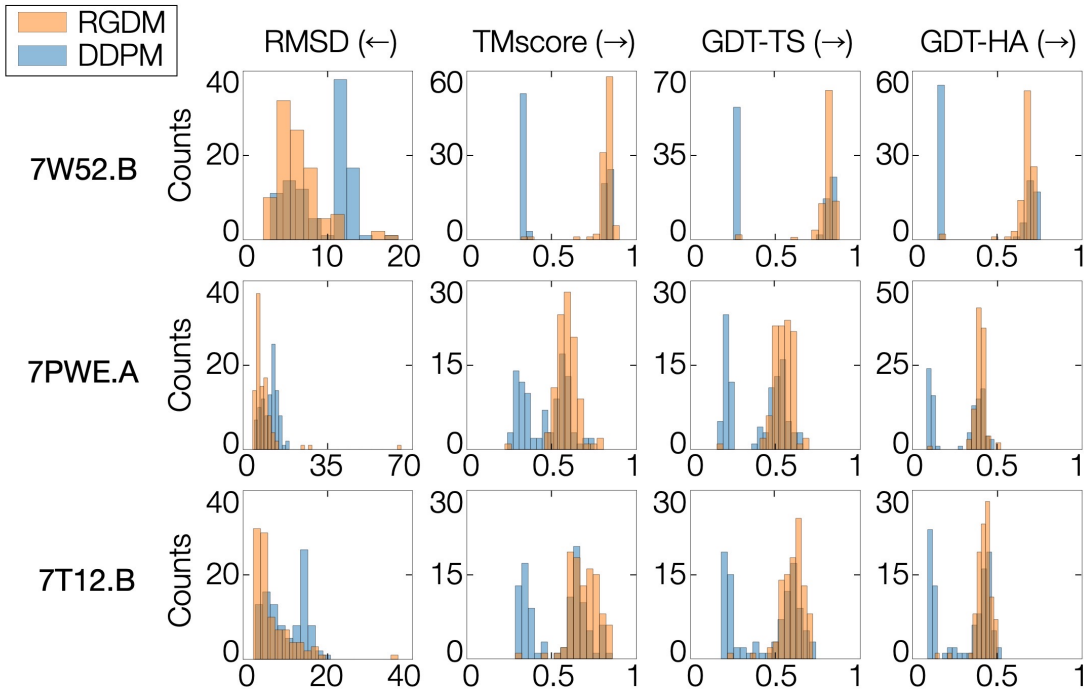
Application 2 : Protein structure prediction

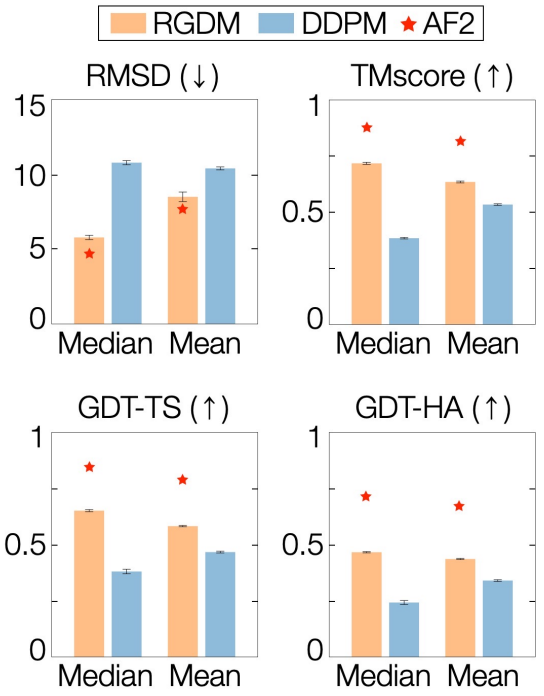


Protein : a chain composed of 20 types of amino acids

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

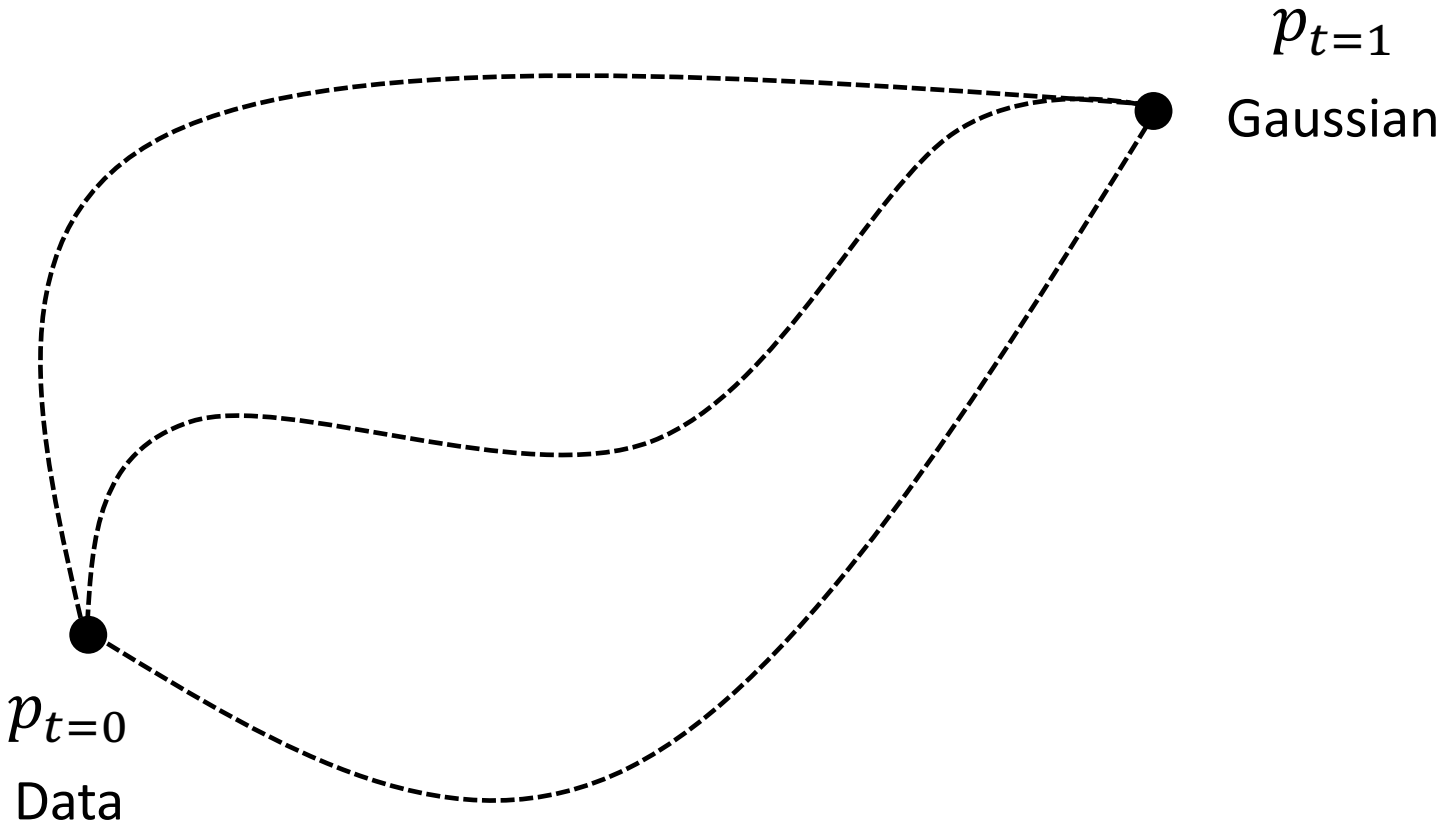
arXiv:2608.23696

Renormalization Group Flow Matching for Scalable Local Generative Modeling

Kanta Masuki[1,*] and Yuto Ashida[1,2,†]

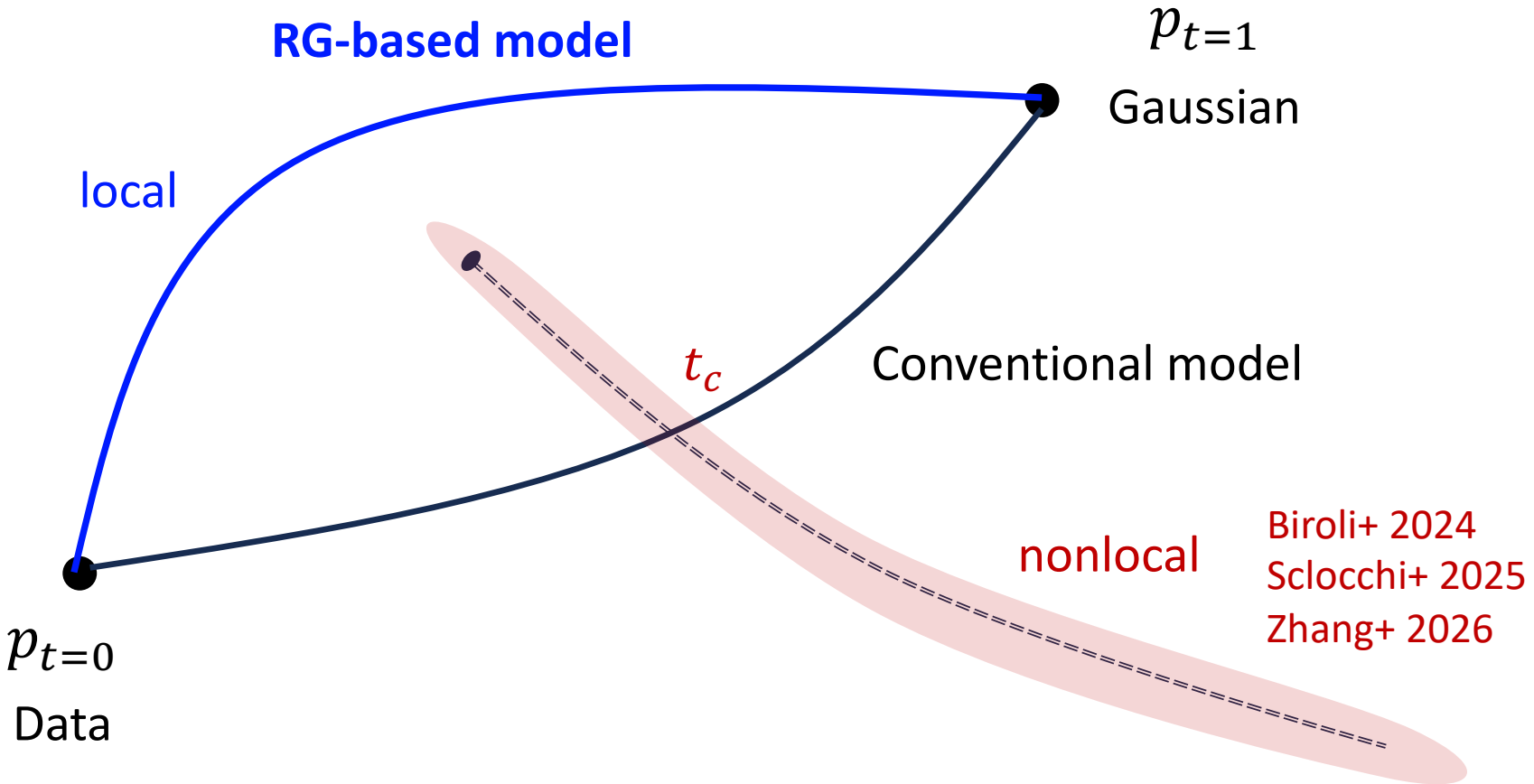
[1]*Department of Physics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan*
[2]*Institute for Physics of Intelligence, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan*
(Dated: August 26, 2026)

Why RG generative model better ? : Locality theorem



Why RG generative model better ? : Locality theorem

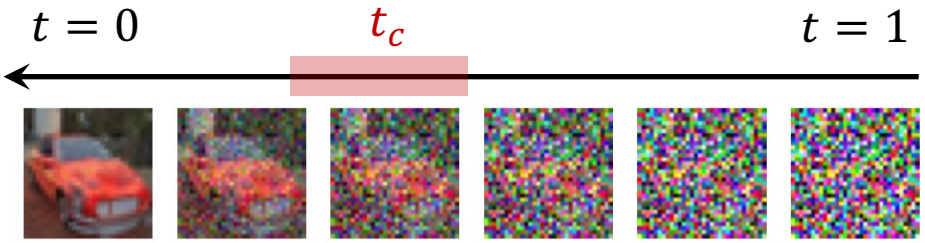
Choose a probability path on which the effective `action' $S \sim -\ln p_t$ remains **local**.



Masuki and YA arXiv:2608.23696

Why RG generative model better ? : Locality theorem

Conventional model



- destroys data at **all** length scales **simultaneously**
- **critical** time regime: **global** information is necessary

  = effective `action' $S \sim -\ln p_t$ becomes **nonlocal** around $t = t_c$

Biroli+ 2024, Sclocchi+ 2025, Zhang+ 2026

RG-based model



- **no** critical regime = **continuously** connects $p_{t=0}$ (data) and $p_{t=1}$ (Gaussian).
- **local** information is enough to recover data distribution

  = effective `action' $S \sim -\ln p_t$ **satisfies quasilocality**

Masuki and YA arXiv:2608.23696

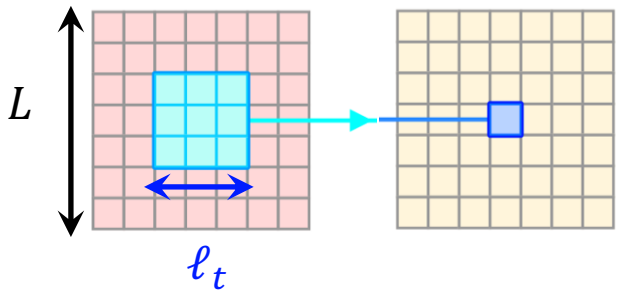
Why RG generative model better ? : Locality theorem

Local denoiser is enough to recover data distribution in RG-based model.

Locality theorem (informal) :

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

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



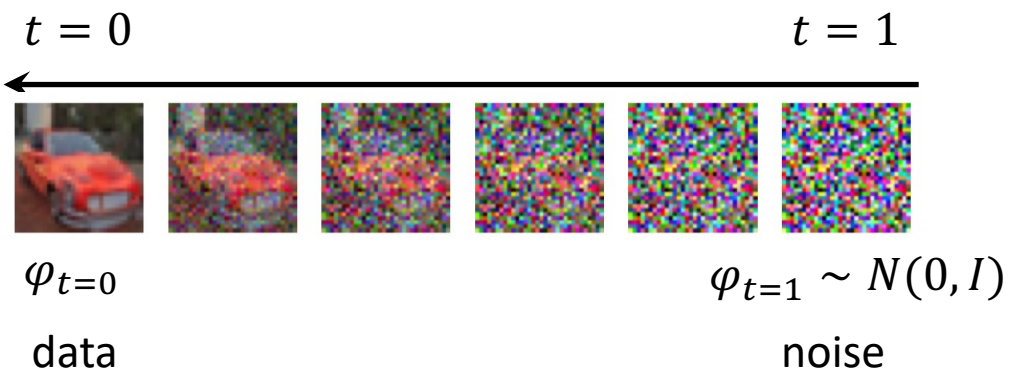
where L is size of a sample, Λ_t is a running momentum cutoff, and the 2-Wasserstein distance is bounded as

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

*conventional model: ℓ_t can be $O(L)$ [= **global** denoiser is necessary].

Masuki and YA arXiv:2608.23696

Flow matching : deterministic probability transport



Flow matching (FM)

ODE update of sample: $\frac{d\varphi_t}{dt} = v_\theta(\varphi_t, t)$ (velocity)

- ✓ continuous time
- ✓ deterministic

Conventional FM

Probability path (Gaussian kernel):

$$p_t(\varphi) = \int d\varphi_0 \, p_t(\varphi|\varphi_0) p_0(\varphi_0)$$

$$p_t(\varphi|\varphi_0) = N(\varphi; (1-t)\,\varphi_0, t^2 I)$$

Velocity:

$$v_\theta = -\frac{1}{1-t}\varphi - \frac{t}{1-t}\frac{\delta}{\delta\varphi}\ln p_t$$

local — nonlocal (score)

Zhang+ 2026

→ global info is necessary

RG Flow Matching: local probability transport



Flow matching (FM)

ODE update of sample: $\frac{d\varphi_t}{dt} = v_\theta(\varphi_t, t)$ (velocity)

- ✓ continuous time
- ✓ deterministic

RGFM

Probability path (**ERG flow eq.**):

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

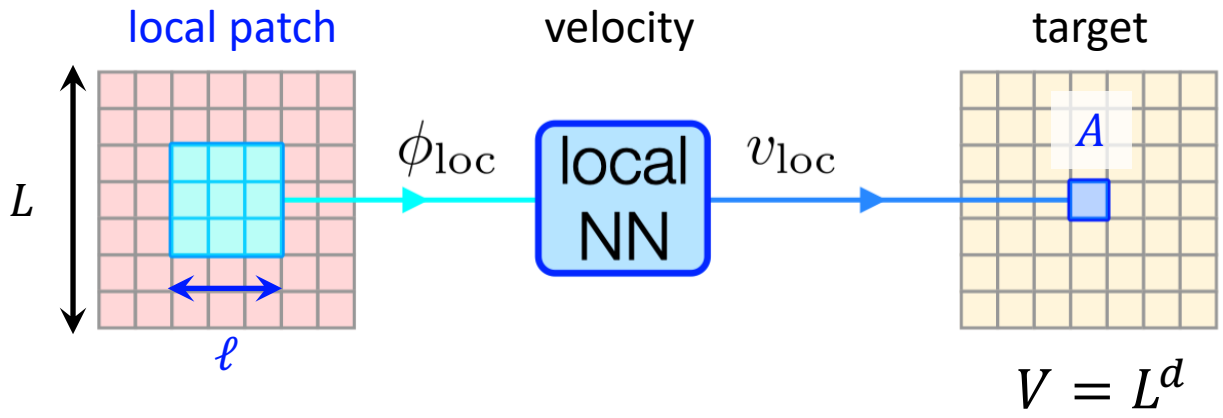
Velocity:

$$v_\theta = \frac{1}{\Lambda_t^2 t}\frac{\delta V_{\Lambda_t}}{\delta\phi}$$

quasilocal → local info is enough

$$S_{\Lambda_t}(\phi) = \frac{1}{2}\int_k \left(k^2 + m^2\right)|\phi_k|^2 + V_{\Lambda_t}(\phi) \qquad p_{\Lambda_t}(\phi) \propto e^{-S_{\Lambda_t}(\phi)}$$

Local Flow Matching



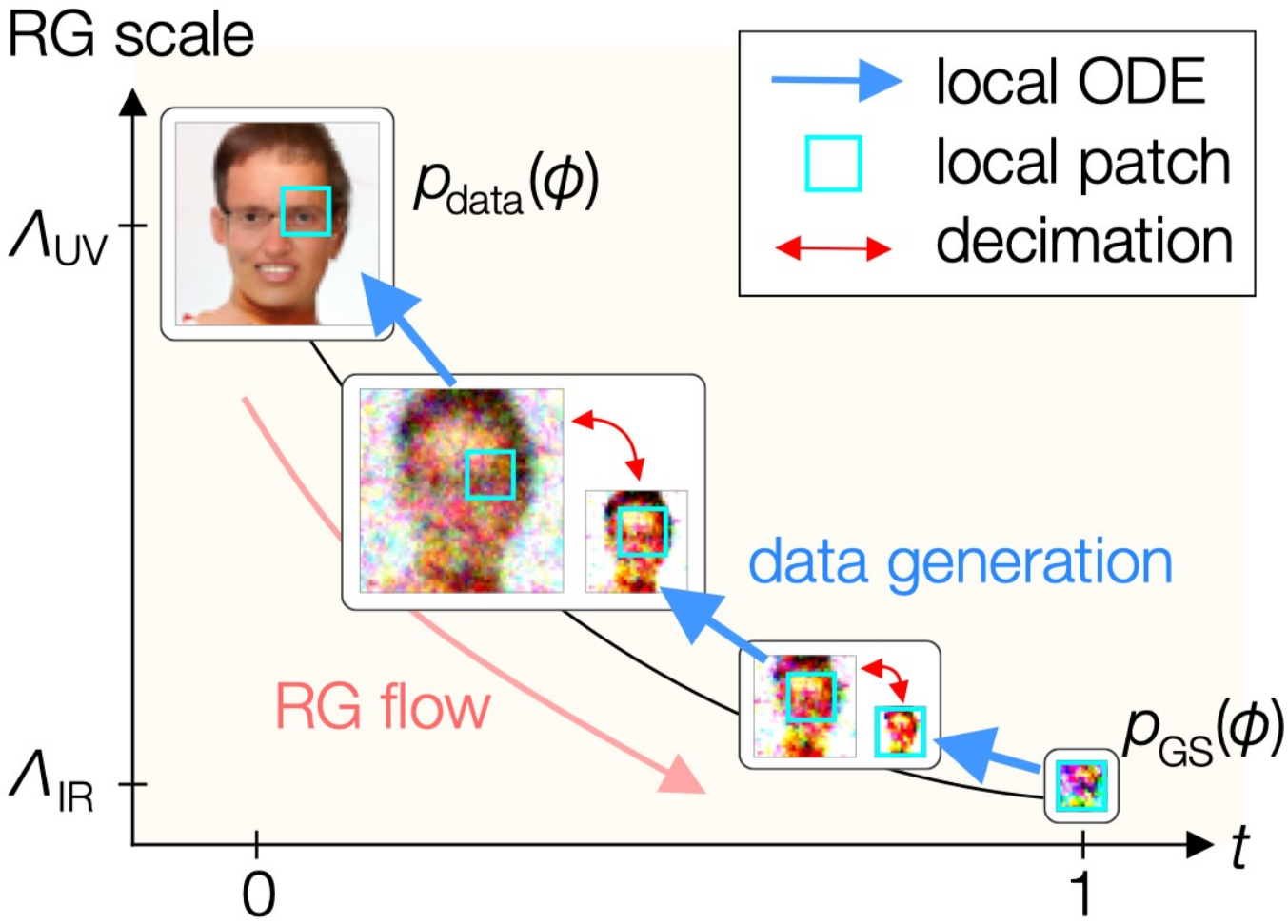
Local approximation of velocity: $v^A(\phi, t) \approx v^A_{\text{loc}}(\phi_{\text{loc}}, t)$

Local FM: $\dfrac{d\phi_A}{dt} = v^A_{\text{loc}}(\phi_{\text{loc}}, t)$

memory costs:
$\sim V^2 = L^{2d}$ (global)
$\sim \ell^d V = \ell^d L^d$ (local)

RG Local Flow Matching

Local ODE: $\frac{d\phi}{dt} = v_{\mathrm{loc}}(\phi_{\mathrm{loc}}, t)$ **RG choice:** $v_{\mathrm{loc}} = \frac{1}{\Lambda_t^2 t}\frac{\delta V_{\Lambda_t}}{\delta \phi}$



Generate a sample by using only local information

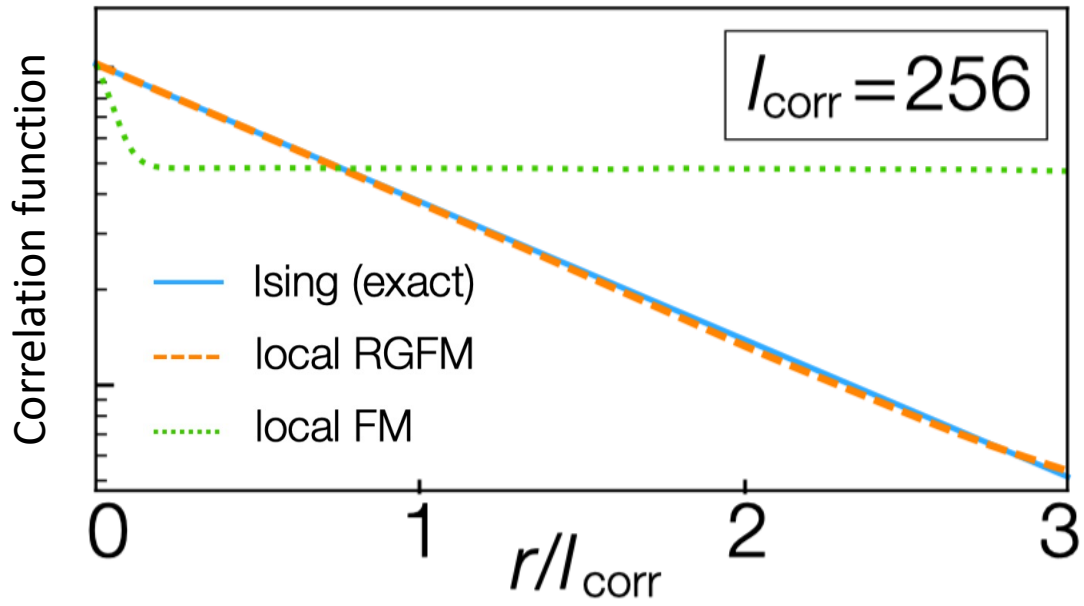
Numerical demonstration: 1D Ising model



local patch
(25 lattice sites)

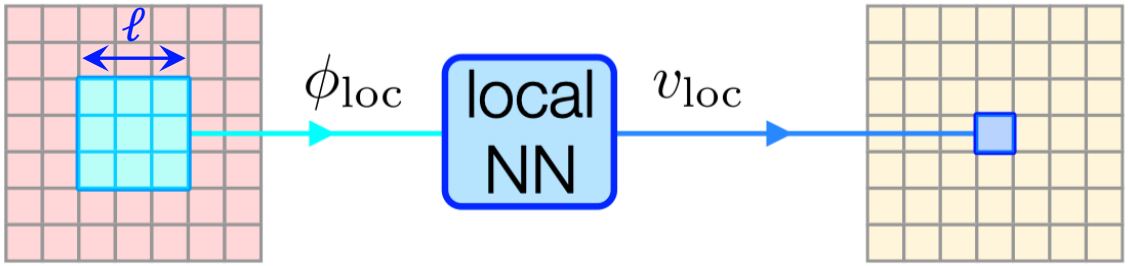
$$v^A(\phi,t) \approx v^A_{\rm loc}(\phi_{\rm loc},t)$$

local ODE: $\dfrac{d\phi_A}{dt} = v^A_{\rm loc}(\phi_{\rm loc},t)$



RGFM: local velocity is enough to recover data distribution

Numerical demonstration: 2D image

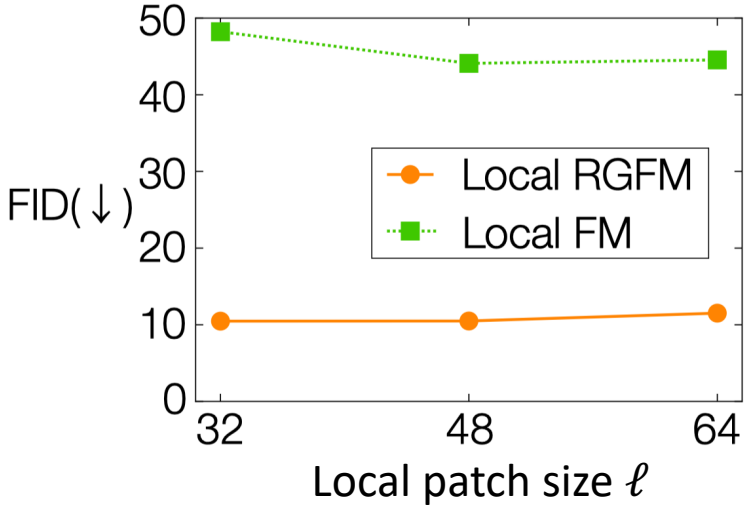


local RGFM (L=64, 256)



local FM (L=64, 256)





Computational time of 256×256

Global FM: $O(10^2)$ GPU days training

Rombach+ 2022

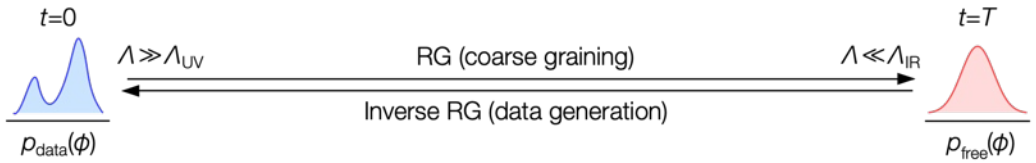
Local RGFM: 6 GPU days training

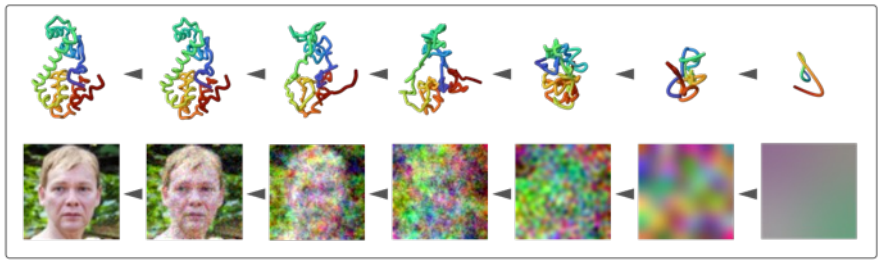
[on a single RTX6000 GPU]

Summary

- Proposed DM/FM based on the exact RG.
- The RG models can generate data only by using local information.
- Numerical experiments suggest the possibility of improving sample efficiency through the RG.

All the codes available in Github; please try!





Masuki and YA,
arXiv:2501.09064 and 2608.23696

