Sony AI

Enhancing Accuracy and Efficiency in Diffusion Models

Sony AI @ Tokyo

Chieh-Hsin (Jesse) LAI

Series Talks in 2024

My Collaborators

Sony AI

Stanford University

Chieh-Hsin (Jesse) Lai

- Researcher/

  DGM Tech Lead @ Sony AI

- Ph.D. Math, U. of MN

Dongjun Kim

- Postdoc @ Stanford
- Intern @ Sony AI
- Ph.D. IE, KAIST

Stefano Ermon

- Prof. @ Stanford

WeiHsiang Liao

Murata Naoki

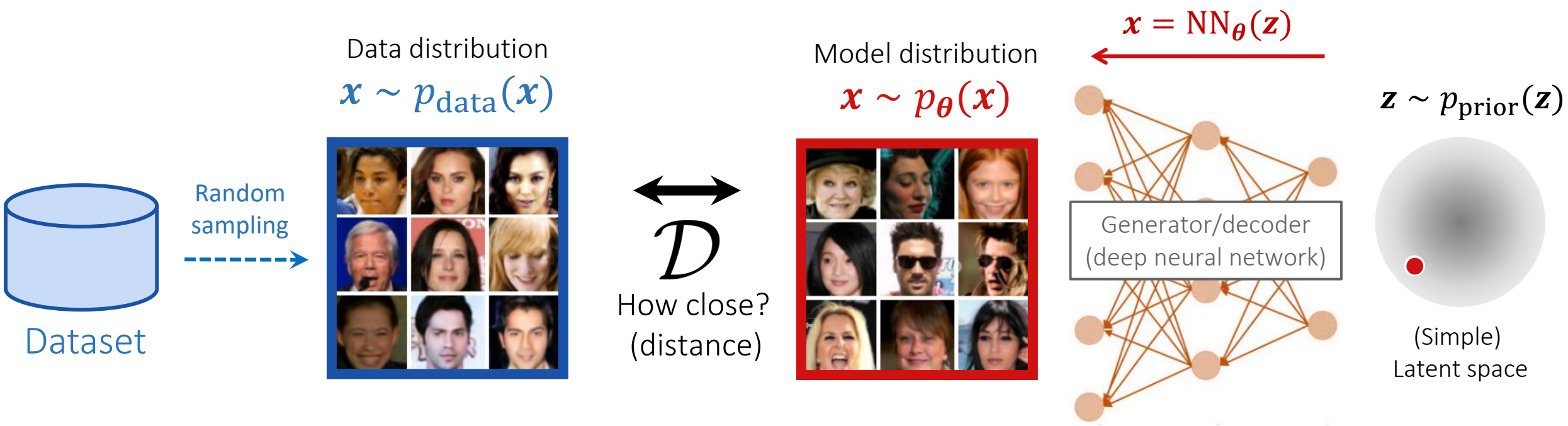
Yuhta Takida

Toshimitsu Uesaka

Yuki Mitsufuji

Deep Generative Model

Goal: use NN to learn target distribution explicitly/implicitly

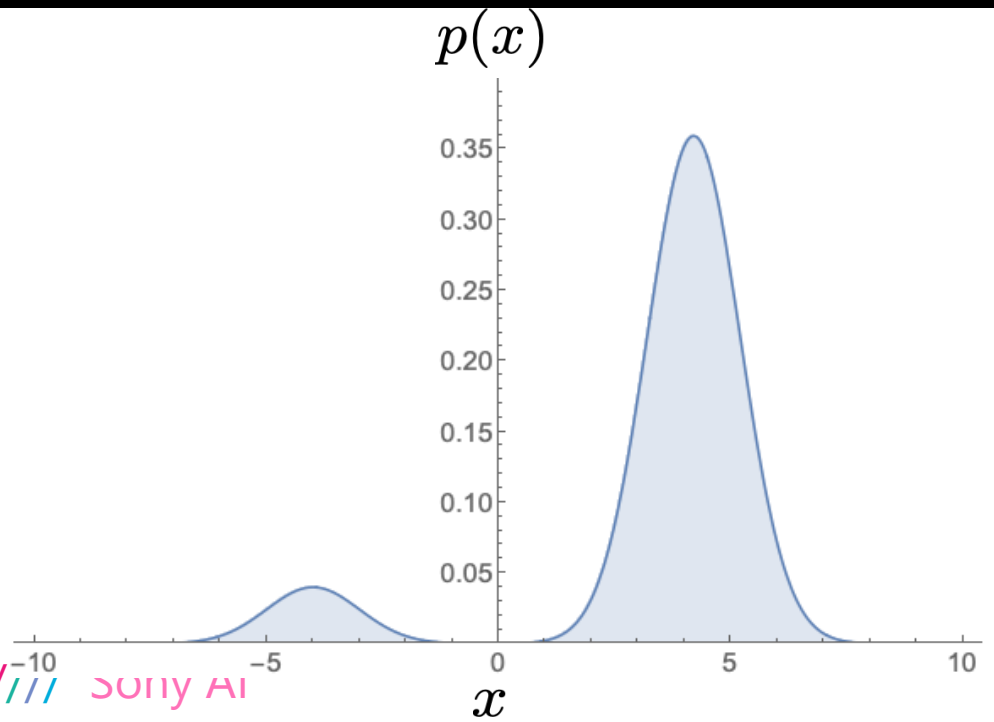


$$\min_\theta \mathcal{D}(p_\theta(x), p_{\text{data}}(x))$$

WHAT IS DIFFUSION MODELS?

Using “Score” (Slope) for Generation

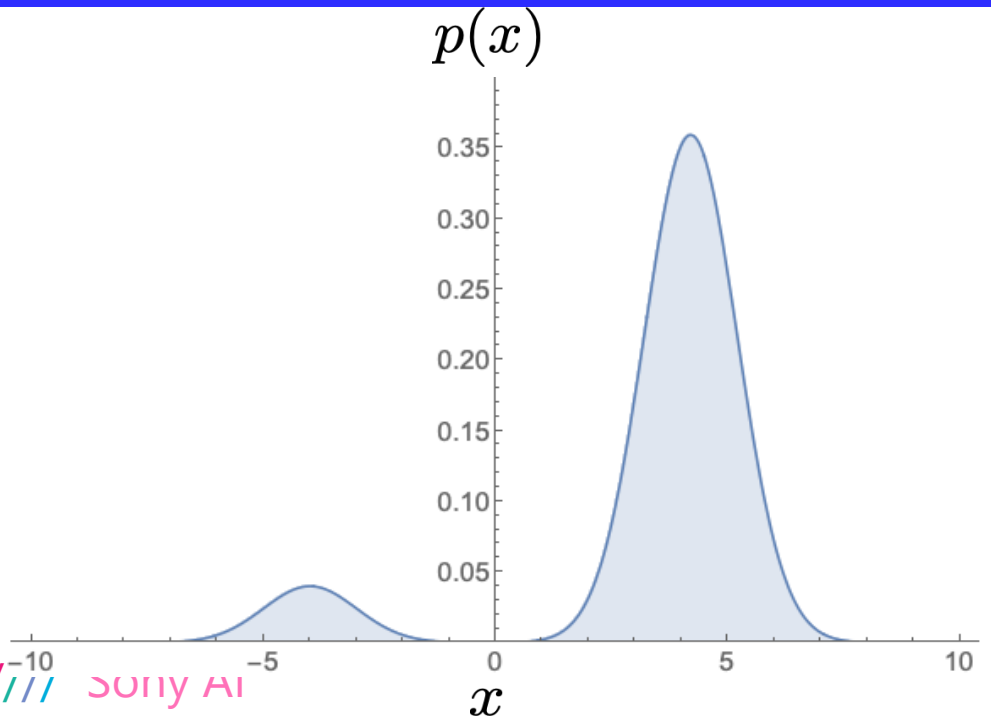
$$p_{\text{data}}(\boldsymbol{x})$$

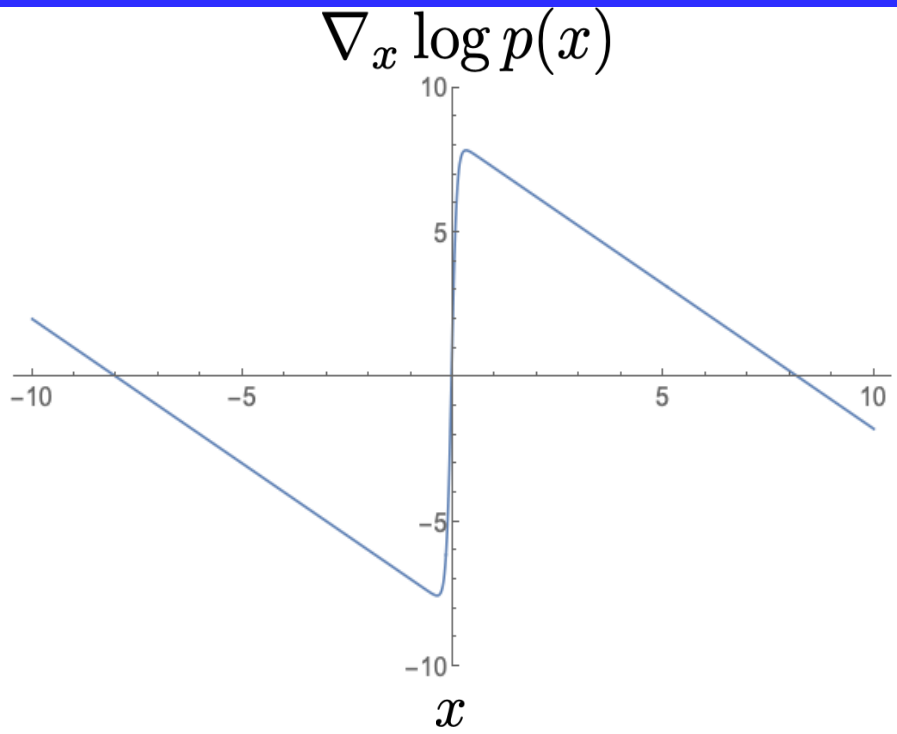


Using “Score” (Slope) for Generation

Moving toward more likely direction…

$$\nabla_x \log p_{\text{data}}(\boldsymbol{x})$$





Using “Score” (Slope) for Generation

Moving toward more likely direction…

$$\nabla_x \log p_{\text{data}}(x)$$

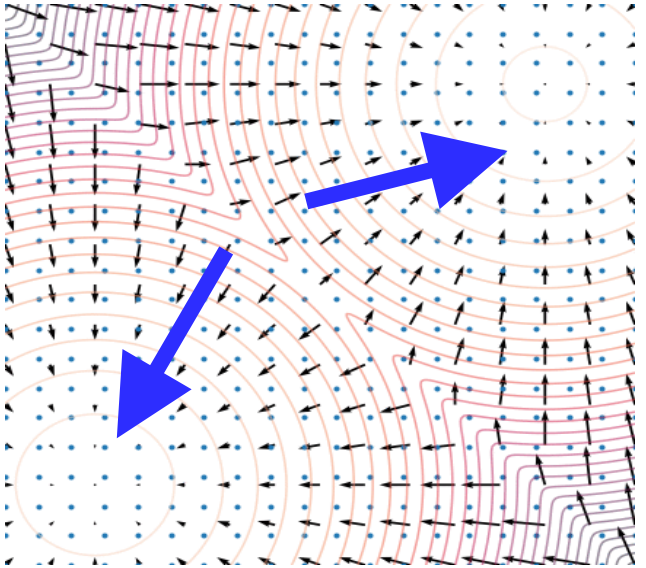
Phase I. Score Matching
Phase II. Denoising Score Matching
Phase III. Noised Conditioned Denoising Score Matching

Phase IV. Score-Based Continuous Time Diffusion Model

Phase I (2005). Score Matching

Score function:

$$\nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})$$

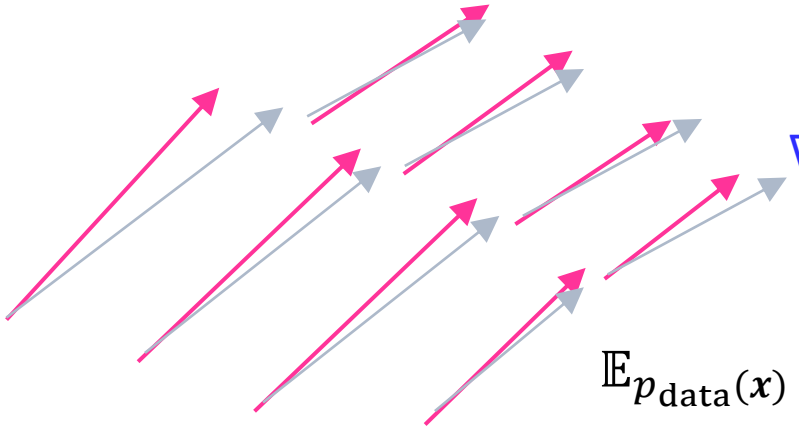


[Generation] Langevin sampling:

$$\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t + \eta \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x}) + \sqrt{2}\,\boldsymbol{\epsilon}_t, \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

$$\approx \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})$$

[Training] Score Matching:



$\nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})$

$$\mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\left[\frac{1}{2}\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log p_{\text{data}}(\boldsymbol{x})\|_2^2\right] = \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}\left[\frac{1}{2}\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})\|_2^2 + \text{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}))\right] + C$$

[Hyvärinen '05]

Phase II (2011). Denoising Score Matching

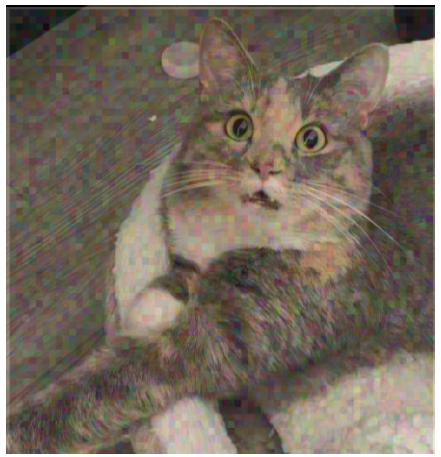
[Training] Denoising Score Matching

$$\mathbb{E}_{p(\boldsymbol{x}_{\text{perturb}}|\boldsymbol{x})}\left[\frac{1}{2}\left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})-\left(\frac{\boldsymbol{x}-\boldsymbol{x}_{\text{perturb}}}{\sigma^2}\right)\right\|_2^2\right]$$

$\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})$

+ **Minor Noise** $\mathcal{N}(0,\sigma^2 I)$

$p(\boldsymbol{x}_{\text{perturb}}|\boldsymbol{x})$



$$\nabla \log \mathcal{N}(\boldsymbol{x}_{\text{perturb}}; \boldsymbol{x}, \sigma^2 \mathrm{I})$$

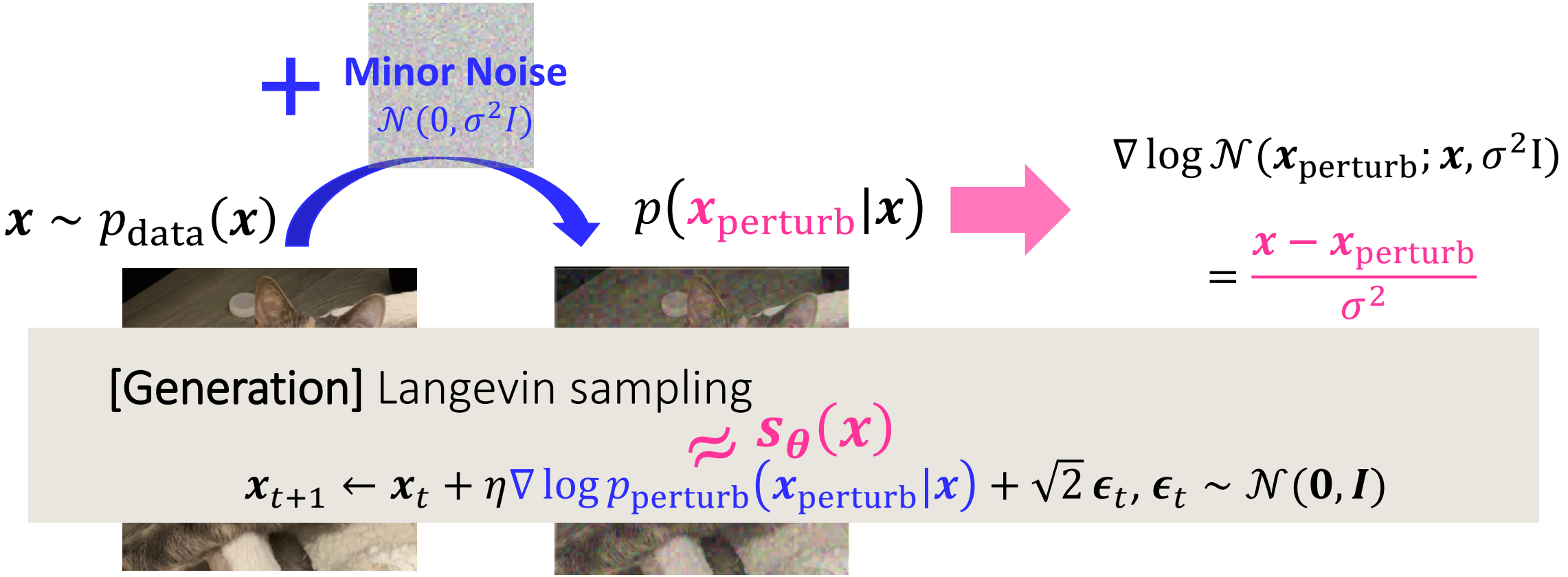
$$= \frac{\boldsymbol{x}-\boldsymbol{x}_{\text{perturb}}}{\sigma^2}$$

[Vincent '11]

Phase II (2011). Denoising Score Matching

[Training] Denoising Score Matching

$$\mathbb{E}_{p(\boldsymbol{x}_{\text{perturb}}|\boldsymbol{x})}\left[\frac{1}{2}\left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})-\left(\frac{\boldsymbol{x}-\boldsymbol{x}_{\text{perturb}}}{\sigma^2}\right)\right\|_2^2\right]$$



[Generation] Langevin sampling

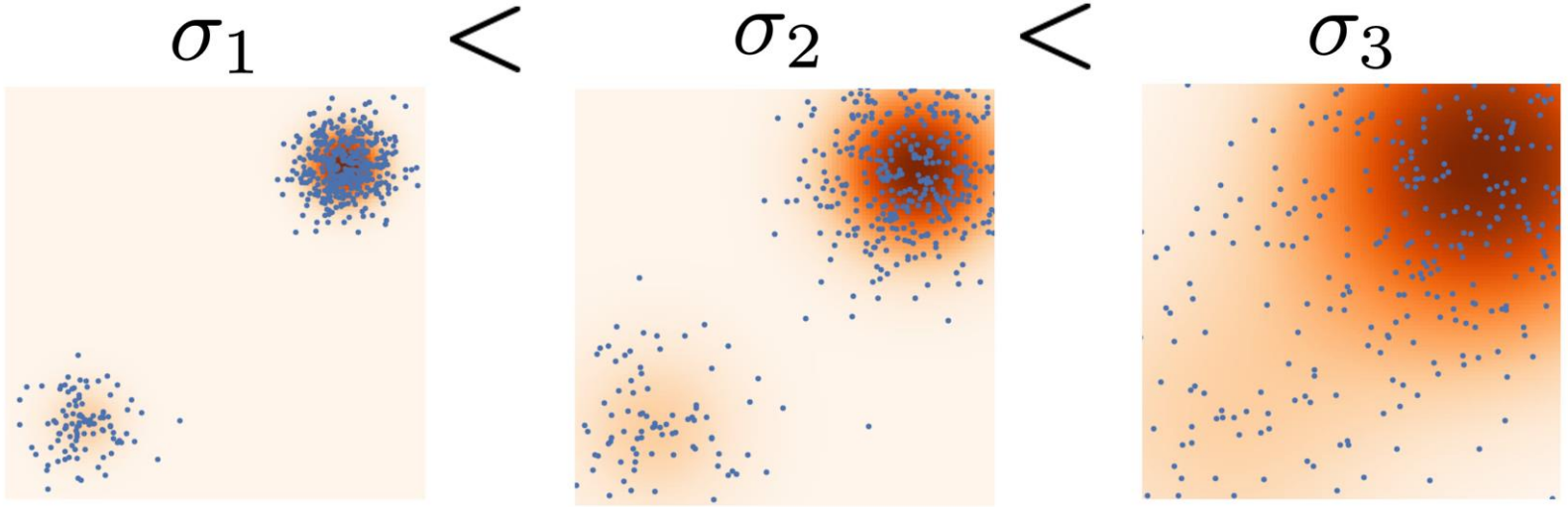
$$\boldsymbol{x}_{t+1}\leftarrow \boldsymbol{x}_t+\eta\nabla\log p_{\text{perturb}}(\boldsymbol{x}_{\text{perturb}}|\boldsymbol{x})+\sqrt{2}\,\boldsymbol{\epsilon}_t,\ \boldsymbol{\epsilon}_t\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I})$$

$$\approx \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x})$$

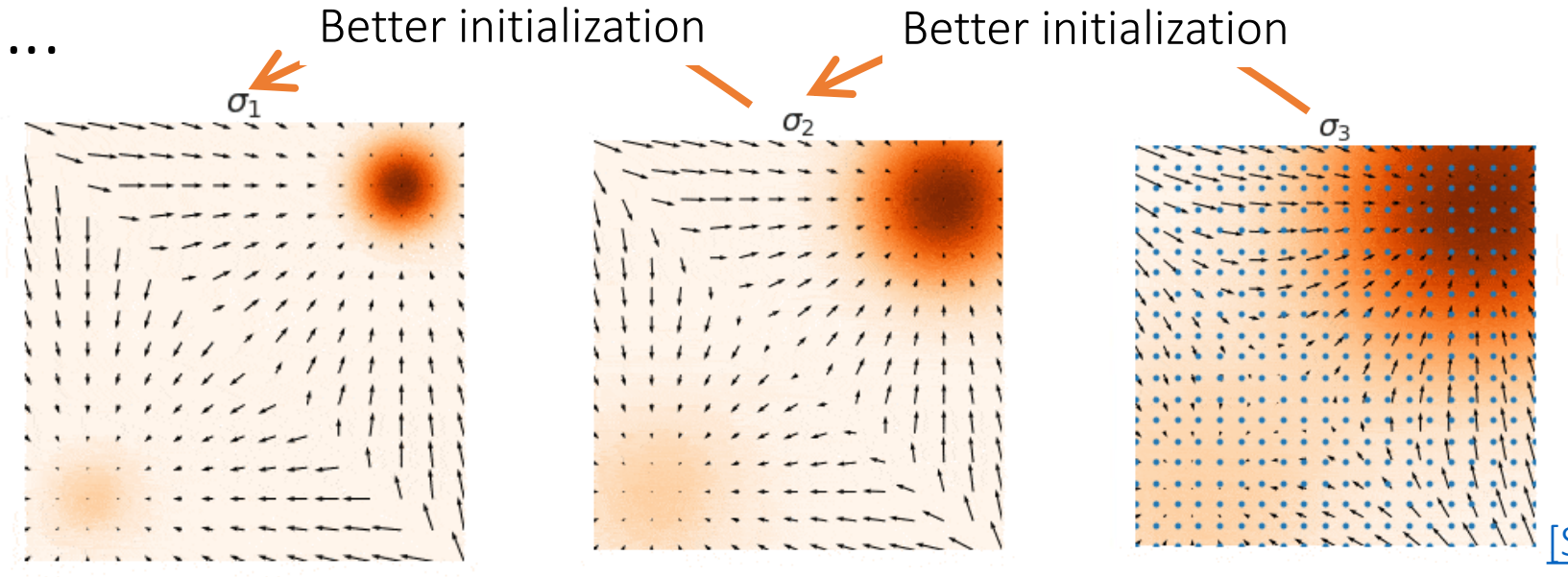
[Vincent '11]

Phase III (2019). Noised Conditioned Denoising Score Matching

Injecting incremental multiple levels of noises to data



In generation….

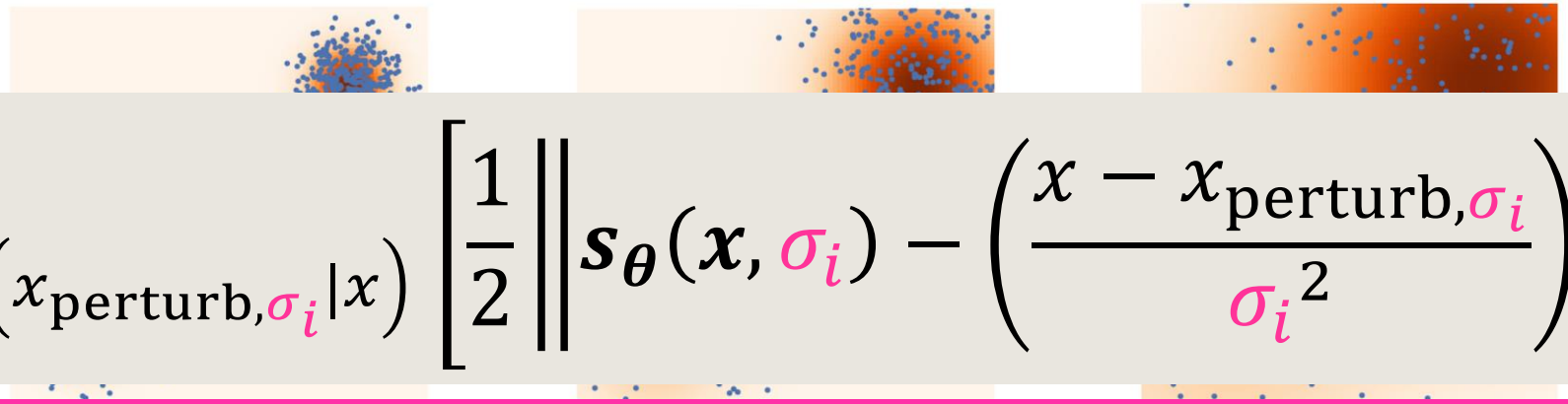


[Song+ NeurIPS'19]

Phase III (2019). Noised Conditioned Denoising Score Matching

Injecting incremental multiple levels of noises to data

$$\sigma_1 < \sigma_2 < \sigma_3$$

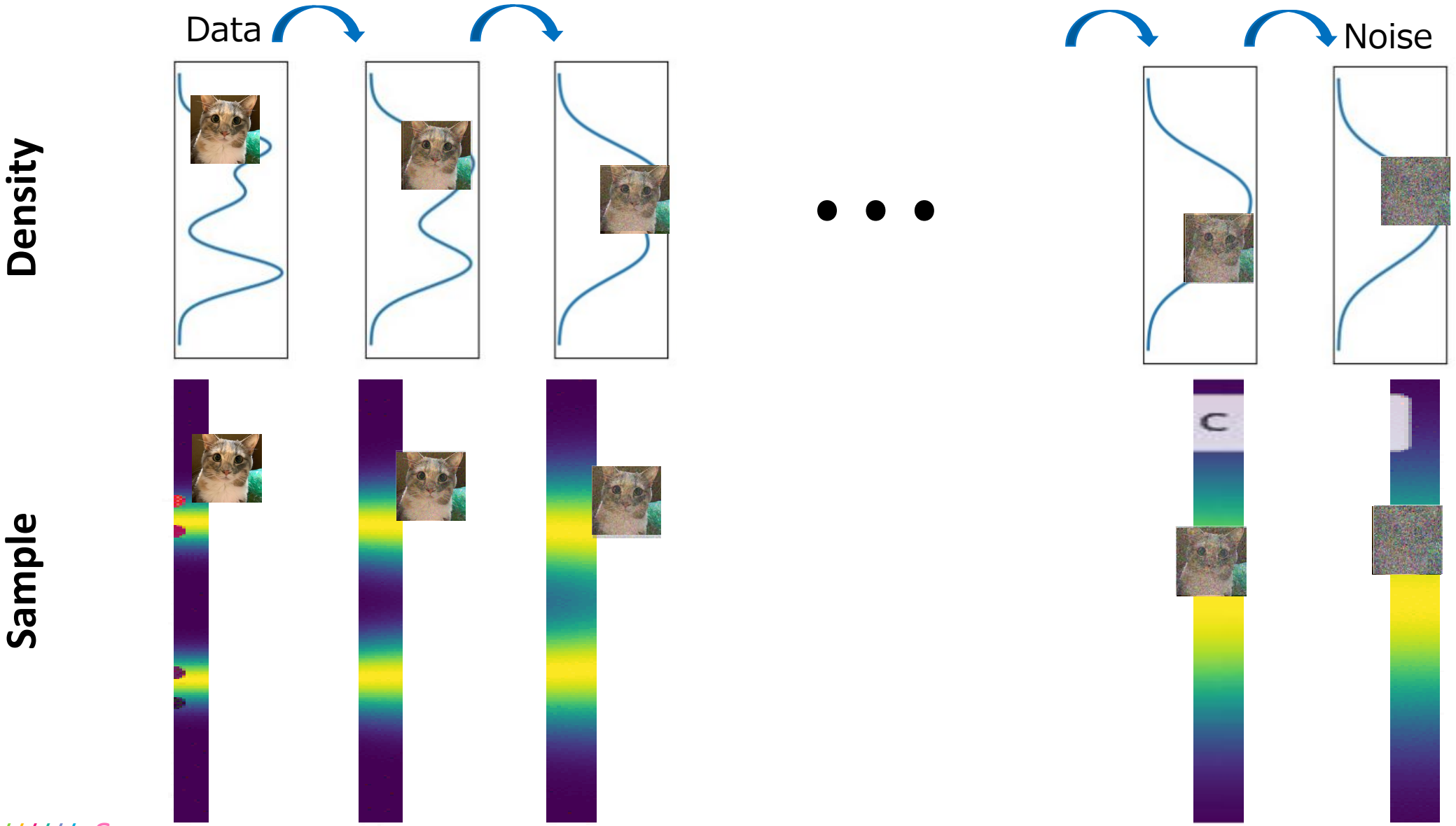


$$\sum_{i=1}^{3} \omega_i \, \mathbb{E}_{p\left(x_{\text{perturb},\sigma_i} | x\right)} \left[\frac{1}{2} \left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, \sigma_i) - \left(\frac{x - x_{\text{perturb},\sigma_i}}{{\sigma_i}^2} \right) \right\|_2^2 \right]$$

The vibe of Diffusion Model appeared…

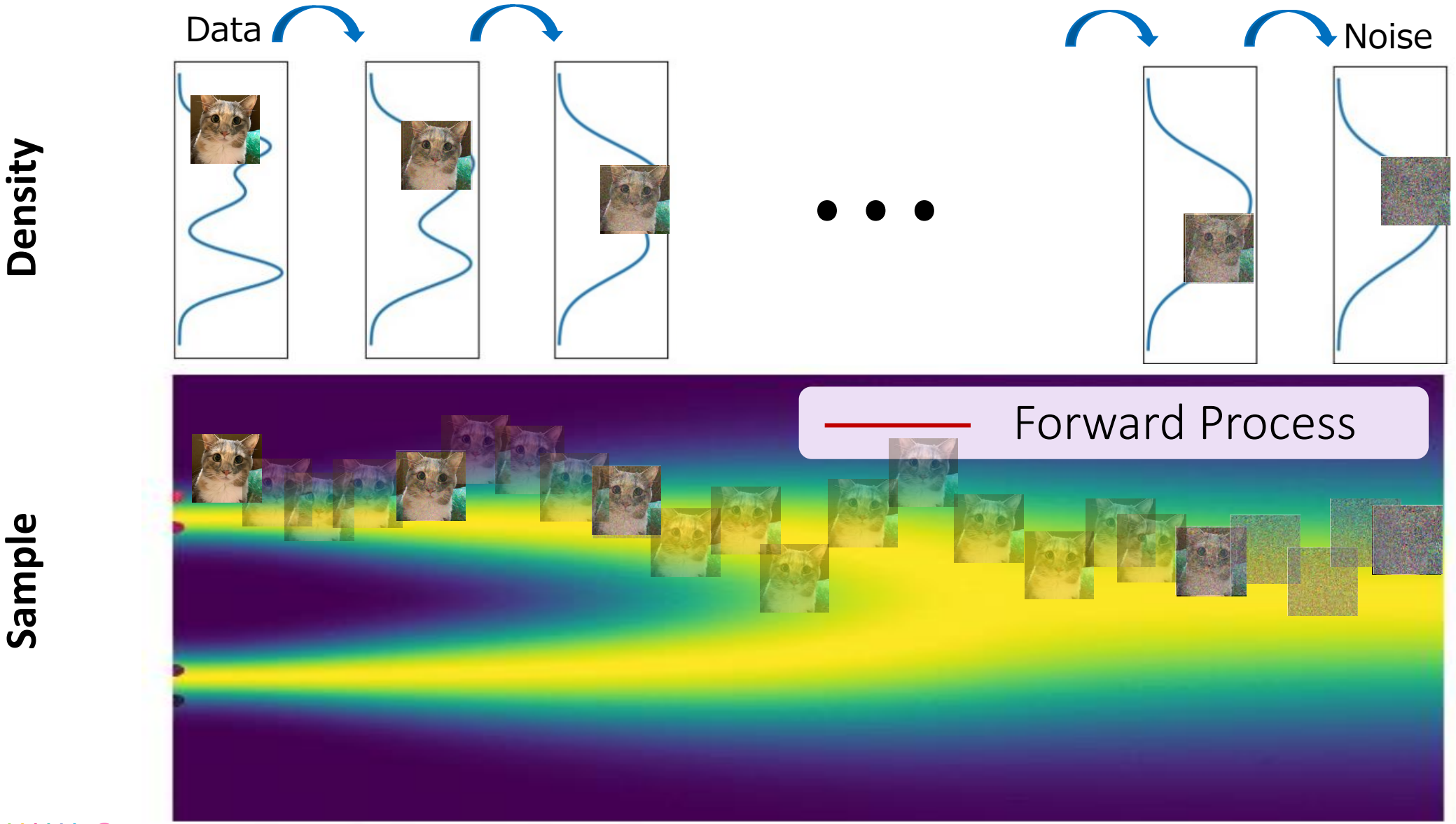
[Song+ NeurIPS'19]

Phase IV (2020). Score-Based Continuous Time Diffusion Model



[Song+ ICLR'21]

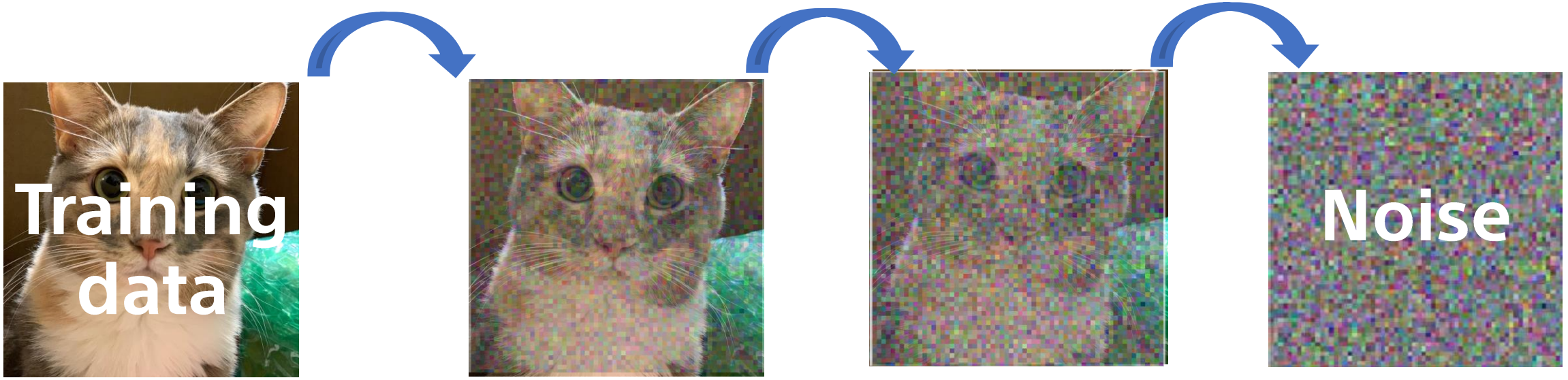
Phase IV (2020). Score-Based Continuous Time Diffusion Model



[Song+ ICLR'21]

Diffusion Model at High Level

Forward Process: Training data → white noise (simple space)



Diffusion Model at High Level

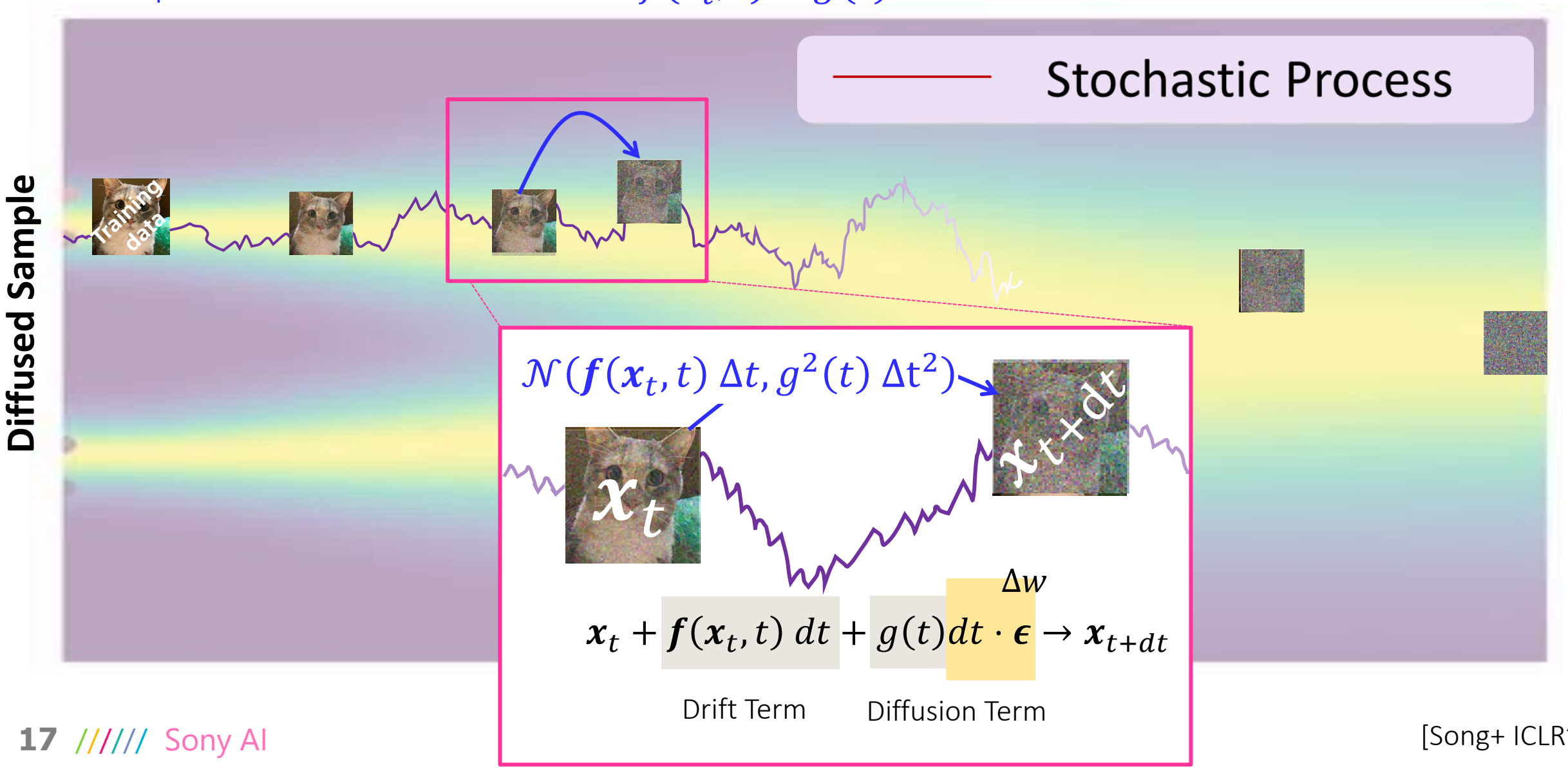
Backward Process: Learn NN to gradually denoise

Generated new image

Gradually transforming noise to a new image

Diffusion Model's **Forward** Process

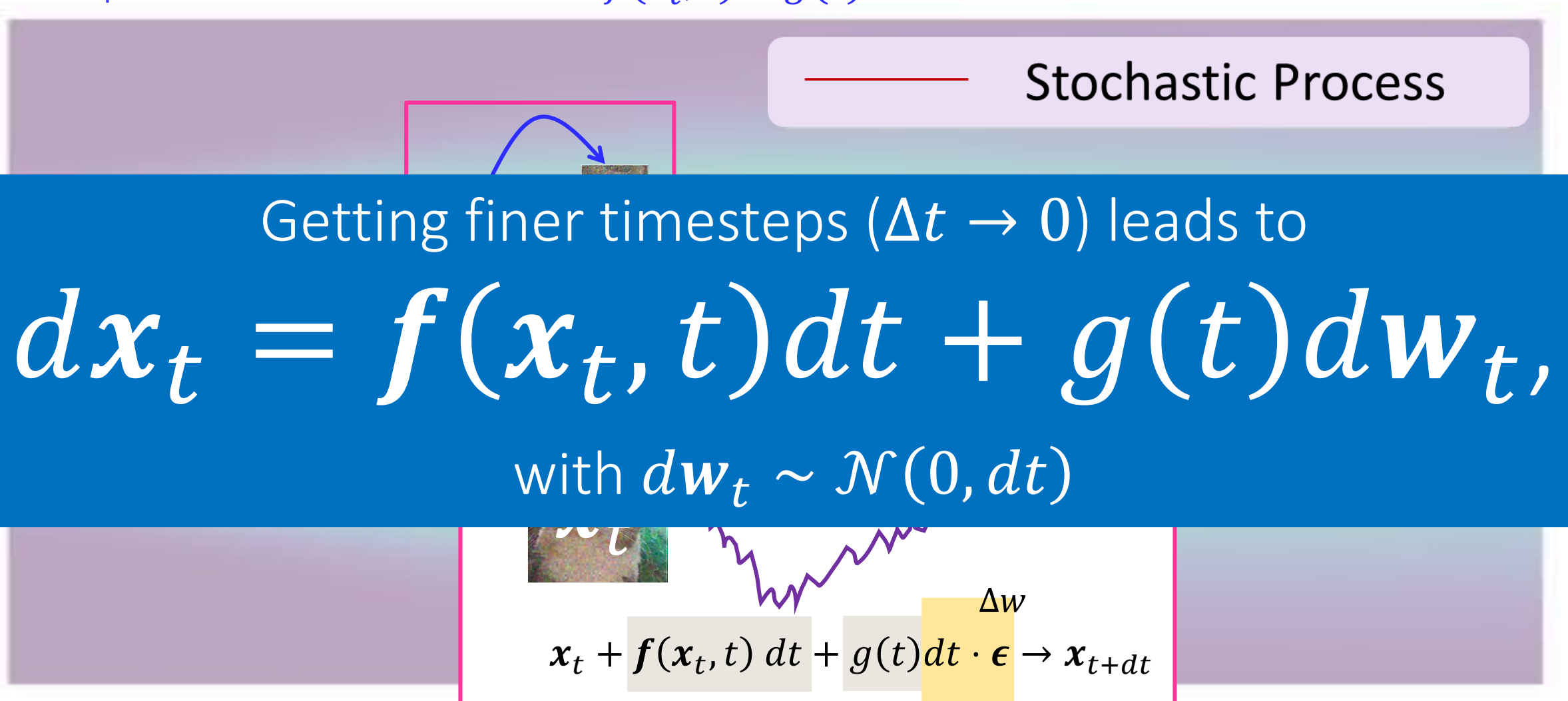
Users predefined noise schedulers: $f(x_t, t)$ & $g(t)$



[Song+ ICLR'21]

Diffusion Model's **Forward** Process

Users predefined noise schedulers: $f(x_t, t)$ & $g(t)$



Getting finer timesteps ($\Delta t \rightarrow 0$) leads to

$$dx_t = f(x_t, t)dt + g(t)dw_t,$$

with $dw_t \sim \mathcal{N}(0, dt)$

[Song+ ICLR'21]

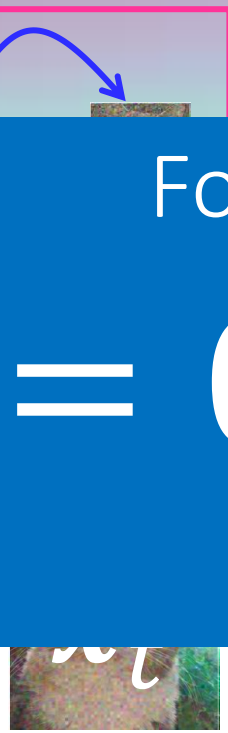
Diffusion Model's **Forward** Process

Users predefined noise schedulers: $f(x_t, t)$ & $g(t)$

Stochastic Process

For example,

$$f(x_t, t) = 0 \;\&\; g(t) = \sqrt{2t}$$



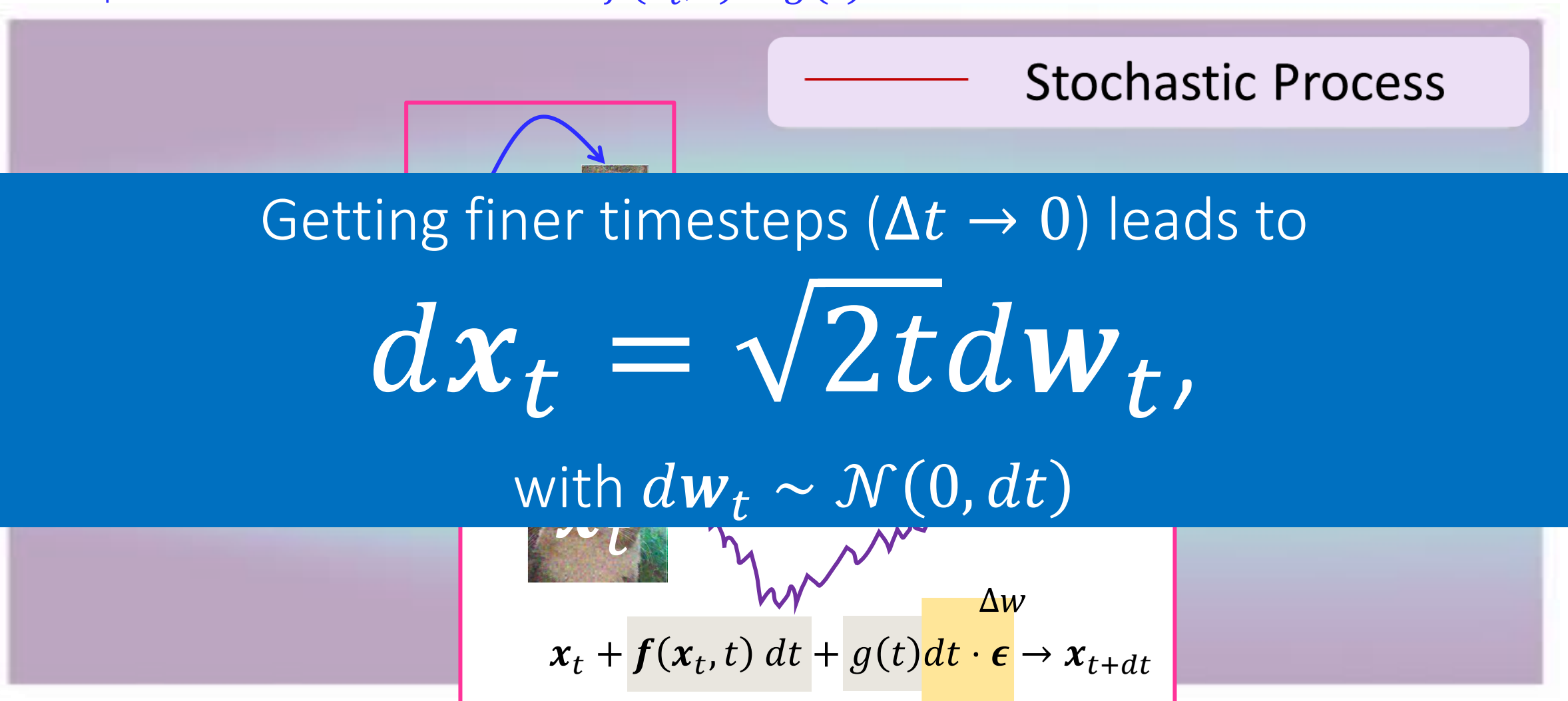
$$x_t + f(x_t, t)\,dt + g(t)\underbrace{dt \cdot \epsilon}_{\Delta w} \rightarrow x_{t+dt}$$

Drift Term Diffusion Term

[Song+ ICLR'21]

Diffusion Model's **Forward** Process

Users predefined noise schedulers: $f(x_t, t)$ & $g(t)$



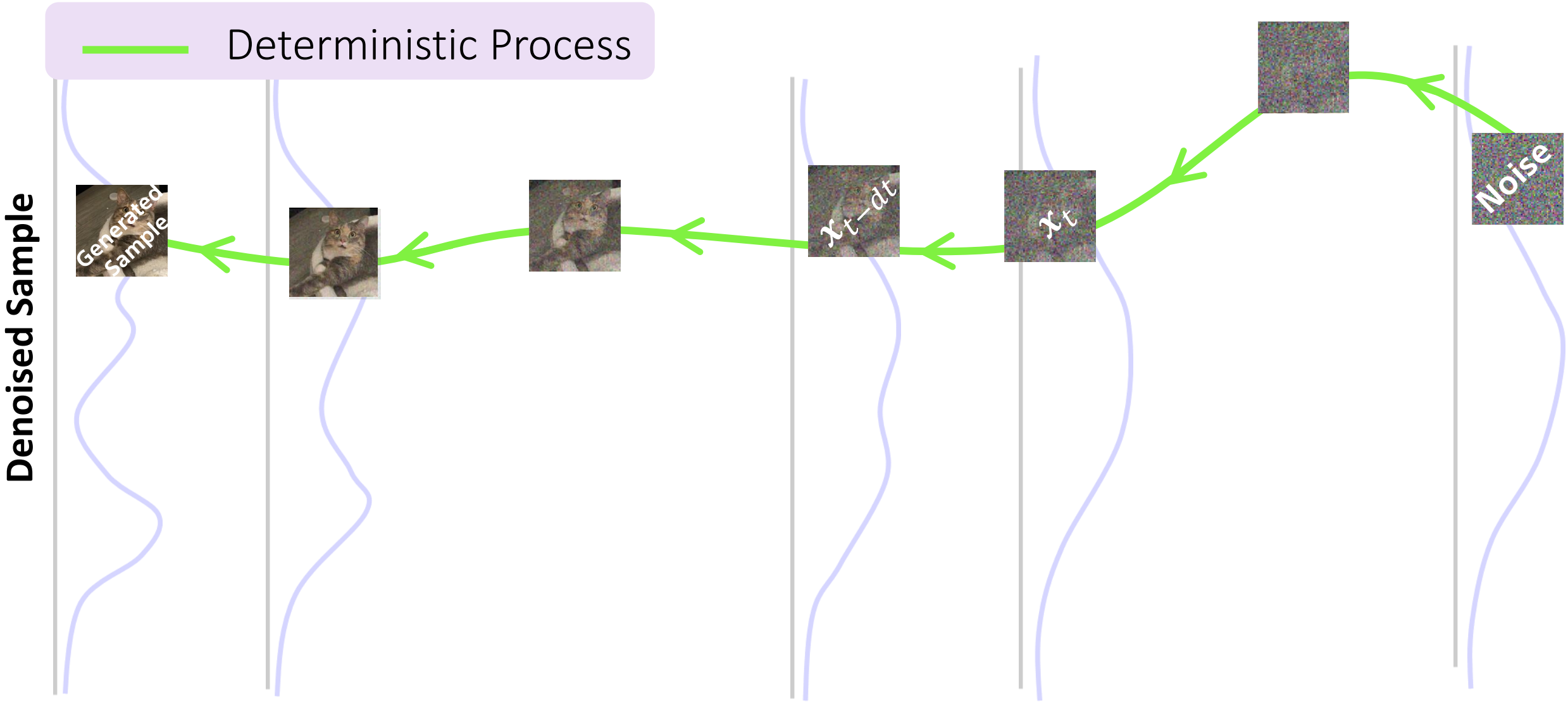
Getting finer timesteps ($\Delta t \rightarrow 0$) leads to

$$dx_t = \sqrt{2t}dw_t,$$

with $dw_t \sim \mathcal{N}(0, dt)$

[Song+ ICLR'21]

Diffusion Model's **Backward Deterministic** Process (Generation)

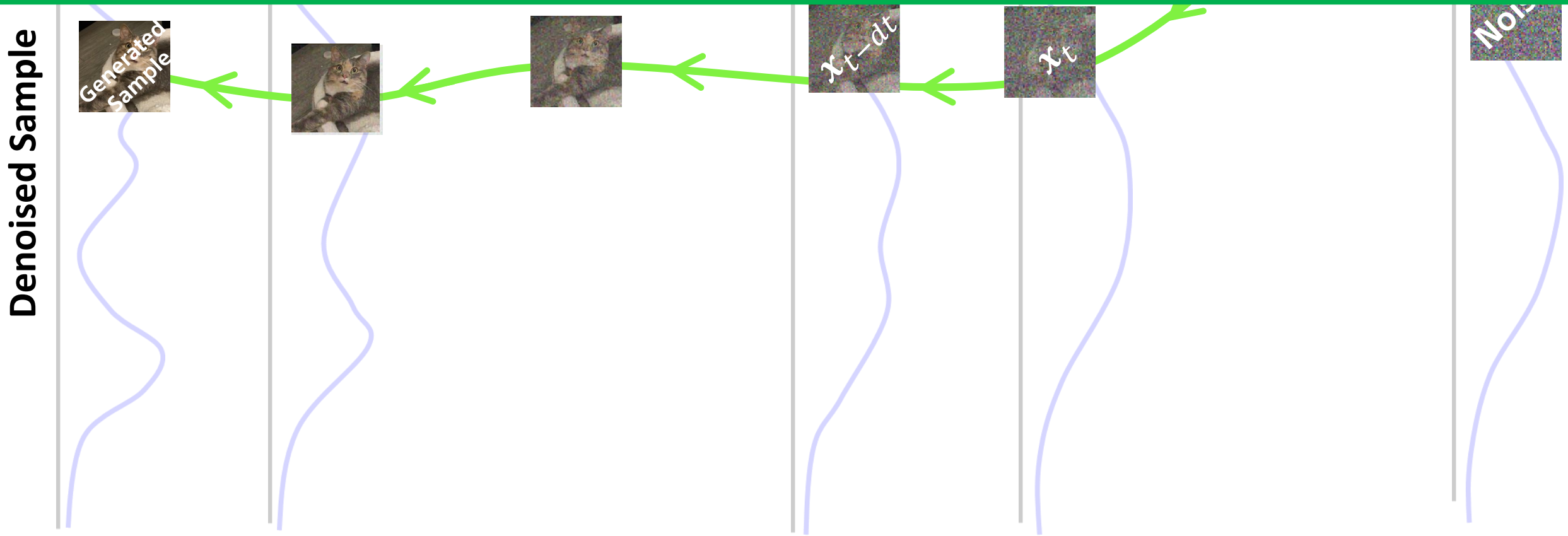


[Song+ ICLR'21]

Diffusion Model's **Backward Deterministic** Process (Generation)

Getting finer timesteps (dt→0) leads to

$$\frac{dx_t}{dt} = f(t) - \frac{1}{2} g^2(t) \nabla \log p_t(\boldsymbol{x}_t),\ \boldsymbol{x}_T \sim p_{\text{prior}}$$

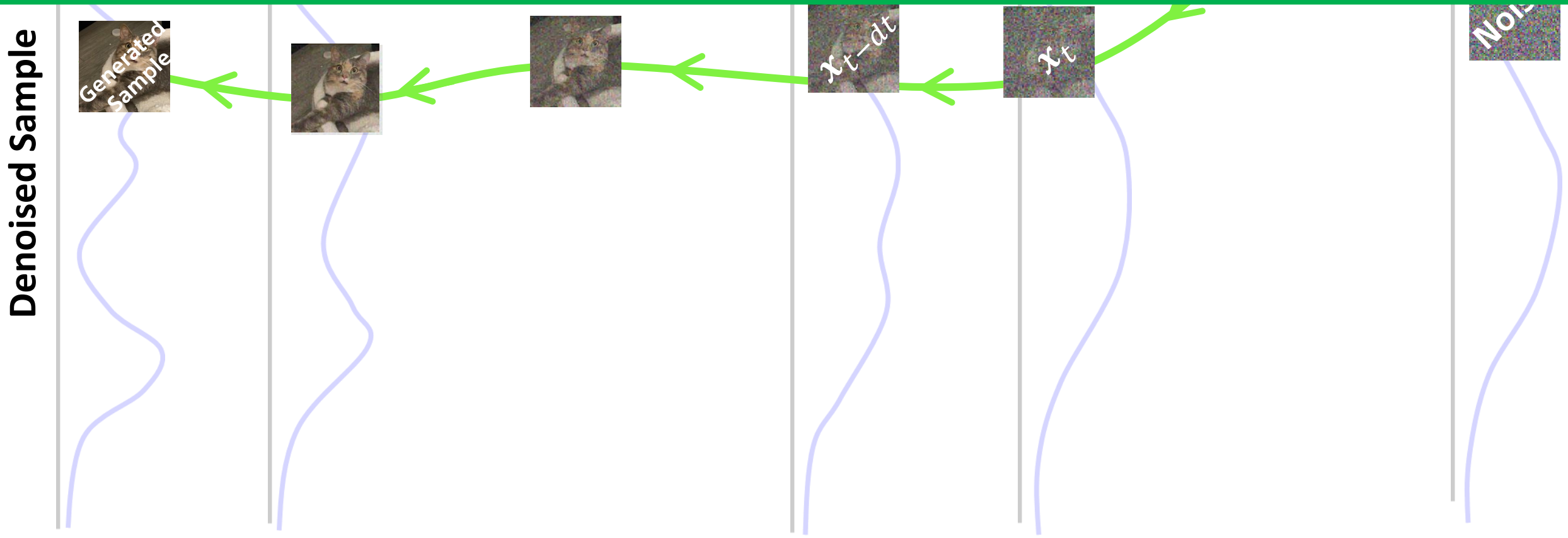


[Song+ ICLR'21]

Diffusion Model's **Backward Deterministic** Process (Generation)

In our example,

$$\frac{dx_t}{dt} = -t\nabla \log p_t(\boldsymbol{x}_t),\ \boldsymbol{x}_T \sim p_{\text{prior}}$$

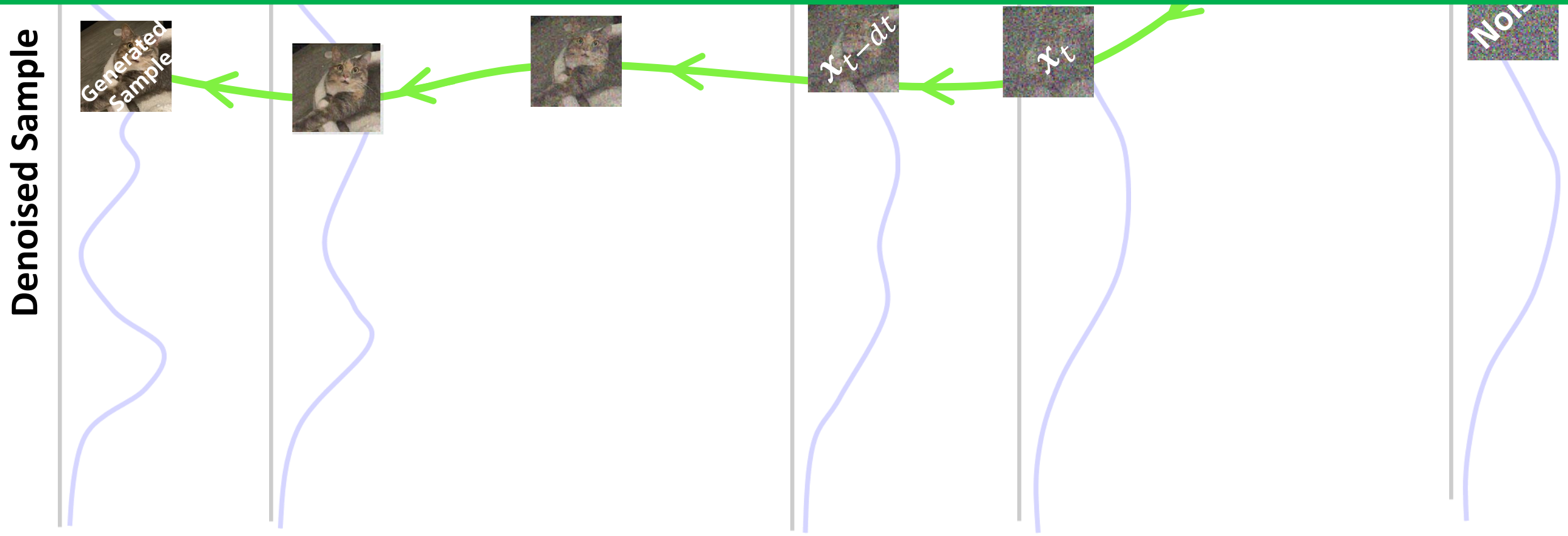


[Song+ ICLR'21]

Diffusion Model's **Backward Deterministic** Process (Generation)

For example,

$$f(x_t, t) = 0 \;\&\; g(t) = \sqrt{2t}$$

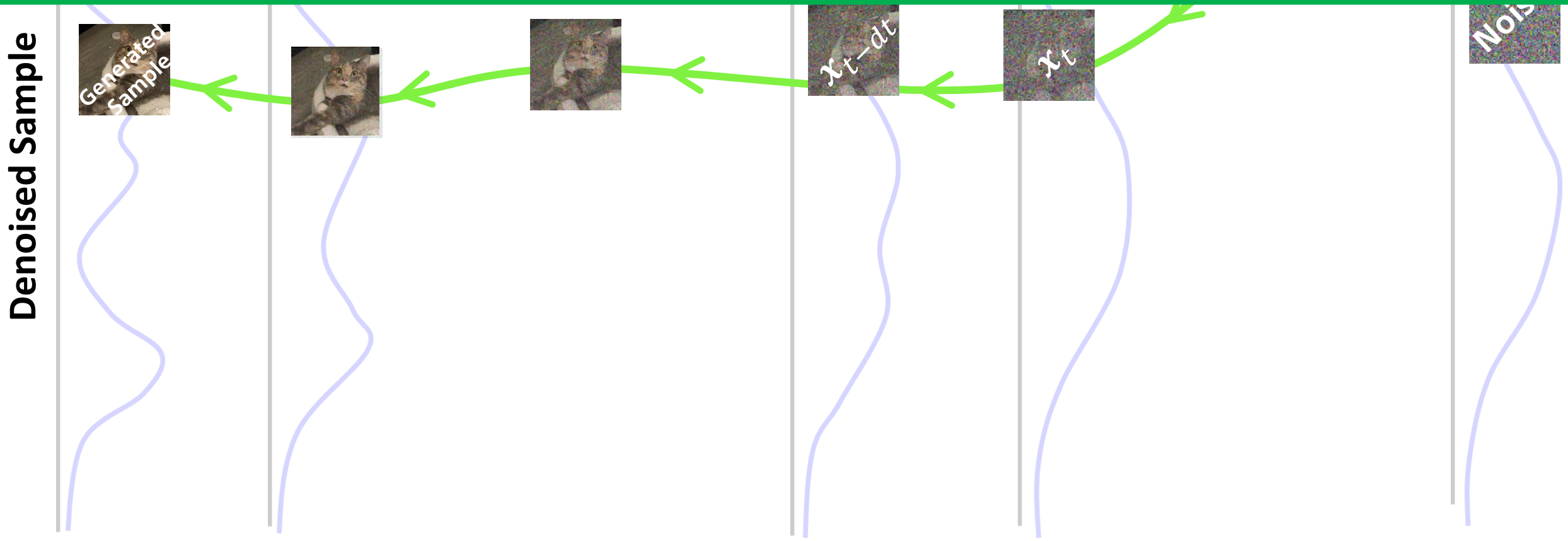


[Song+ ICLR'21]

Diffusion Model's **Backward Deterministic** Process (Generation)

In our example,

$$\frac{dx_t}{dt} = -t\nabla \log p_t(\boldsymbol{x}_t),\ \boldsymbol{x}_T \sim p_{\text{prior}}$$

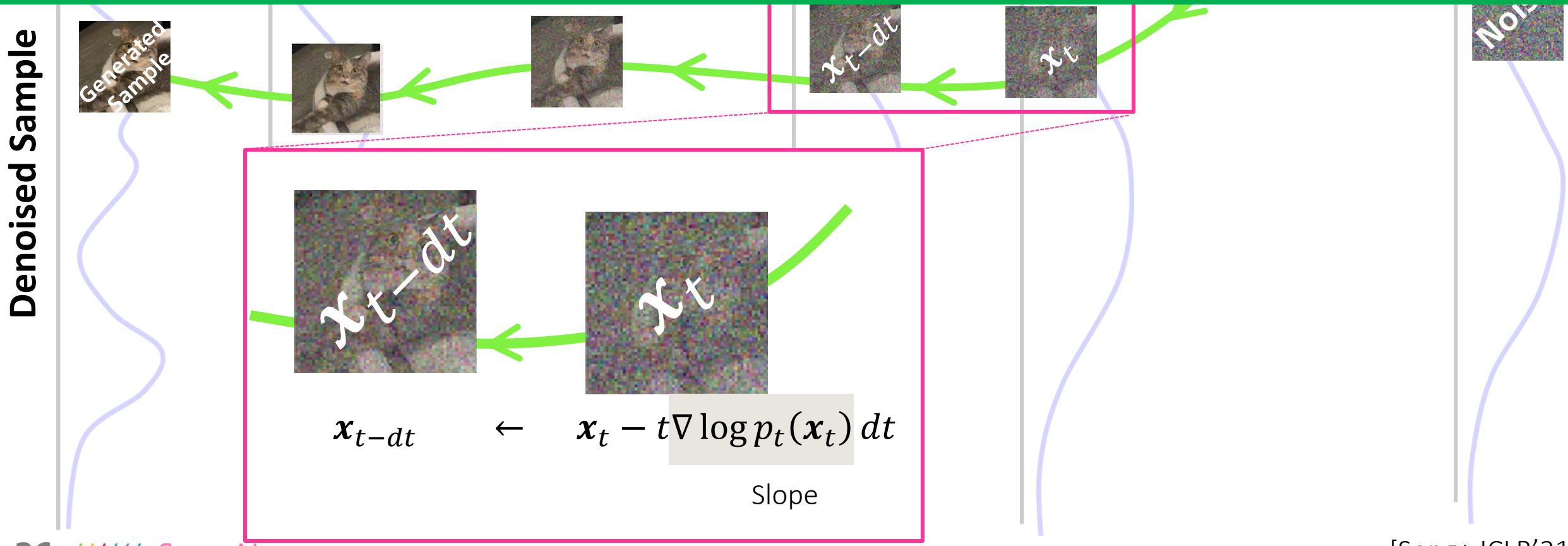


[Song+ ICLR'21]

Diffusion Model's **Backward Deterministic** Process (Generation)

In our example,

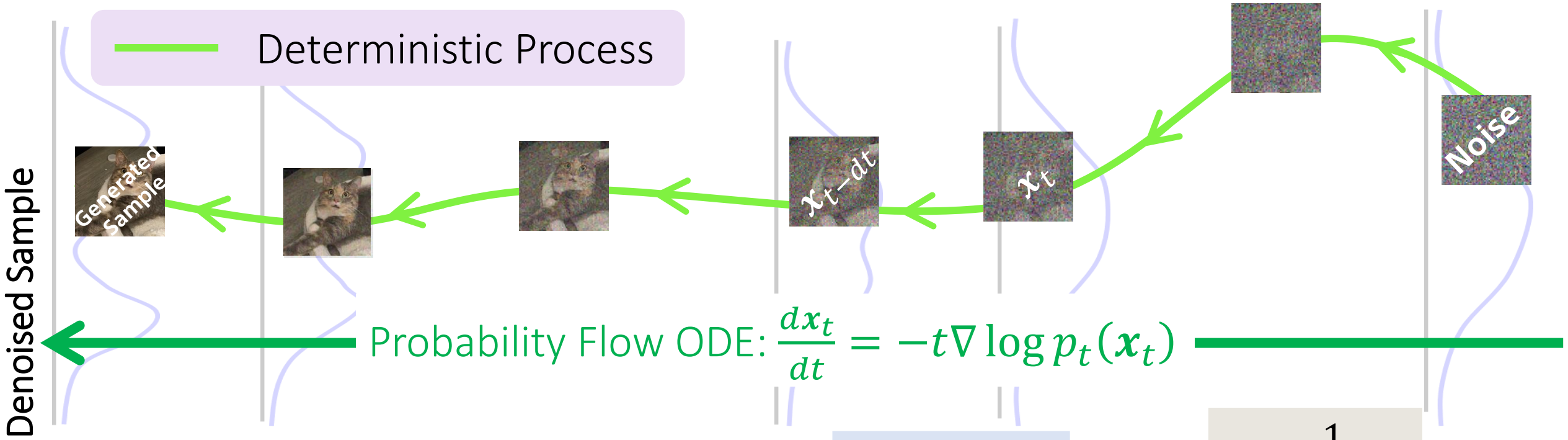
$$\frac{dx_t}{dt} = -t\nabla \log p_t(\boldsymbol{x}_t),\ \boldsymbol{x}_T \sim p_{\text{prior}}$$



[Song+ ICLR'21]

Score-Based Diffusion Model's Training and Sampling

Users predefined noise schedulers: $\boldsymbol{x_t} = a_t \boldsymbol{x_0} + b_t \boldsymbol{\epsilon}$ with $a_t = 1$ & $b_t = t$



$= \boldsymbol{x}_0 + t\boldsymbol{\epsilon}$

$= -\frac{1}{t}\boldsymbol{\epsilon}$

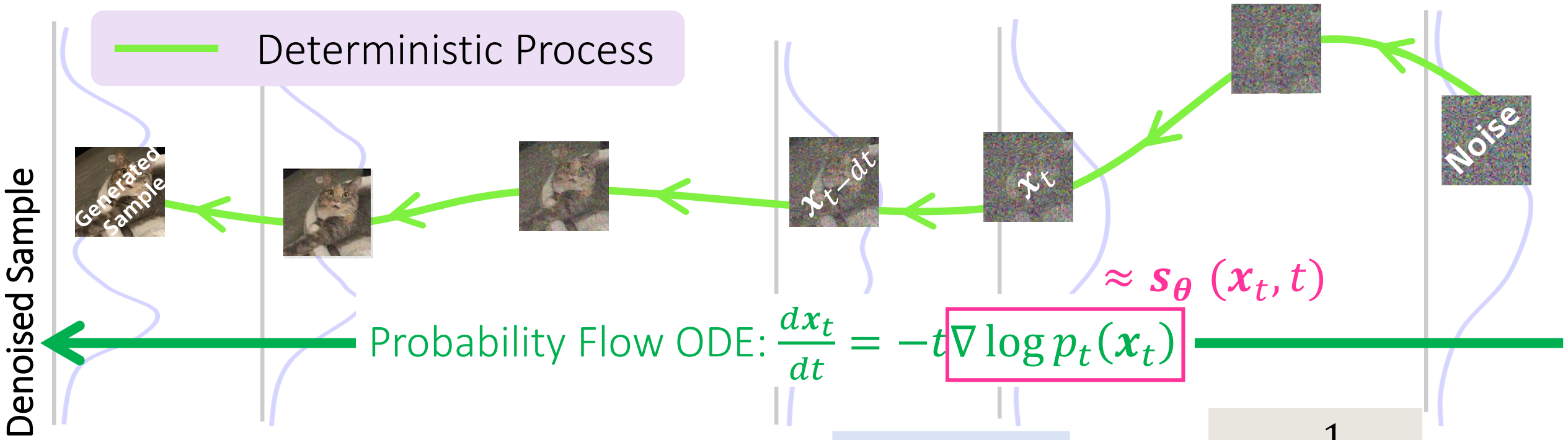
Denoising Score Matching (DSM):

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\mathbb{E}_{t\in\mathcal{U}[0,1]}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\lambda_t\left\|\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t) - \nabla \log p_t(\boldsymbol{x}_t|\boldsymbol{x}_0)\right\|^2\right]$$

[Song+ ICLR'21]

Score-Based Diffusion Model's Training and Sampling

Users predefined noise schedulers: $\boldsymbol{x_t} = a_t \boldsymbol{x_0} + b_t \boldsymbol{\epsilon}$ with $a_t = 1$ & $b_t = t$



$= x_0 + t\epsilon$

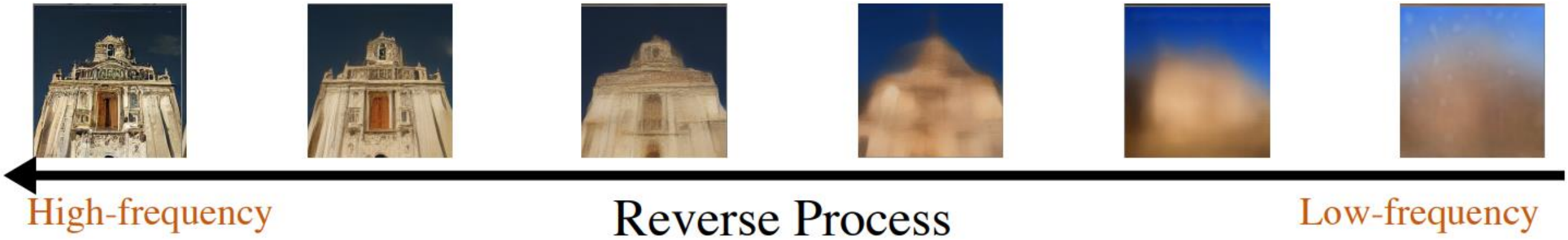
$= -\frac{1}{t}\epsilon$

Denoising Score Matching (DSM):

$$\mathbb{E}_{x\sim p_{\text{data}}(x)}\mathbb{E}_{t\in\mathcal{U}[0,1]}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\lambda_t\left\|s_\theta(x_t,t) - \nabla \log p_t(x_t|x_0)\right\|^2\right]$$

[Song+ ICLR'21]

What is Happening in Generation?

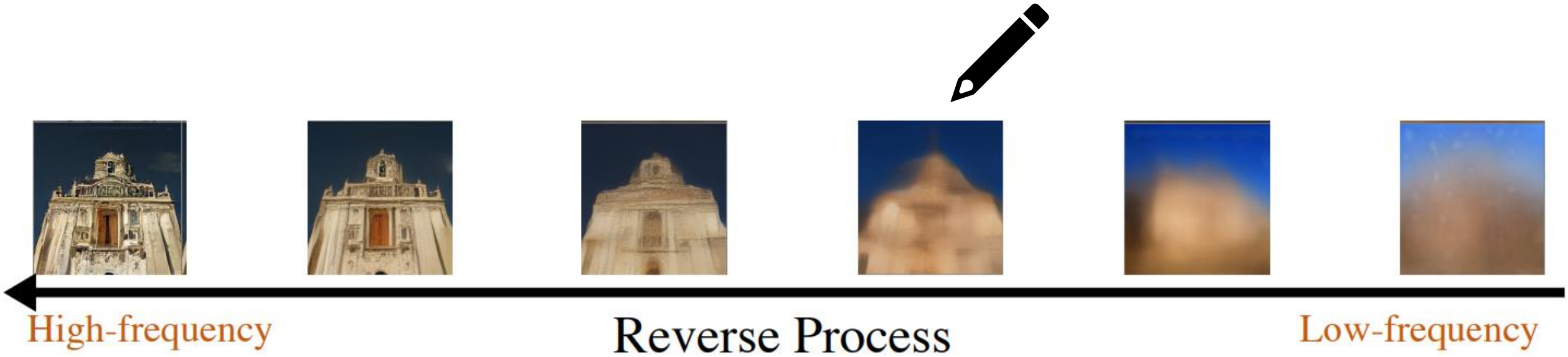


Fine-grained details

Global structure

[Yang+ CVPR'23]

What is Happening in Generation?

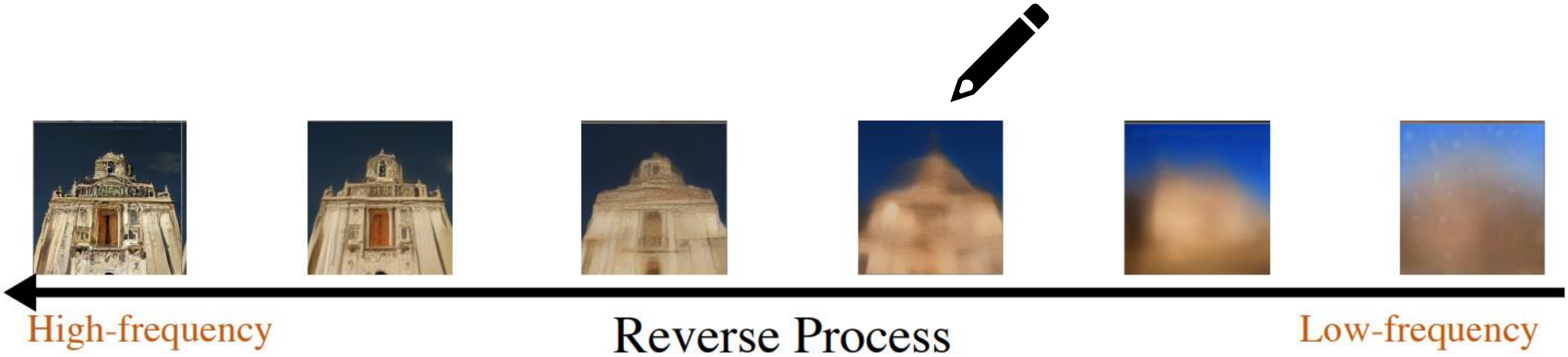


Fine-grained details

Global structure

[Yang+ CVPR'23]

What is Happening in Generation?



Fine-grained details

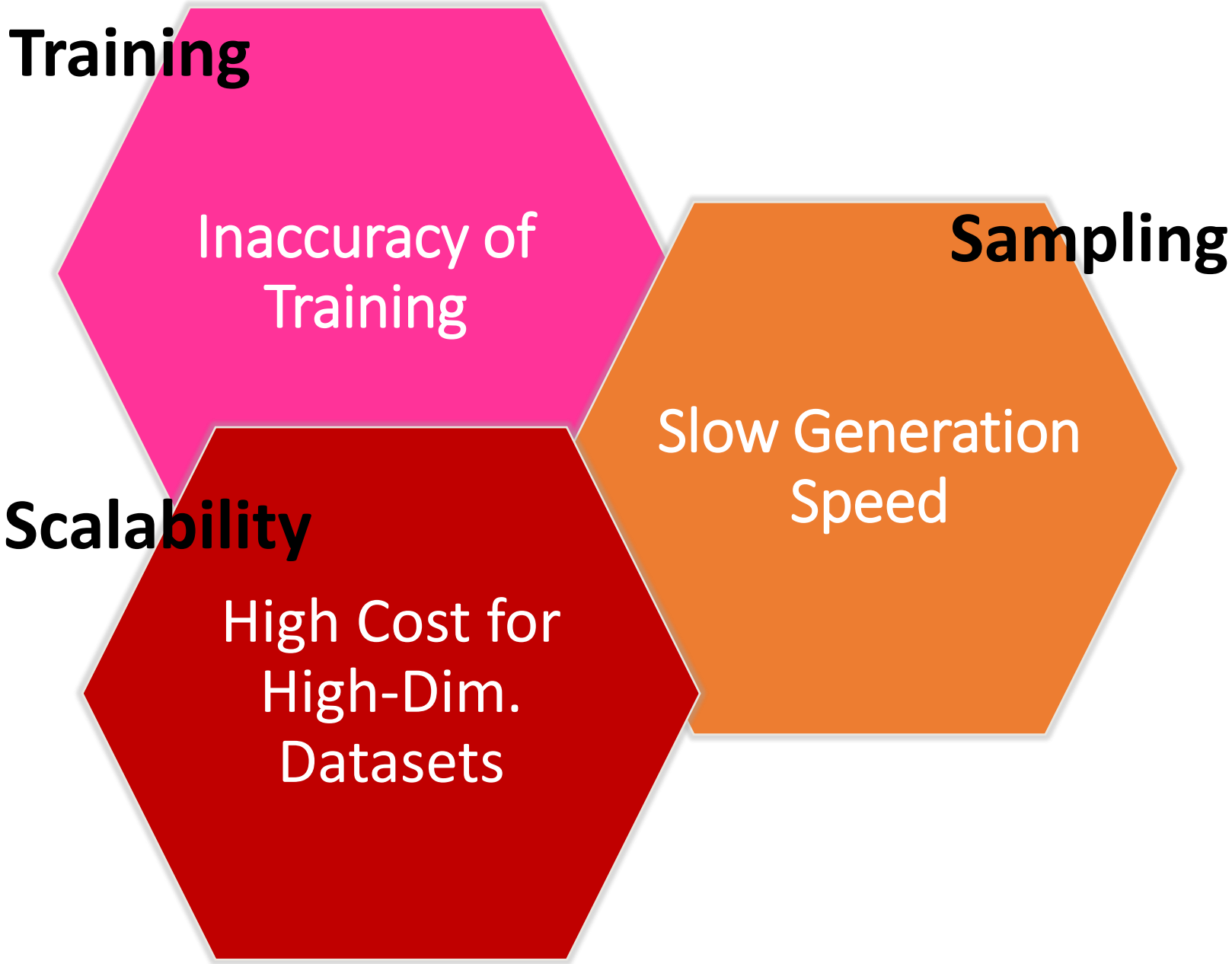
Global structure

Gradual denoising process, on the other hand,
offers flexibility and control!

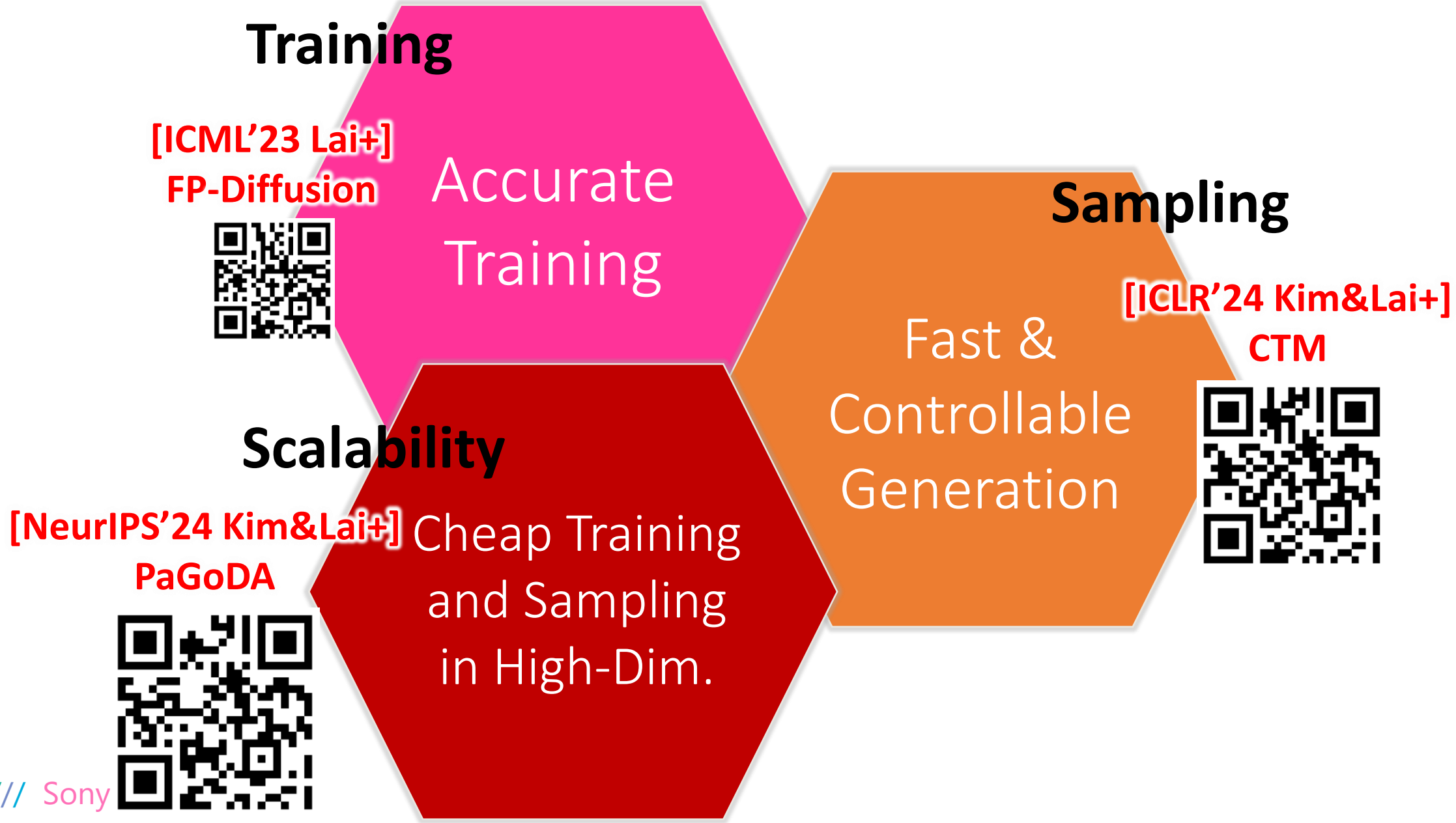
[Yang+ CVPR'23]

Problems of Diffusion Models

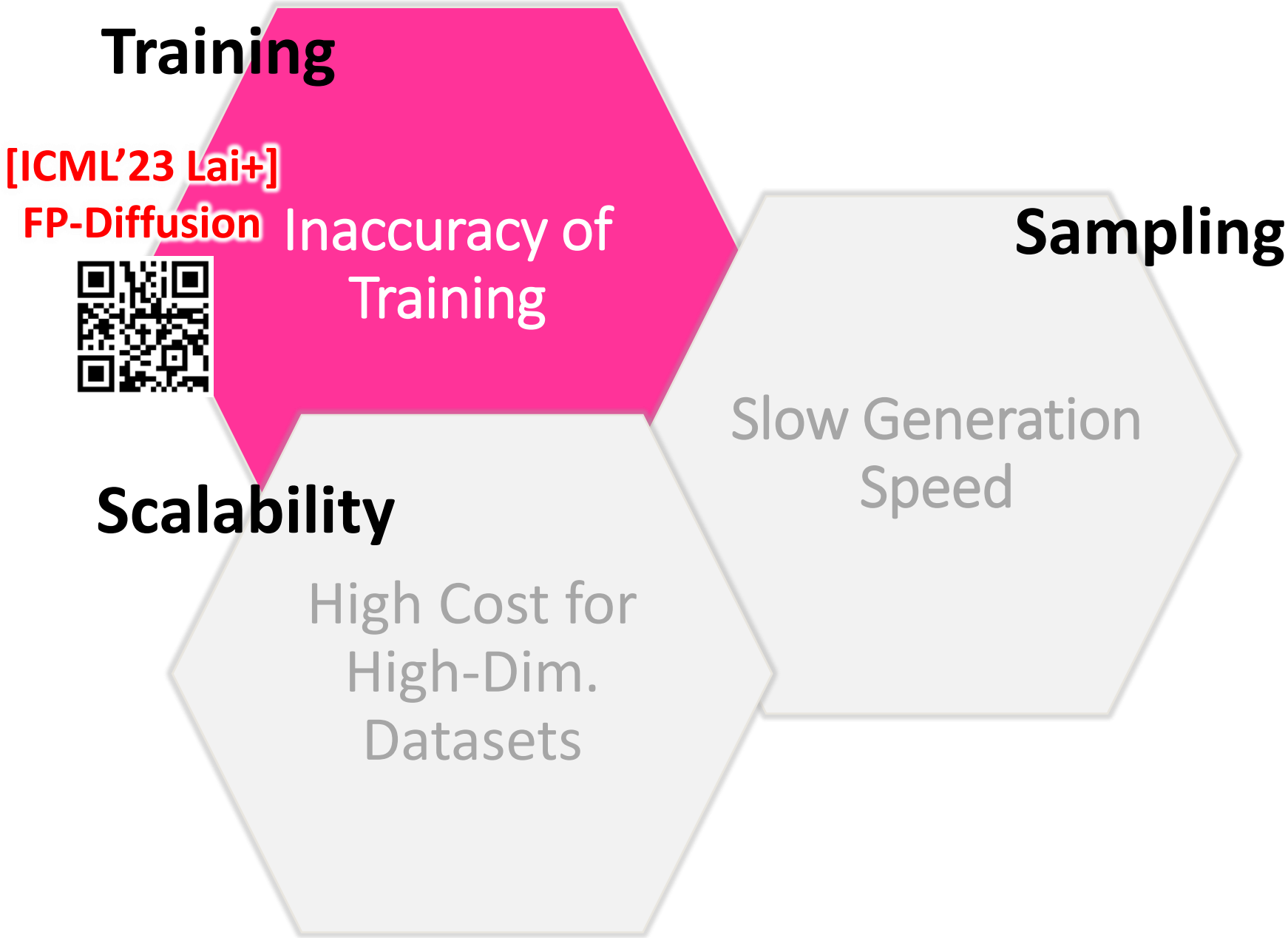
Problems of Diffusion Models



Making Diffusion Models More Powerful



Making Diffusion Model More Powerful



Motivation of FP-Diffusion

$$dx_t = \left[f_t - \frac{1}{2} g_t^2 \nabla \log p_t(x_t) \right] dt$$

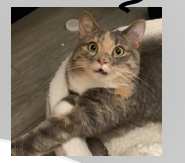
$$\approx \boldsymbol{s}_{\boldsymbol{\theta}}(x_t, t)$$

$t = T$

Prior Distribution = Gaussian

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\theta}) := \mathbb{E}_{t, p_t(x_t)} \| \boldsymbol{s}_{\boldsymbol{\theta}}(x_t, t) - \nabla \log p_t(x_t) \|^2$$

Data manifold



$t = 0$

Motivation of FP-Diffusion

$$dx_t = \left[f_t - \frac{1}{2} g_t^2 \nabla \log p_t(x_t) \right] dt$$

$$\approx \boldsymbol{s}_{\boldsymbol{\theta}}(x_t, t)$$

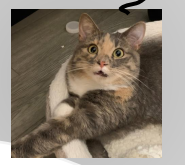
$t = T$

Prior Distribution = Gaussian

$$\mathcal{L}_{\mathrm{SM}}(\boldsymbol{\theta}) \coloneqq \mathbb{E}_{t, p_t(x_t)} \| \boldsymbol{s}_{\boldsymbol{\theta}}(x_t, t) - \nabla \log p_t(x_t) \|^2$$

NN is hard to learn functions with large Lipschitzness

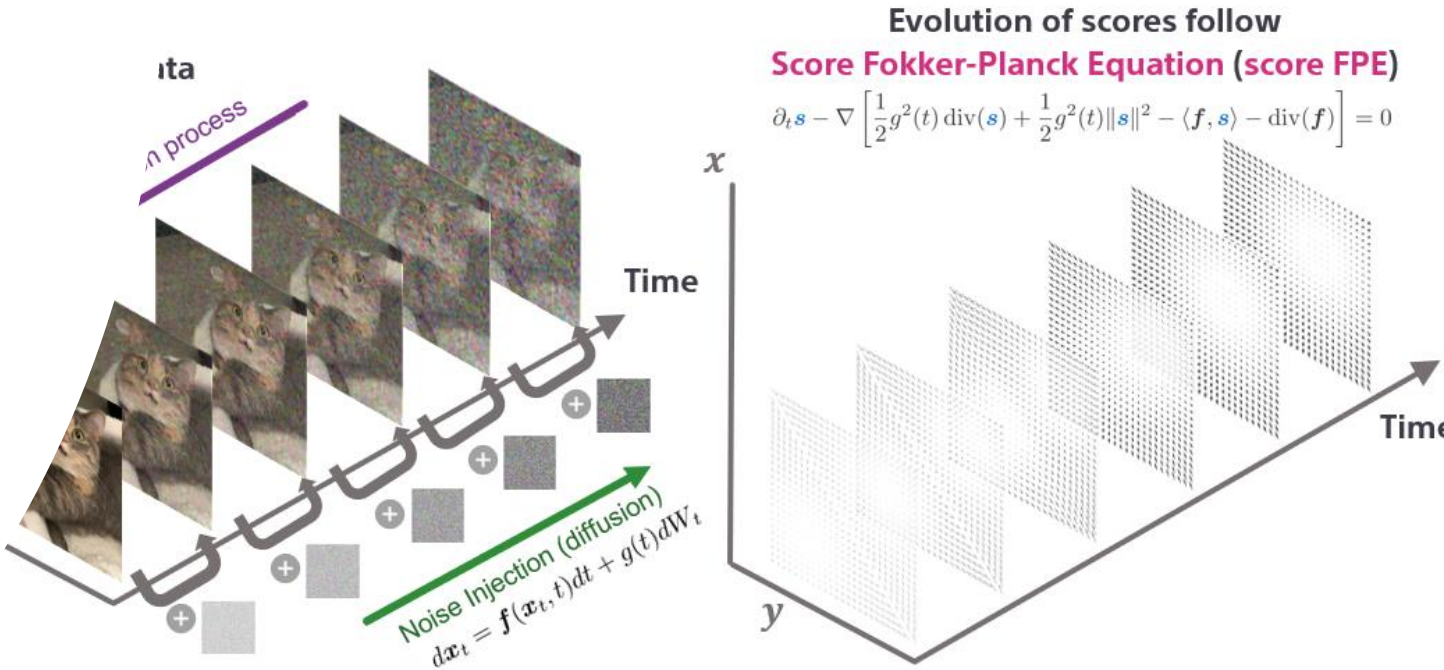
Data manifold



$t = 0$

FP-Diffusion

FP-Diffusion



New perspective of score evolution: Score FPE

Diffusion

Generation

Density

Densities follow **Fokker-Planck Equation (FPE)**

Forward SDE: $d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t$

Data

Data follow SDE

Probability flow ODE: $d\boldsymbol{x}_t = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g^2(t)\nabla_x \log p_t(\boldsymbol{x}_t)\right]dt$

New perspective of score evolution: Score FPE

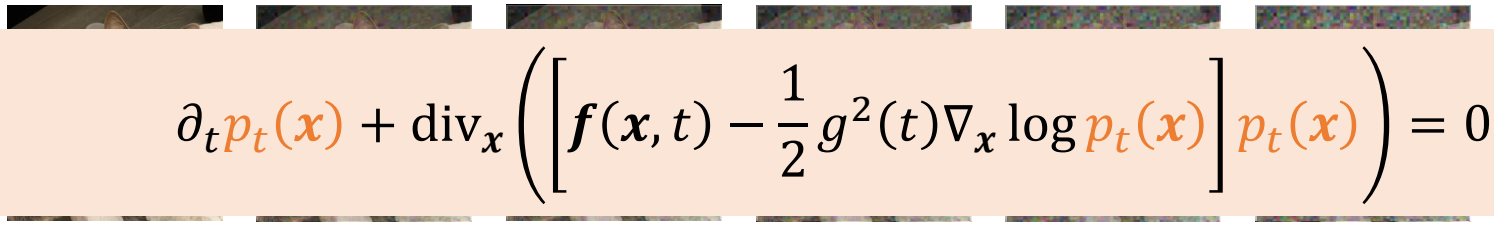
Diffusion

Generation

Density

Densities follow **Fokker-Planck Equation (FPE)**

Forward SDE: $d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t$



$$\partial_t p_t(\boldsymbol{x}) + \mathrm{div}_{\boldsymbol{x}}\left(\left[\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] p_t(\boldsymbol{x})\right) = 0$$

Share the same density PDE

Probability flow ODE: $d\boldsymbol{x}_t = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g^2(t)\nabla_x \log p_t(\boldsymbol{x}_t)\right]dt$

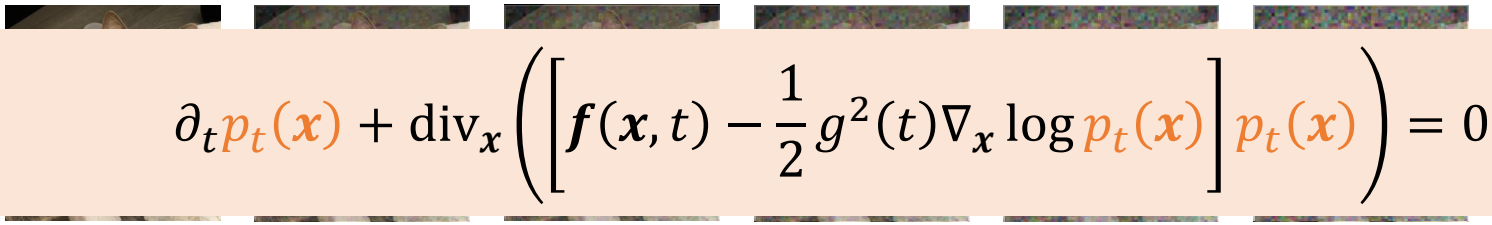
New perspective of score evolution: Score FPE

Diffusion

Generation

Score: $\boldsymbol{s}(\boldsymbol{x}, t) := \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$

Forward SDE: $d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t$



$$\partial_t p_t(\boldsymbol{x}) + \mathrm{div}_{\boldsymbol{x}}\left(\left[\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2}g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right] p_t(\boldsymbol{x})\right) = 0$$

(FPE)

Share the same density PDE

Probability flow ODE: $d\boldsymbol{x}_t = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2}g^2(t)\nabla_x \log p_t(\boldsymbol{x}_t)\right]dt$

New perspective of score evolution: Score FPE

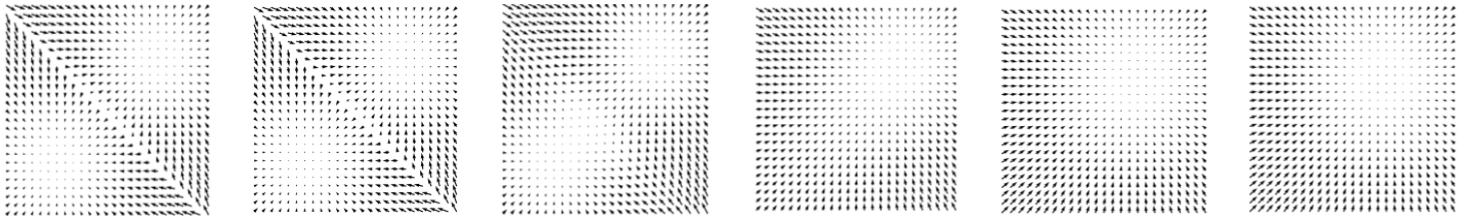
[ICML'23 Lai+]

$$\partial_t \boldsymbol{s} - \nabla_{\boldsymbol{x}} \left[\frac{1}{2} g^2(t) \mathrm{div}_{\boldsymbol{x}}(\boldsymbol{s}) + \frac{1}{2} g^2(t) \|\boldsymbol{s}\|_2^2 - \langle \boldsymbol{f}, \boldsymbol{s} \rangle - \mathrm{div}(\boldsymbol{f}) \right] = 0$$

Diffusion

Generation

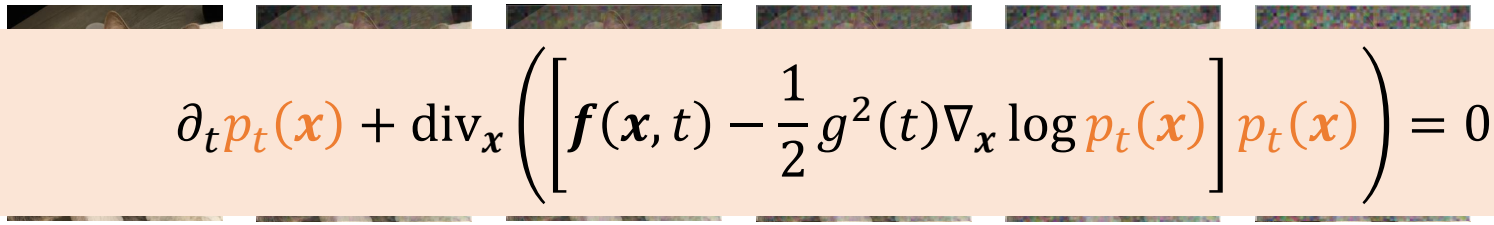
Score



Scores follow **Score FPE**

Score: $\boldsymbol{s}(\boldsymbol{x}, t) := \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$

Forward SDE: $d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t$



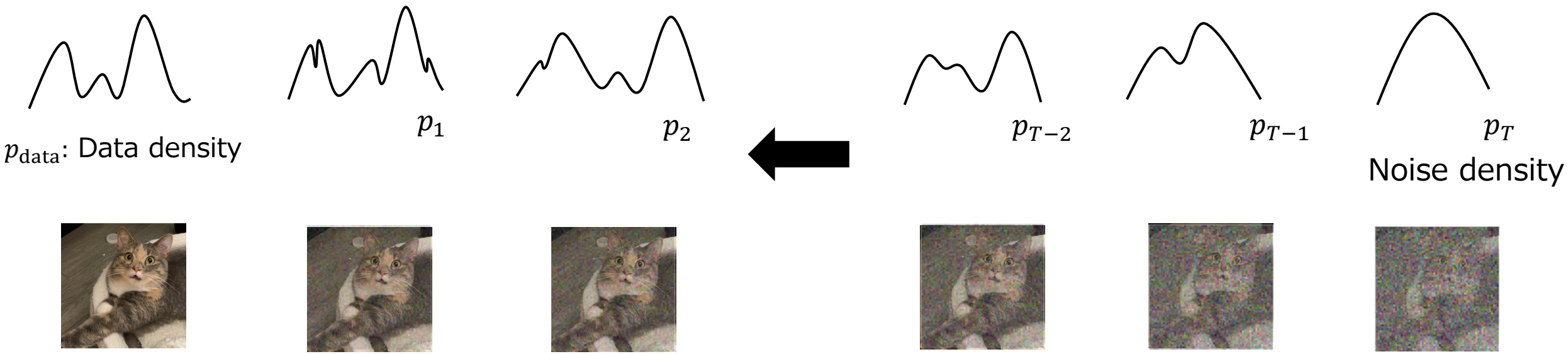
$$\partial_t p_t(\boldsymbol{x}) + \mathrm{div}_{\boldsymbol{x}} \left(\left[\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2} g^2(t) \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) \right] p_t(\boldsymbol{x}) \right) = 0$$

Probability flow ODE: $d\boldsymbol{x}_t = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{1}{2} g^2(t) \nabla_x \log p_t(\boldsymbol{x}_t) \right] dt$

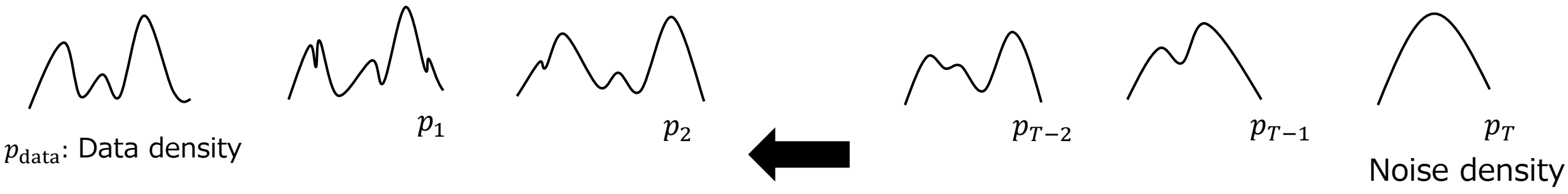
(FPE)

Share the same density PDE

Score FPE-error ↙ ⇒ density estimation quality ↗



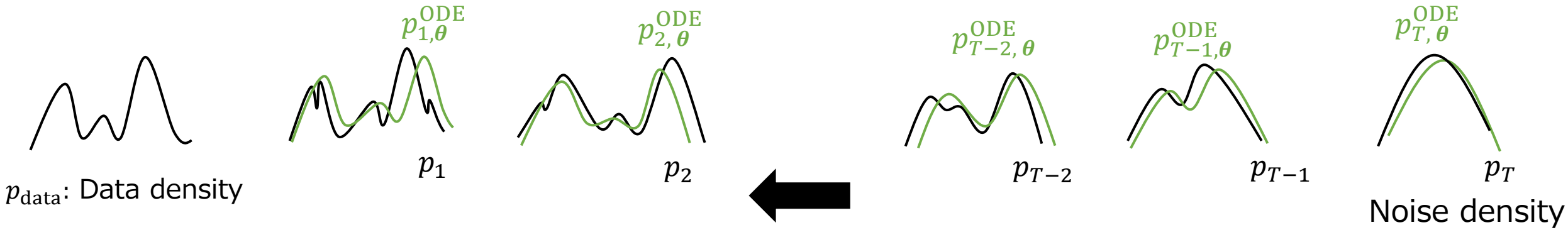
Score FPE-error ↙ ⇒ density estimation quality ↗



$$\mathcal{J}_{\text{ScoreFPE}}(\boldsymbol{\theta}) \coloneqq \left\| \partial_t \boldsymbol{s}_{\boldsymbol{\theta}}^{\text{NEW}} - t\nabla_x \left[\text{div}_x(\boldsymbol{s}_{\boldsymbol{\theta}}^{\text{NEW}}) + \|\boldsymbol{s}_{\boldsymbol{\theta}}^{\text{NEW}}\|_2^2 \right] \right\|_{\ell_p}^m$$

Score FPE-error ↙ ⇒ density estimation quality ↗

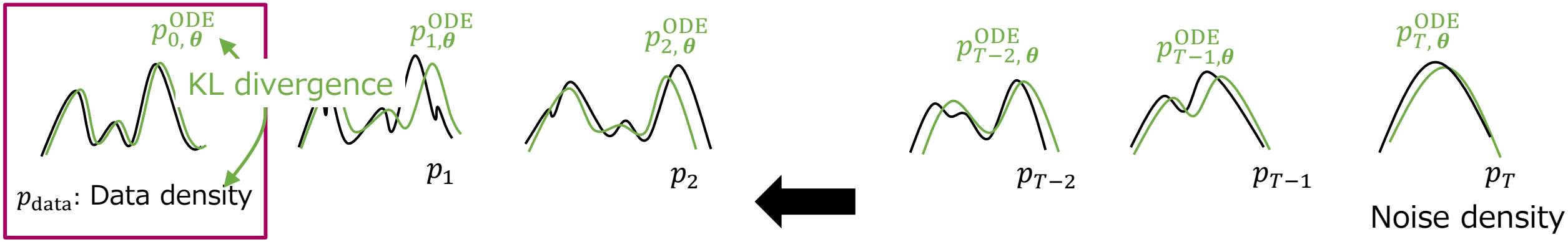
Probability Flow ODE: $\frac{dx_t}{dt} = -t s_{\theta}^{\mathrm{NEW}}(x_t, t)$



$$\mathcal{J}_{\mathrm{ScoreFPE}}(\boldsymbol{\theta}) \coloneqq \left\| \partial_t \boldsymbol{s}_{\boldsymbol{\theta}}^{\mathrm{NEW}} - t\nabla_x \left[\mathrm{div}_x(\boldsymbol{s}_{\boldsymbol{\theta}}^{\mathrm{NEW}}) + \|\boldsymbol{s}_{\boldsymbol{\theta}}^{\mathrm{NEW}}\|_2^2 \right] \right\|_{\ell_p}^m$$

Score FPE-error ↙ ⇒ density estimation quality ↗

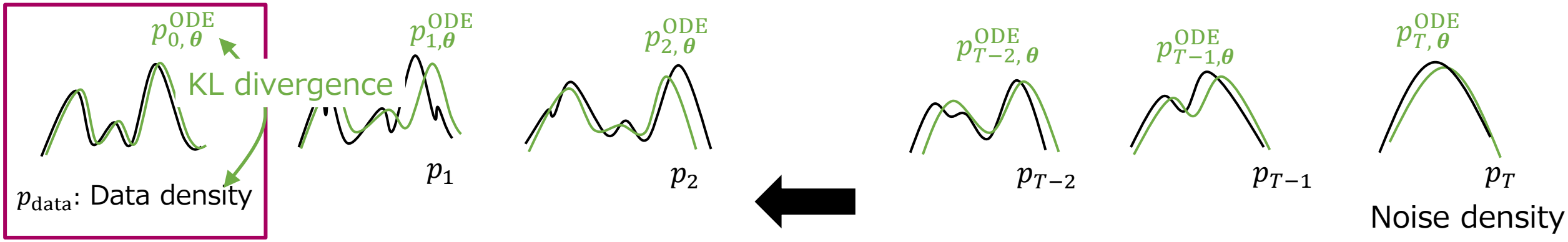
Probability Flow ODE: $\frac{dx_t}{dt} = -t s_{\theta}^{\mathrm{NEW}}(x_t, t)$



$$\mathcal{J}_{\mathrm{ScoreFPE}}(\boldsymbol{\theta}) \coloneqq \left\| \partial_t \boldsymbol{s}_{\boldsymbol{\theta}}^{\mathrm{NEW}} - t\nabla_x \left[\mathrm{div}_x(\boldsymbol{s}_{\boldsymbol{\theta}}^{\mathrm{NEW}}) + \left\| \boldsymbol{s}_{\boldsymbol{\theta}}^{\mathrm{NEW}} \right\|_2^2 \right] \right\|_{\ell_p}^m$$

Score FPE-error ↙ ⇒ density estimation quality ↗

Probability Flow ODE: $\frac{dx_t}{dt} = -t s_\theta^{\mathrm{NEW}}(x_t, t)$



$$\mathcal{J}_{\mathrm{ScoreFPE}}(\boldsymbol{\theta}) \coloneqq \left\| \partial_t \boldsymbol{s}_{\boldsymbol{\theta}}^{\mathrm{NEW}} - t\nabla_x \left[\mathrm{div}_x(\boldsymbol{s}_{\boldsymbol{\theta}}^{\mathrm{NEW}}) + \left\| \boldsymbol{s}_{\boldsymbol{\theta}}^{\mathrm{NEW}} \right\|_2^2 \right] \right\|_{\ell_p}^m$$

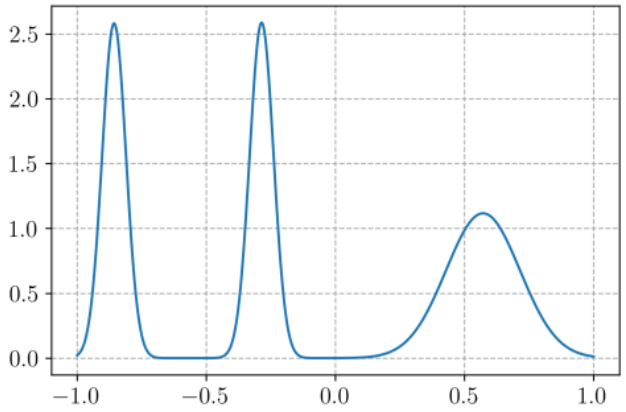
[ICML'23 Lai+]

Theorems.

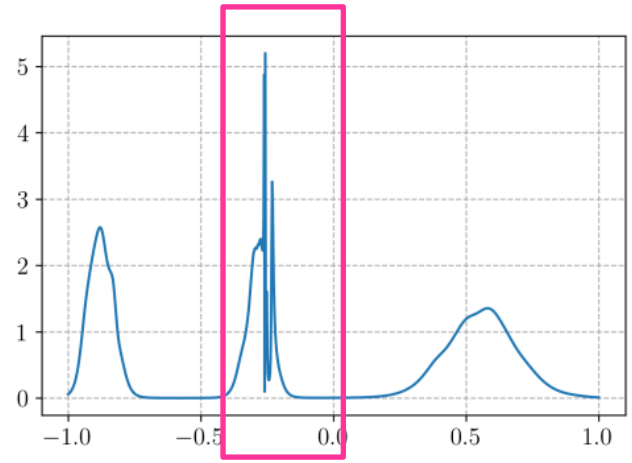
$$D_{\mathrm{KL}}(p_{\mathrm{data}} \| p_{0,\boldsymbol{\theta}}^{\mathrm{ODE}})$$

$$\lesssim D_{\mathrm{KL}}(p_{\mathrm{T}} \| p_{T,\boldsymbol{\theta}}^{\mathrm{ODE}}) + \mathcal{J}_{\mathrm{DSM}}(\boldsymbol{\theta}) + \mathcal{J}_{\mathrm{DSM}}^{\frac{1}{2}}(\boldsymbol{\theta}) \left(\mathcal{J}_{\mathrm{ScoreFPE}}(\boldsymbol{\theta}) + \mathcal{J}_{\mathrm{ScoreFPE}}^{\frac{1}{2}}(\boldsymbol{\theta}) + C \right)^{1/2}$$

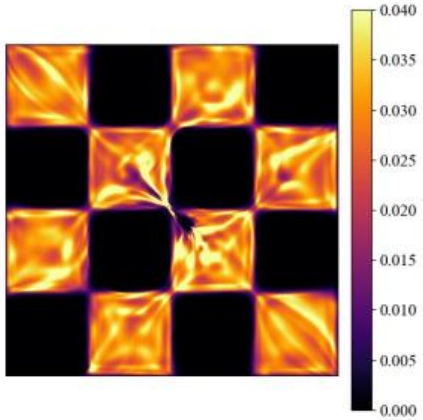
Results on toy datasets



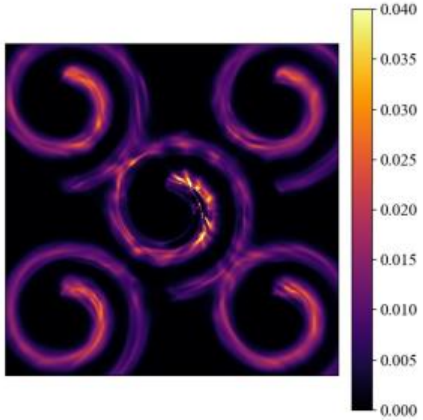
(a) Data density



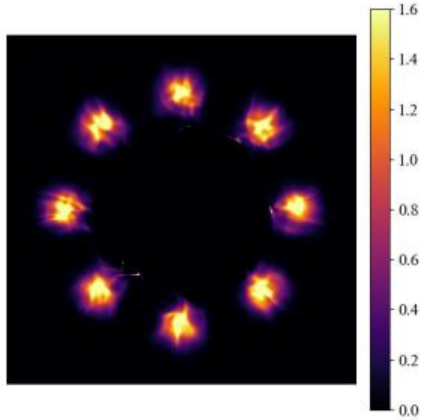
(b) Vanilla DSM ($\alpha = \beta = 0.0$)



(a) Vanilla DSM
($\alpha = \beta = 0.0$)

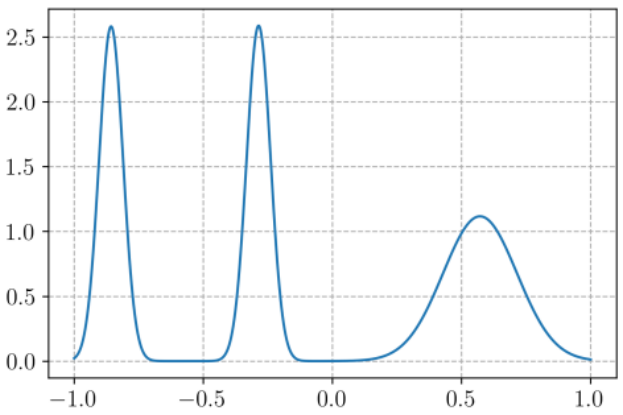


(c) Vanilla DSM
($\alpha = \beta = 0.0$)

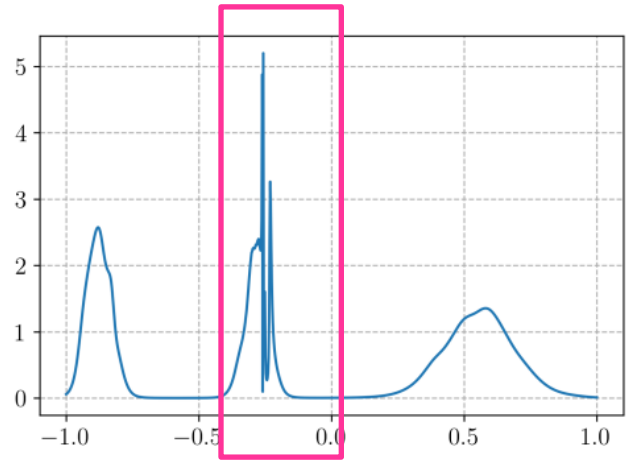


(e) Vanilla DSM
($\alpha = \beta = 0.0$)

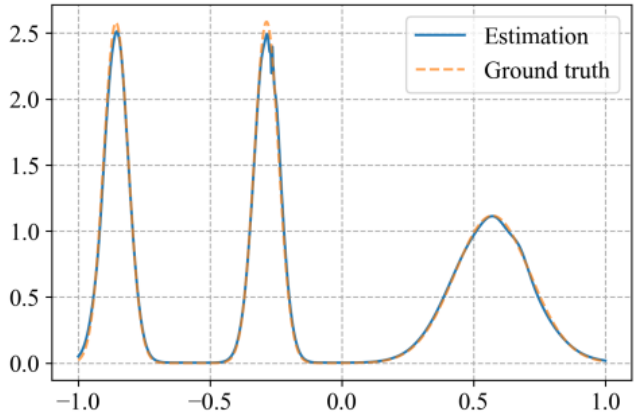
Results on toy datasets



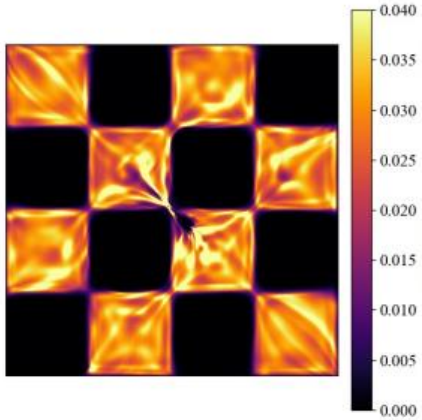
(a) Data density



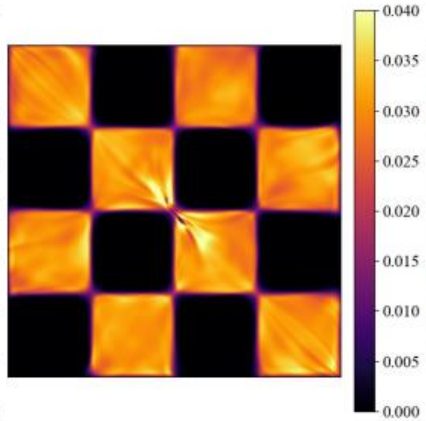
(b) Vanilla DSM ($\alpha = \beta = 0.0$)



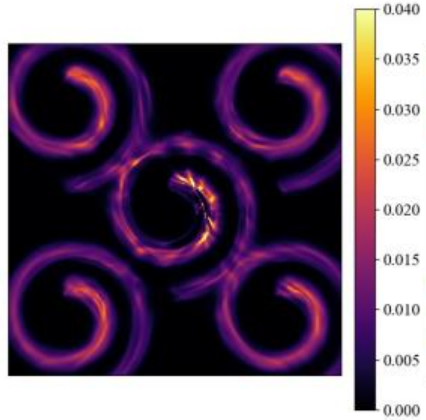
(c) FP-Diffusion (with $\alpha = 0.0015$)



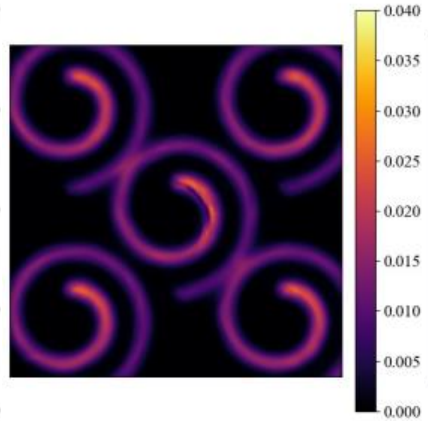
(a) Vanilla DSM ($\alpha = \beta = 0.0$)



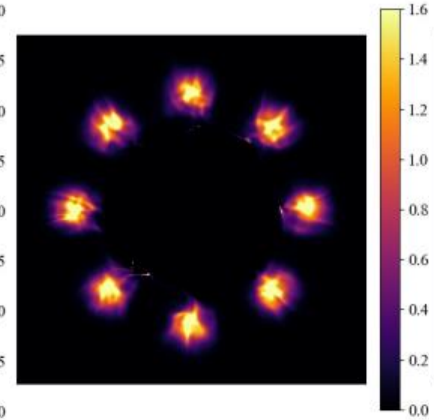
(b) FP-Diffusion (with $\alpha = 0.0015$)



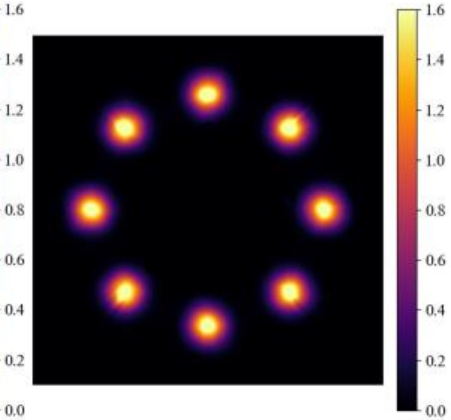
(c) Vanilla DSM ($\alpha = \beta = 0.0$)



(d) FP-Diffusion (with $\alpha = 0.0015$)



(e) Vanilla DSM ($\alpha = \beta = 0.0$)

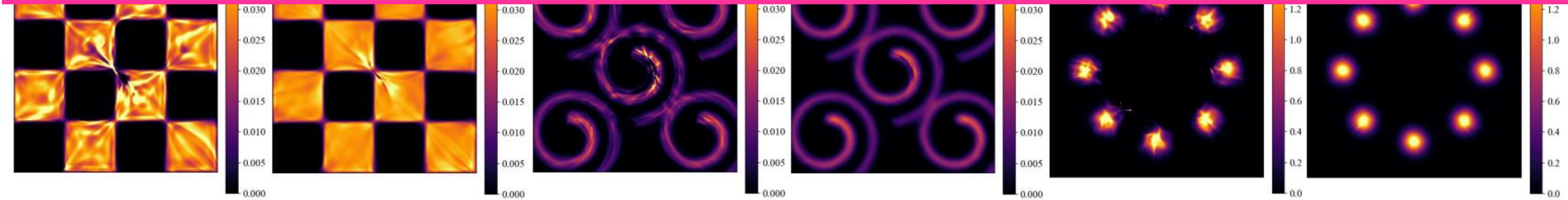


(f) FP-Diffusion (with $\alpha = 0.0015$)

Results on toy datasets

Imposing Score FPE + Vanilla DSM → Improve density learning

Score FPE bridges PDE to practical Diffusion Model



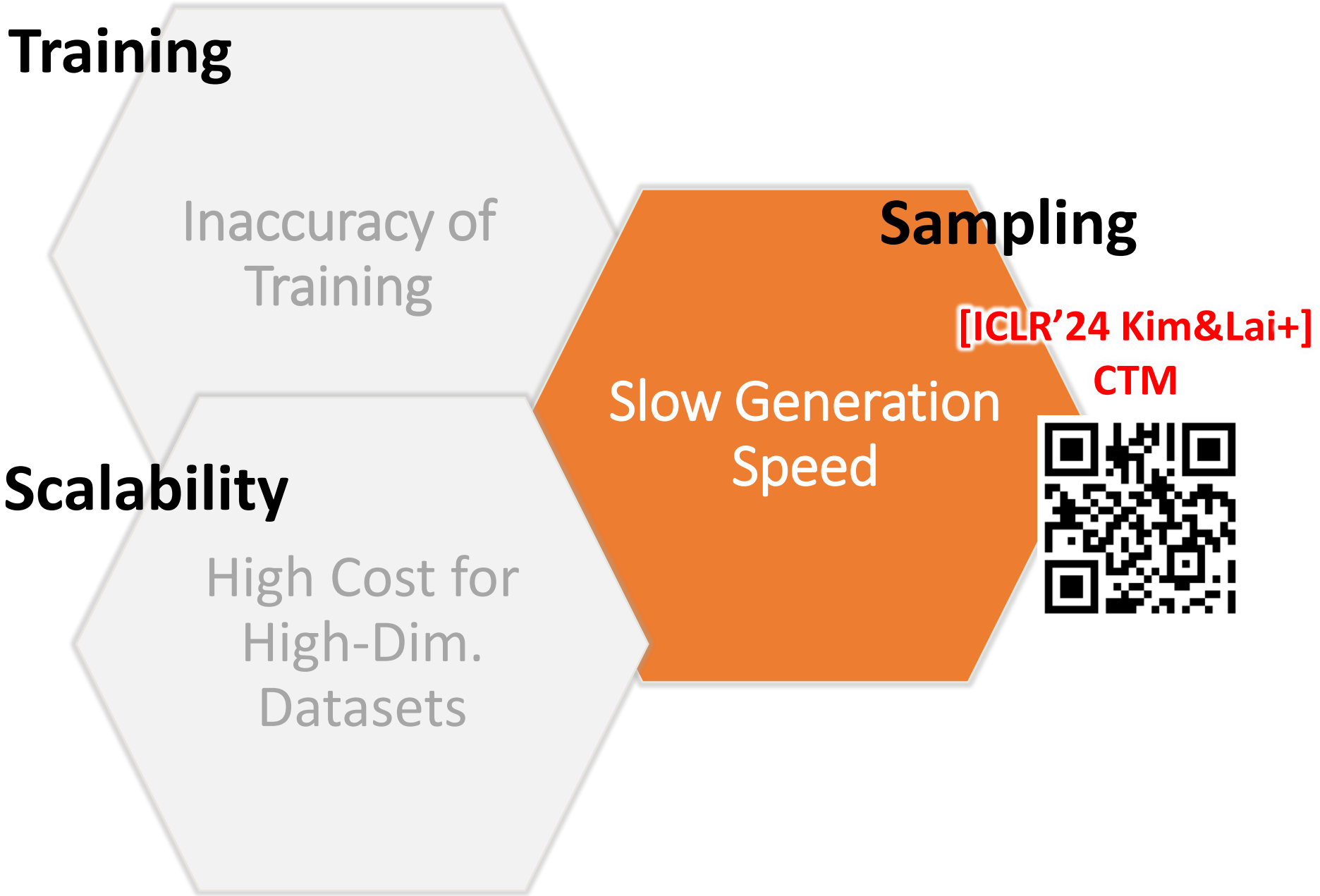
(a) Vanilla DSM ($\alpha = \beta = 0.0$)
(b) FP-Diffusion (with $\alpha = 0.0015$)
(c) Vanilla DSM ($\alpha = \beta = 0.0$)
(d) FP-Diffusion (with $\alpha = 0.0015$)
(e) Vanilla DSM ($\alpha = \beta = 0.0$)
(f) FP-Diffusion (with $\alpha = 0.0015$)

More Than This…

- Relating to conservative field (or energy-based method)
- Using PINNs for solving scores...?
- …

Let’s catch up offline if you’re interested in this!

Making Diffusion Model More Powerful



Motivation of CTM

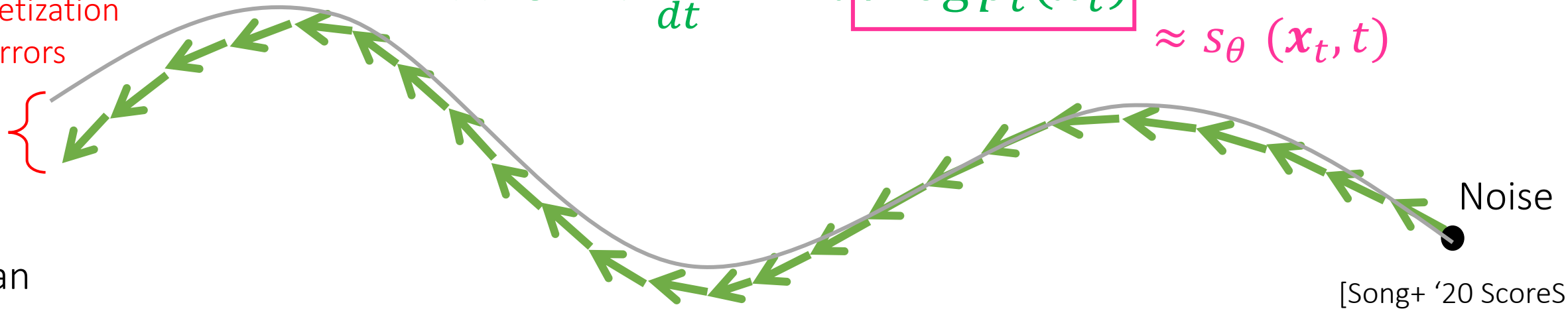
Generated new image

Gradually transforming noise to a new image

PF ODE: $\frac{dx_t}{dt} = -t\nabla \log p_t(\boldsymbol{x}_t)$

$\approx s_\theta\ (\boldsymbol{x}_t, t)$

Discretization errors



[Song+ '20 ScoreSDE]

Motivation of CTM

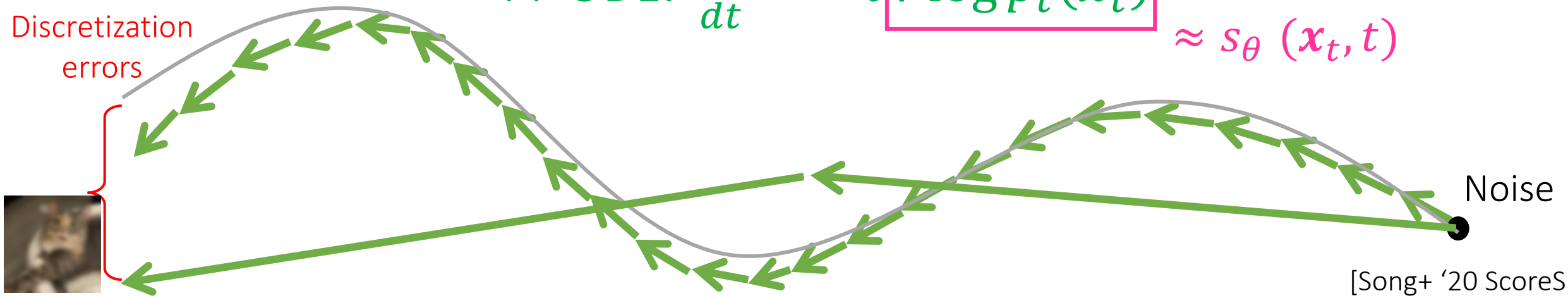
Generated new image

Gradually transforming noise to a new image

PF ODE: $\frac{dx_t}{dt} = -t\nabla \log p_t(\boldsymbol{x}_t)$

$\approx s_\theta\ (\boldsymbol{x}_t, t)$

Discretization errors



[Song+ '20 ScoreSDE]

Motivation of CTM

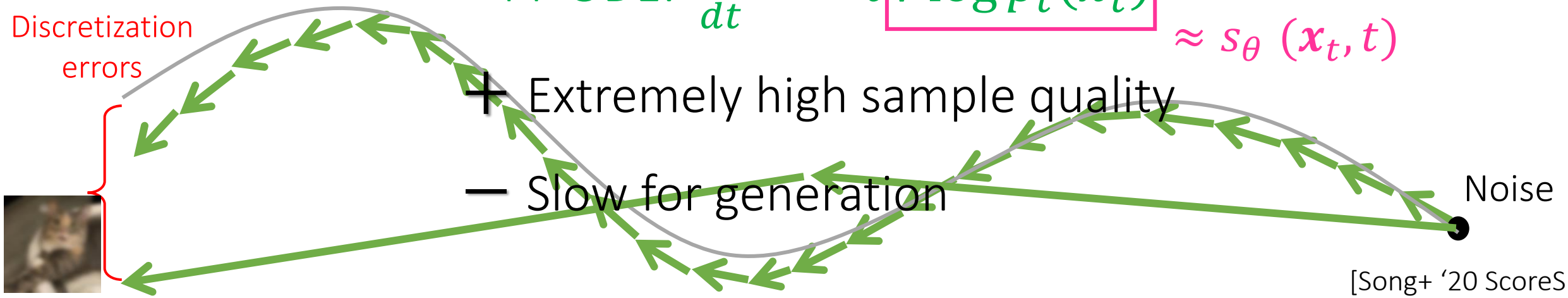
Generated new image

Gradually transforming noise to a new image

PF ODE: $\frac{dx_t}{dt} = -t\nabla \log p_t(\boldsymbol{x}_t)$

$\approx s_\theta\ (\boldsymbol{x}_t, t)$

Discretization errors



\+ Extremely high sample quality

— Slow for generation

[Song+ '20 ScoreSDE]

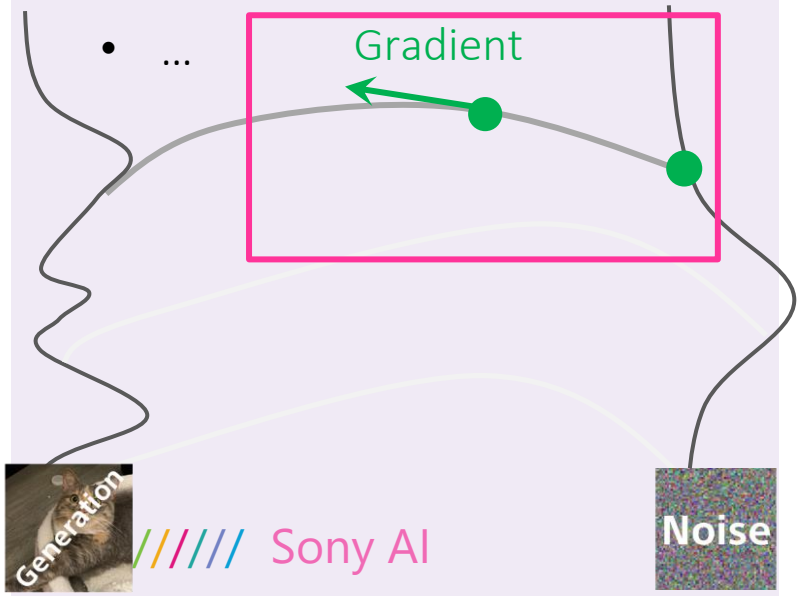
Fast Sampling in Diffusion Model

PF-ODE (2021)

Training-free

Sophisticated solvers (2021-2022+)

- [ICLR'21] DDIM
- [NeurIPS'22] DPM; DPM++
- [ICLR'23] DEIS
- …



////// Sony AI

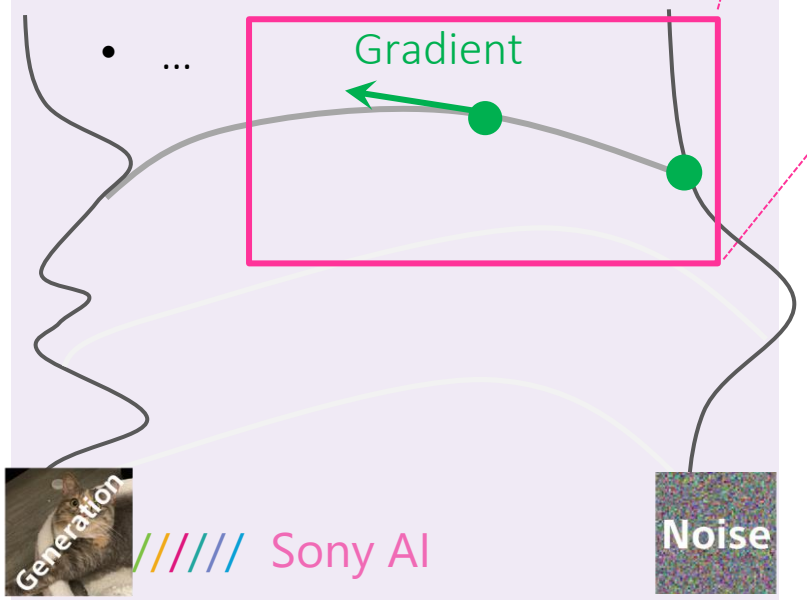
Fast Sampling in Diffusion Model

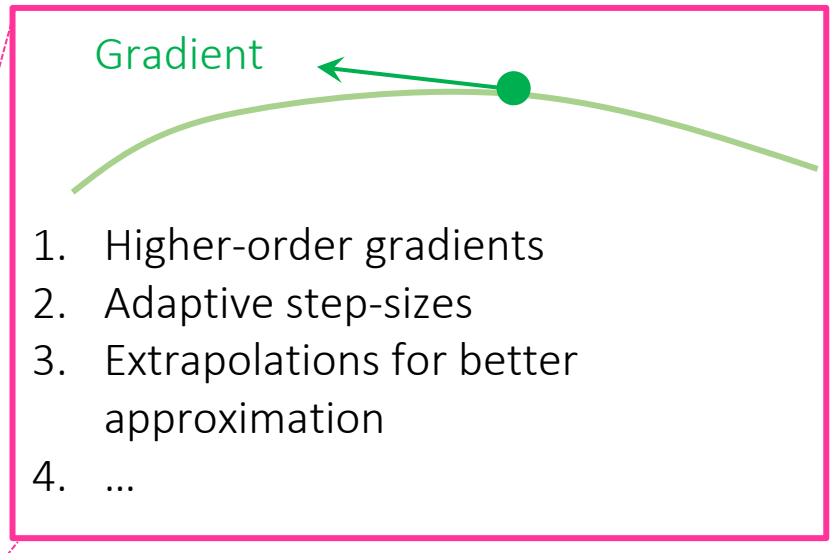
PF-ODE (2021)

Training-free

Sophisticated solvers (2021-2022+)

- [ICLR'21] DDIM
- [NeurIPS'22] DPM; DPM++
- [ICLR'23] DEIS
- …





1. Higher-order gradients
2. Adaptive step-sizes
3. Extrapolations for better approximation
4. …

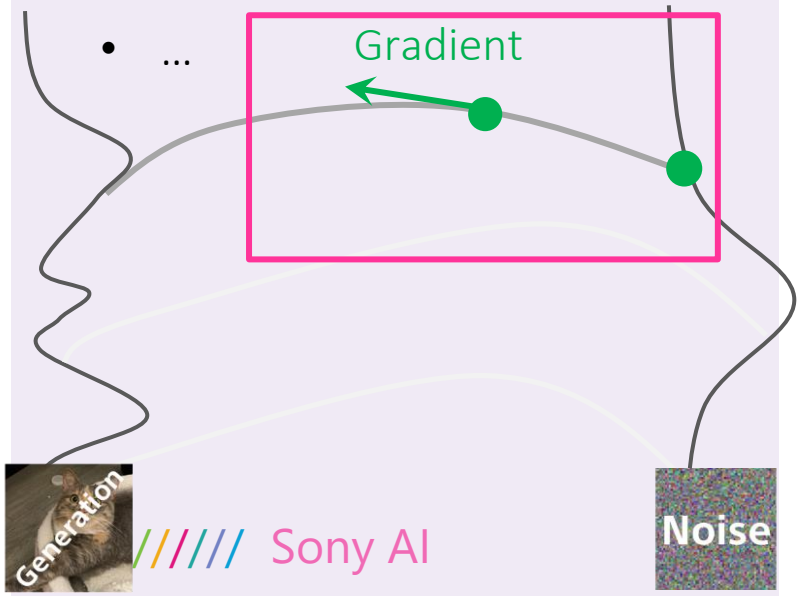
Fast Sampling in Diffusion Model

PF-ODE (2021)

Training-free

Sophisticated solvers (2021-2022+)

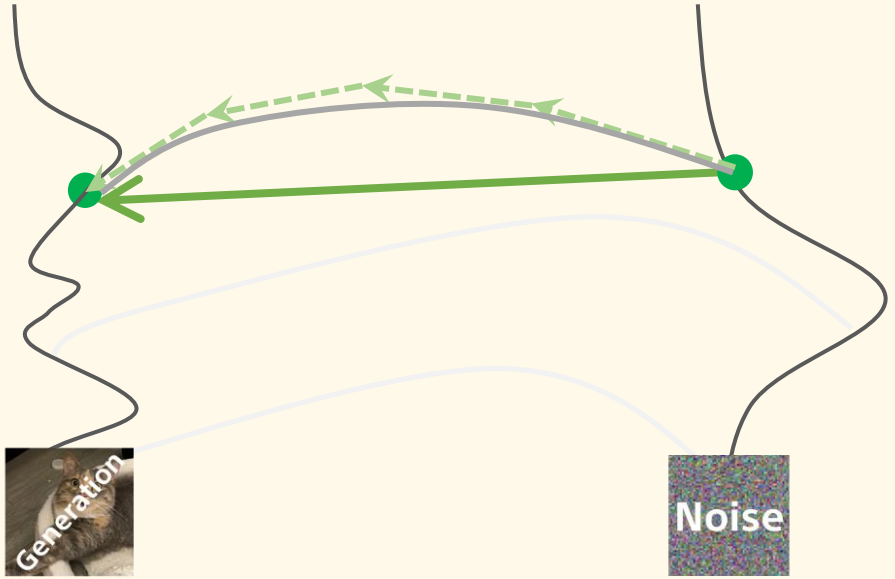
- [ICLR'21] DDIM
- [NeurIPS'22] DPM; DPM++
- [ICLR'23] DEIS
- …



Training-based

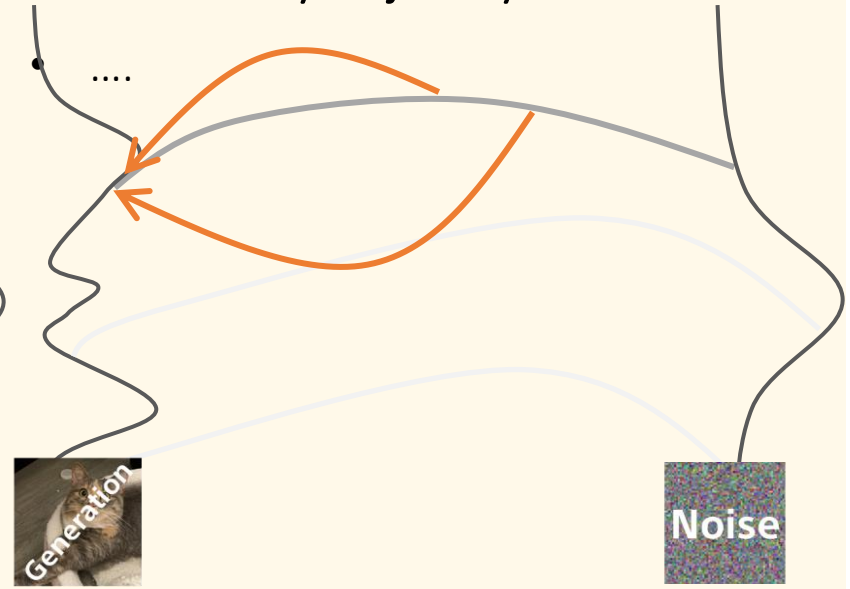
Looking for more optimal trajectories for sampling (2023+)

- [ICLR'23] Rectified Flow
- [ICLR'23] Flow Matching



Distillation from DM (2023-)

- [ICML'23 Song+] Consistency Model
- [ICLR'24 Kim&Lai+] Consistency Trajectory Model
- ….



CONSISTENCY TRAJECTORY MODEL

CTM

istency Trajectory Models: Learning
ility Flow ODE Trajectory of Diffusion

Dongjun Kim*,♨1, Chieh-Hsin Lai*,1,
i-Hsiang Liao[1], Naoki Murata[1], Yuhta Takida[1], Toshimitsu Uesaka[1], Yutong He♨1,3,
Yuki Mitsufuji[1,2], Stefano Ermon[4],
ony AI, [2]Sony Group Corporation, [3]Carnegie Mellon University, [4]Stanford University

ICLR 2024

*Equal Contribution (✉ Dongjun Kim; ✉ Chieh-Hsin Lai)

♨Internship at Sony AI

ICLR'24 OpenReview | arXiv | Codes

Diffusion → Distillation

PF-ODE (2021)

$$d\boldsymbol{x}_u = -u\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u)$$

Diffusion Model

- Training:

$$\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u) \approx \boldsymbol{s}_{\boldsymbol{\phi}}(\boldsymbol{x}_u, u)$$

- Sampling:

$$\boldsymbol{x}_0^{\text{oracle}} = \boldsymbol{x}_T - \int_T^0 u\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u) du$$

$$\approx \boldsymbol{x}_T - \int_T^0 u\boldsymbol{s}_{\boldsymbol{\phi}}(\boldsymbol{x}_u, u) du$$

Distillation-Based Model

- Training:

$$F^{\text{oracle}}(\boldsymbol{x}_t, t) = \boldsymbol{x}_t - \int_t^0 u\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u)\ du$$

Diffusion → Distillation

PF-ODE (2021)

$$d\boldsymbol{x}_u = -u\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u)$$

Diffusion Model

- Training:

$$\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u) \approx \boldsymbol{s}_{\boldsymbol{\phi}}(\boldsymbol{x}_u, u)$$

- Sampling:

$$\boldsymbol{x}_0^{\text{oracle}} = \boldsymbol{x}_T - \int_T^0 u\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u)du$$

$$\approx \boldsymbol{x}_T - \int_T^0 u\boldsymbol{s}_{\boldsymbol{\phi}}(\boldsymbol{x}_u, u)du$$

Distillation-Based Model

- Training:

$$F^{\text{oracle}}(\boldsymbol{x}_t, t) \approx \boxed{\boldsymbol{x}_t - \int_t^0 u\, \boldsymbol{s}_{\boldsymbol{\phi}}^{\text{PreTrained}}(\boldsymbol{x}_u, u)\, du}$$

$$\approx \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

- Sampling: few steps!

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_T, T)$$

This direction is drastically blooming since 2023

PF-ODE (2021)

$$d\boldsymbol{x}_u = -u\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u)$$

Diffusion Model

- Training:

$$\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u) \approx \boldsymbol{s}_{\boldsymbol{\phi}}(\boldsymbol{x}_u, u)$$

- Sampling:

$$\boldsymbol{x}_0^{\text{oracle}} = \boldsymbol{x}_T - \int_T^0 u\nabla_{\boldsymbol{x}} \log p_u^{\text{oracle}}(\boldsymbol{x}_u)du$$

$$\approx \boldsymbol{x}_T - \int_T^0 u\boldsymbol{s}_{\boldsymbol{\phi}}(\boldsymbol{x}_u, u)du$$

Distillation-Based Model

- Training:

$$F^{\text{oracle}}(\boldsymbol{x}_t, t) \approx \boldsymbol{x}_t - \int_t^0 u\, \boldsymbol{s}_{\boldsymbol{\phi}}^{\text{PreTrained}}(\boldsymbol{x}_u, u)\, du$$

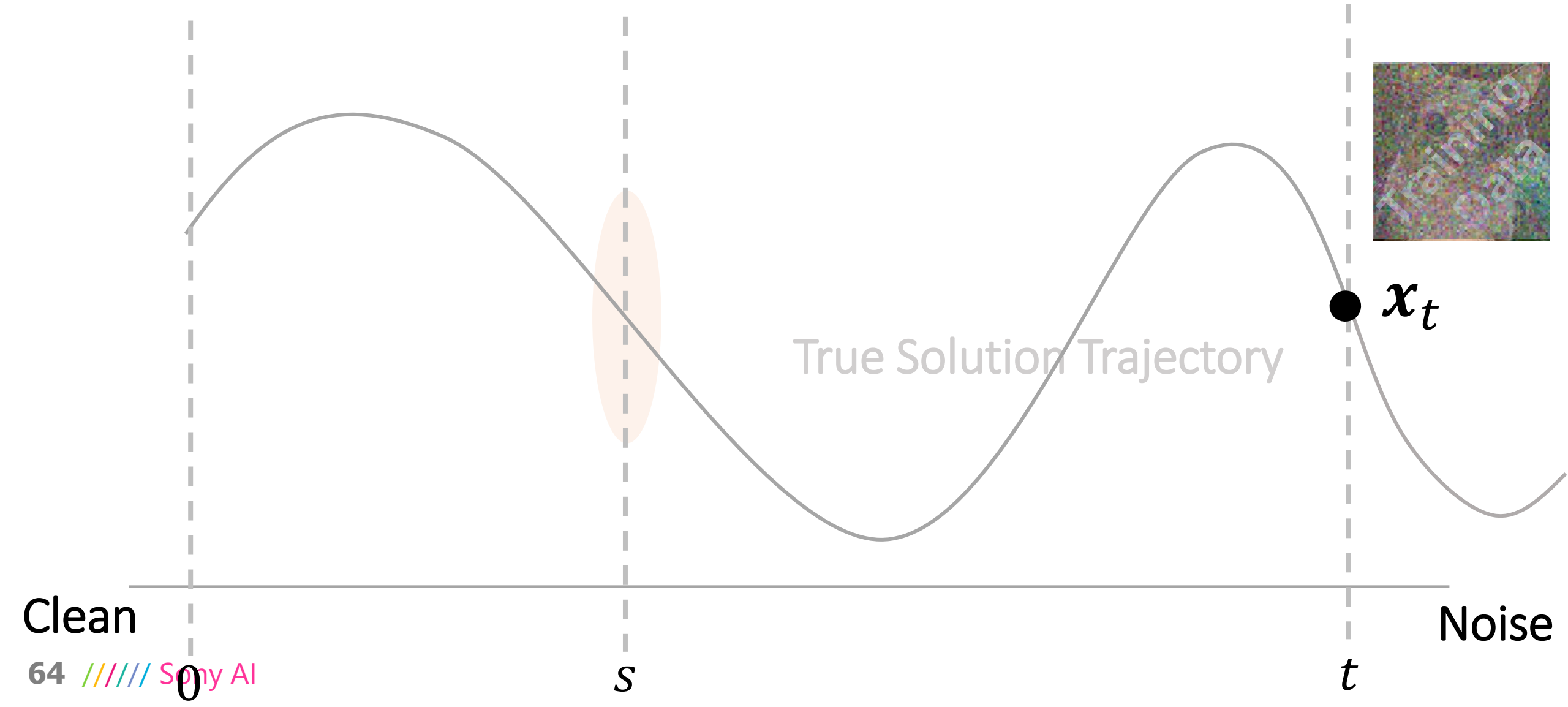
$$\approx \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

- Sampling: few steps!

$$\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}_T, T)$$

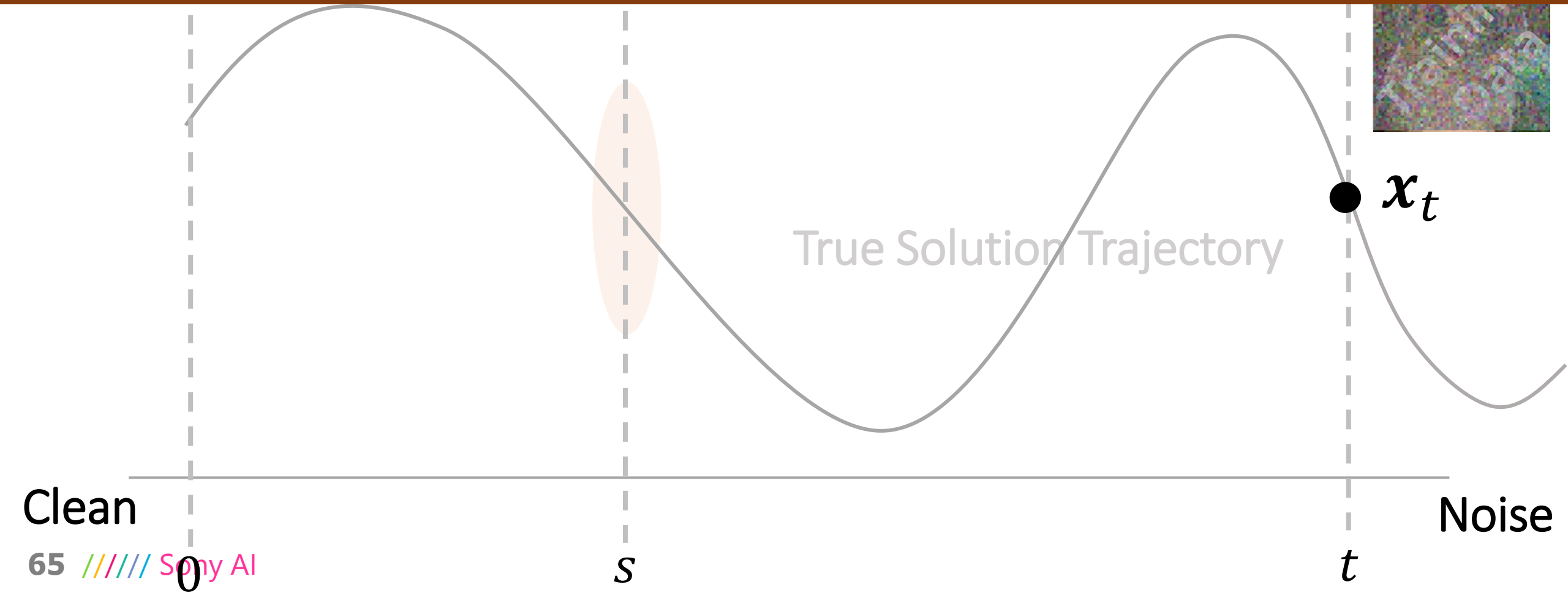
CTM’s Training and Mechanism

Consistency Trajectory Model (CTM)'s Training



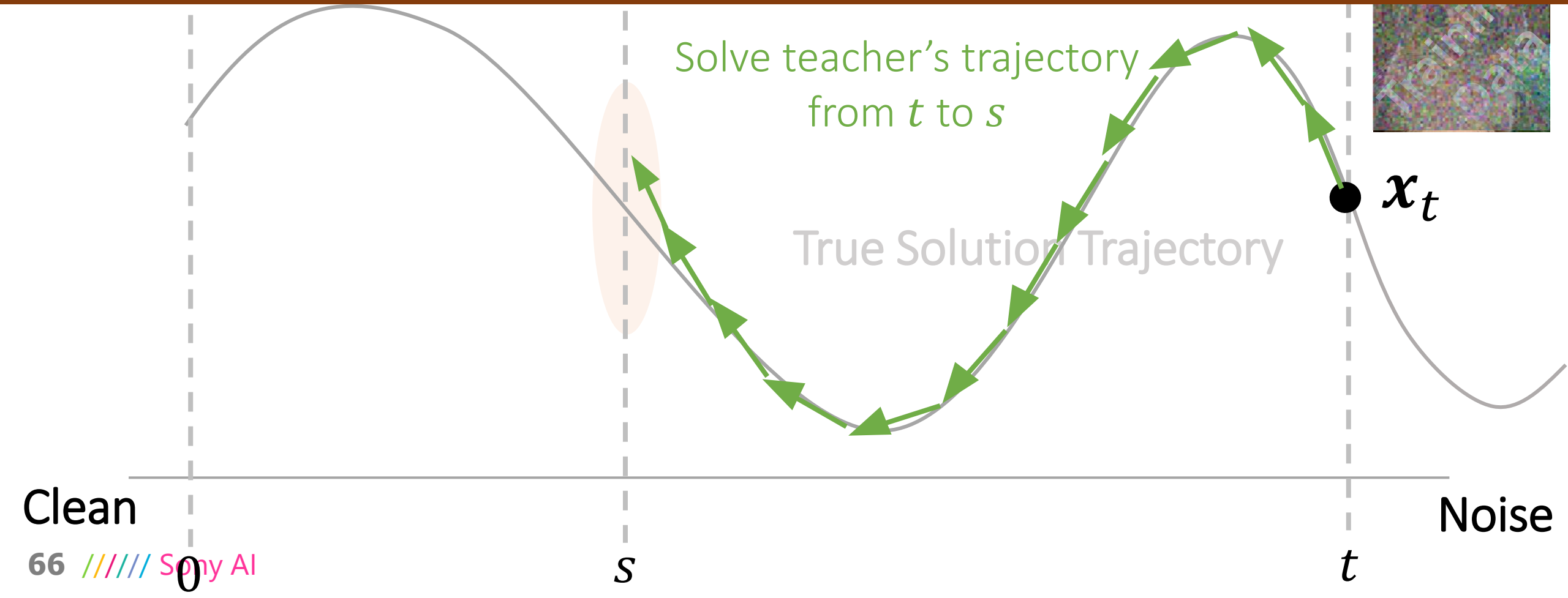
CTM learns any-to-any jump on the trajectory

$$\boldsymbol{x}_t - \int_t^s \tau \nabla \log p_\tau(\boldsymbol{x}_\tau) d\tau$$



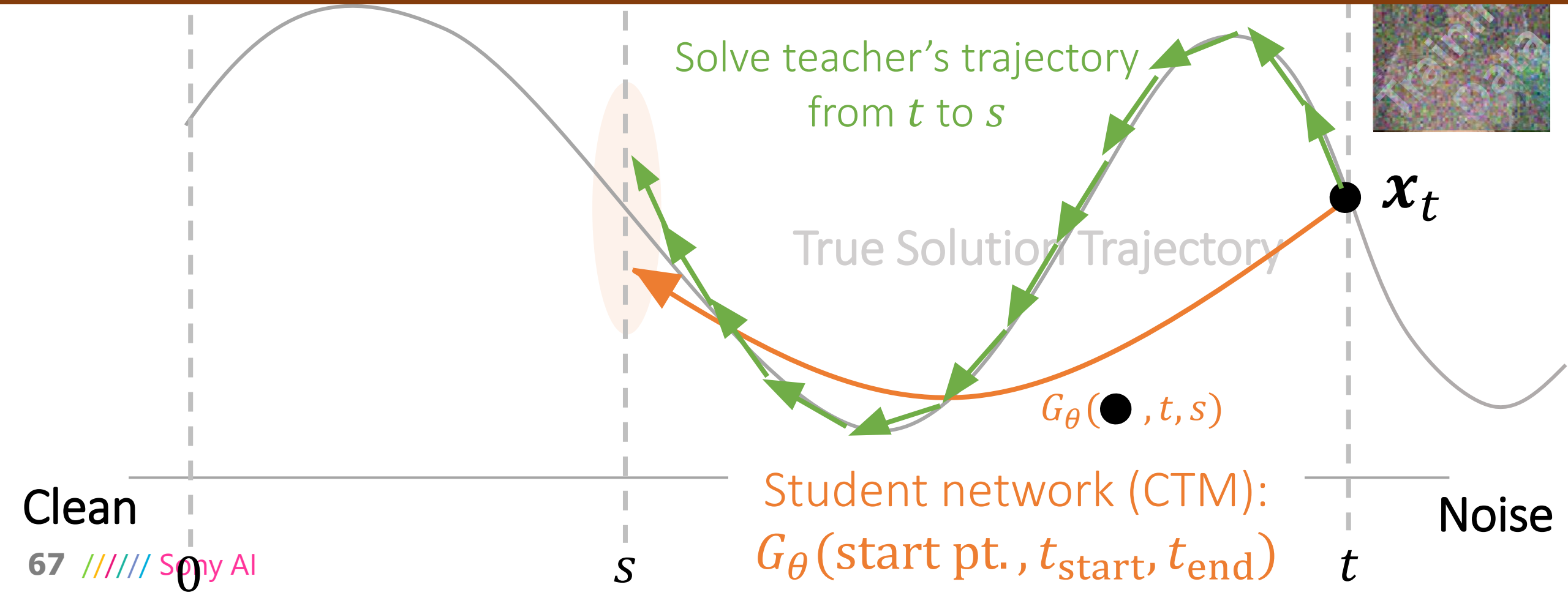
CTM learns any-to-any jump on the trajectory

$$\boldsymbol{x}_t - \int_t^s \tau \boldsymbol{s}_{\text{teacher}}(\boldsymbol{x}_\tau, \tau) d\tau$$



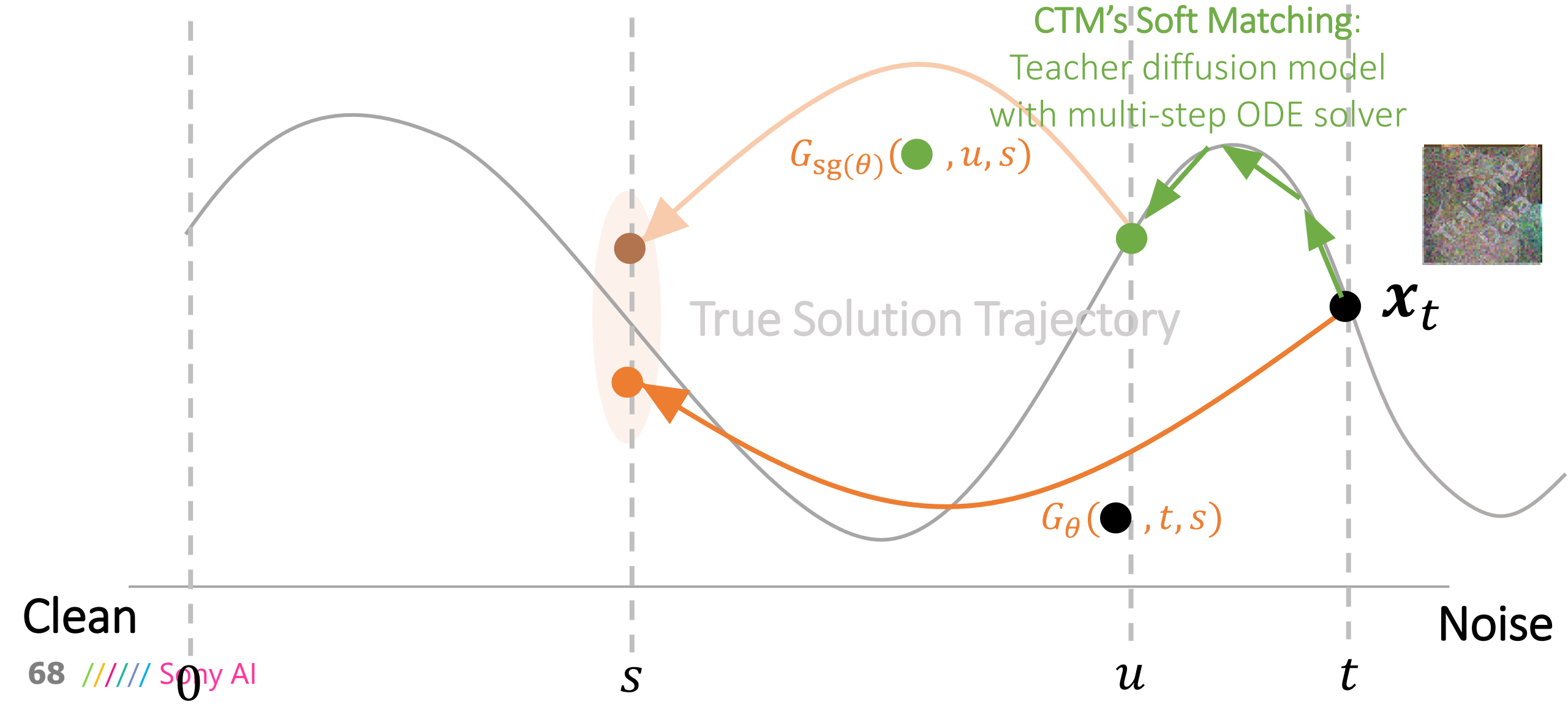
CTM learns any-to-any jump on the trajectory

$$G_\theta(\boldsymbol{x}_t, t, s) \approx \boldsymbol{x}_t - \int_t^s \tau \boldsymbol{s}_{\text{teacher}}(\boldsymbol{x}_\tau, \tau) d\tau$$



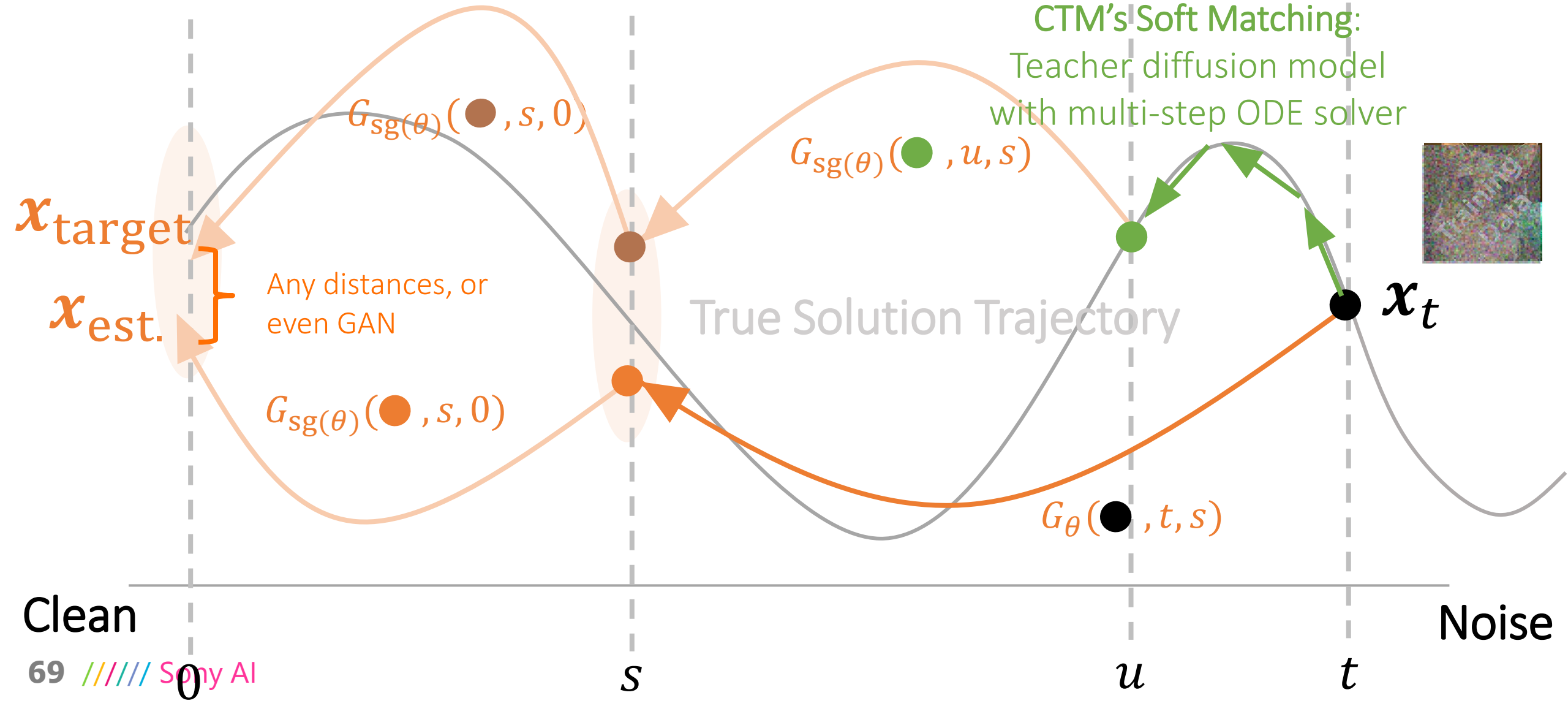
Consistency Trajectory Model (CTM)'s Training

Student network (CTM): $G_\theta(\text{start pt.}, t_{\text{start}}, t_{\text{end}})$



Consistency Trajectory Model (CTM)'s Training

Student network (CTM): $G_\theta(\text{start pt.}, t_{\text{start}}, t_{\text{end}})$

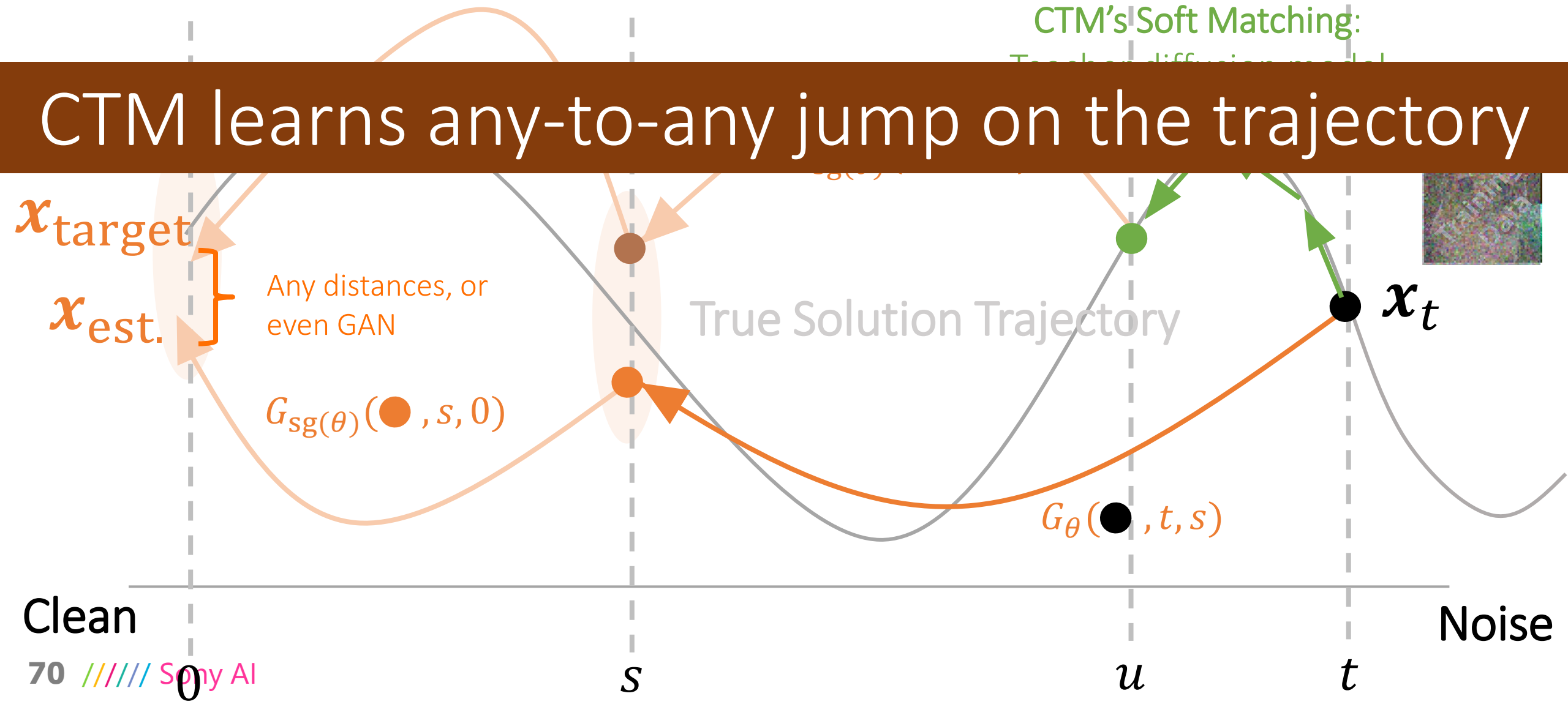


Consistency Trajectory Model (CTM)'s Training

Student network (CTM): $G_\theta(\text{start pt.}, t_{\text{start}}, t_{\text{end}})$

CTM's Soft Matching:

CTM learns any-to-any jump on the trajectory



Consistency Trajectory Model (CTM)'s Training

Student network (CTM): $G_\theta(\text{start pt.}, t_{\text{start}}, t_{\text{end}})$

CTM's Soft Matching:

CTM learns any-to-any jump on the trajectory

x_{target}

$x_{\text{est.}}$

Any distances, or even GAN

True Solution Trajectory

x_t

$\mathcal{L}_{\text{CTM}} + \mathcal{L}_{\text{DSM}}$

$G_\theta(\bullet, t, s)$

Clean

Noise

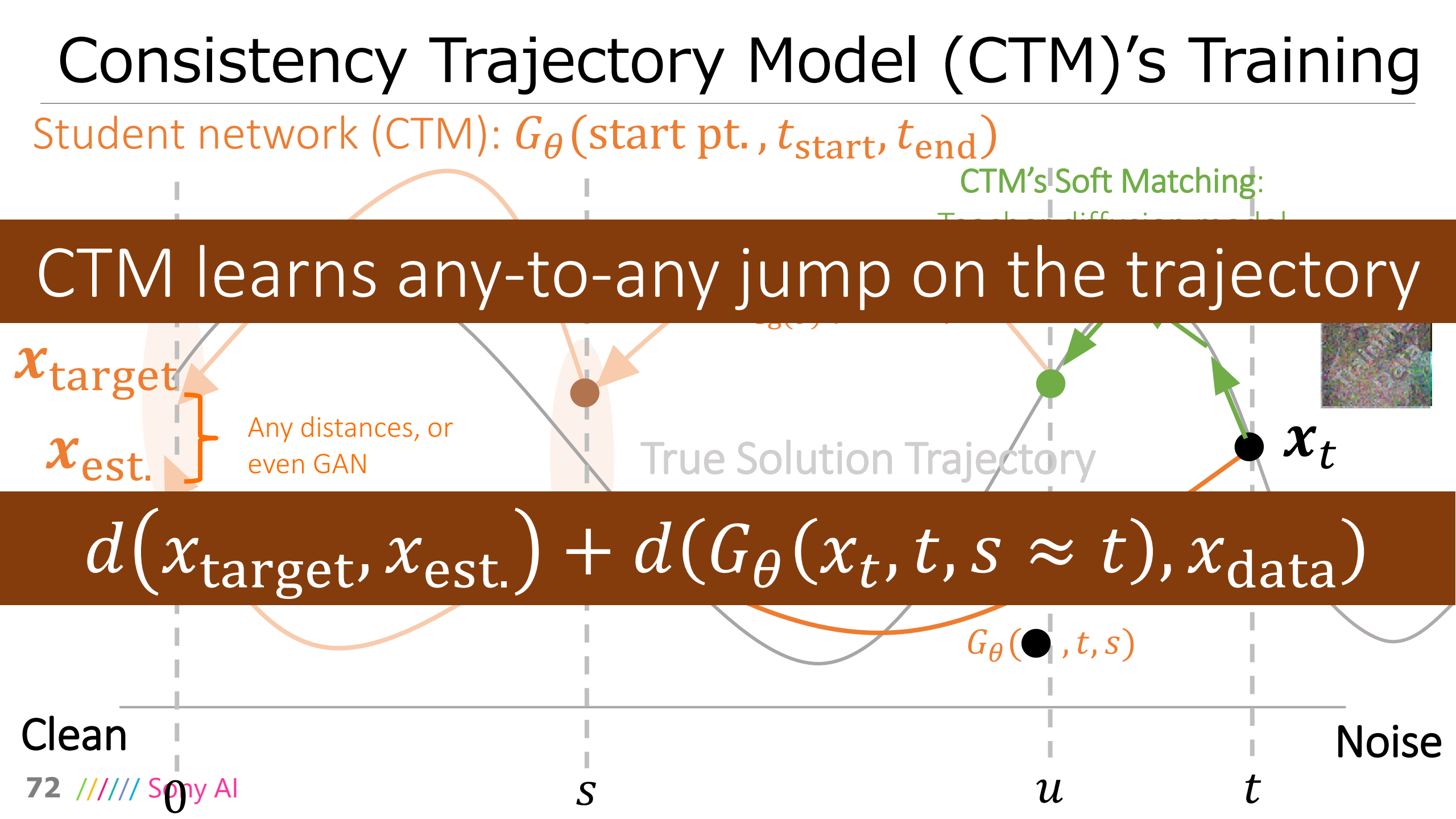
0 s u t

Consistency Trajectory Model (CTM)'s Training

Student network (CTM): $G_\theta(\text{start pt.}, t_{\text{start}}, t_{\text{end}})$

CTM's Soft Matching:

CTM learns any-to-any jump on the trajectory



$$d\left(x_{\text{target}}, x_{\text{est.}}\right) + d\left(G_\theta(x_t, t, s \approx t), x_{\text{data}}\right)$$

CTM’s flexibility in sampling

CTM's Sampling Flexibility

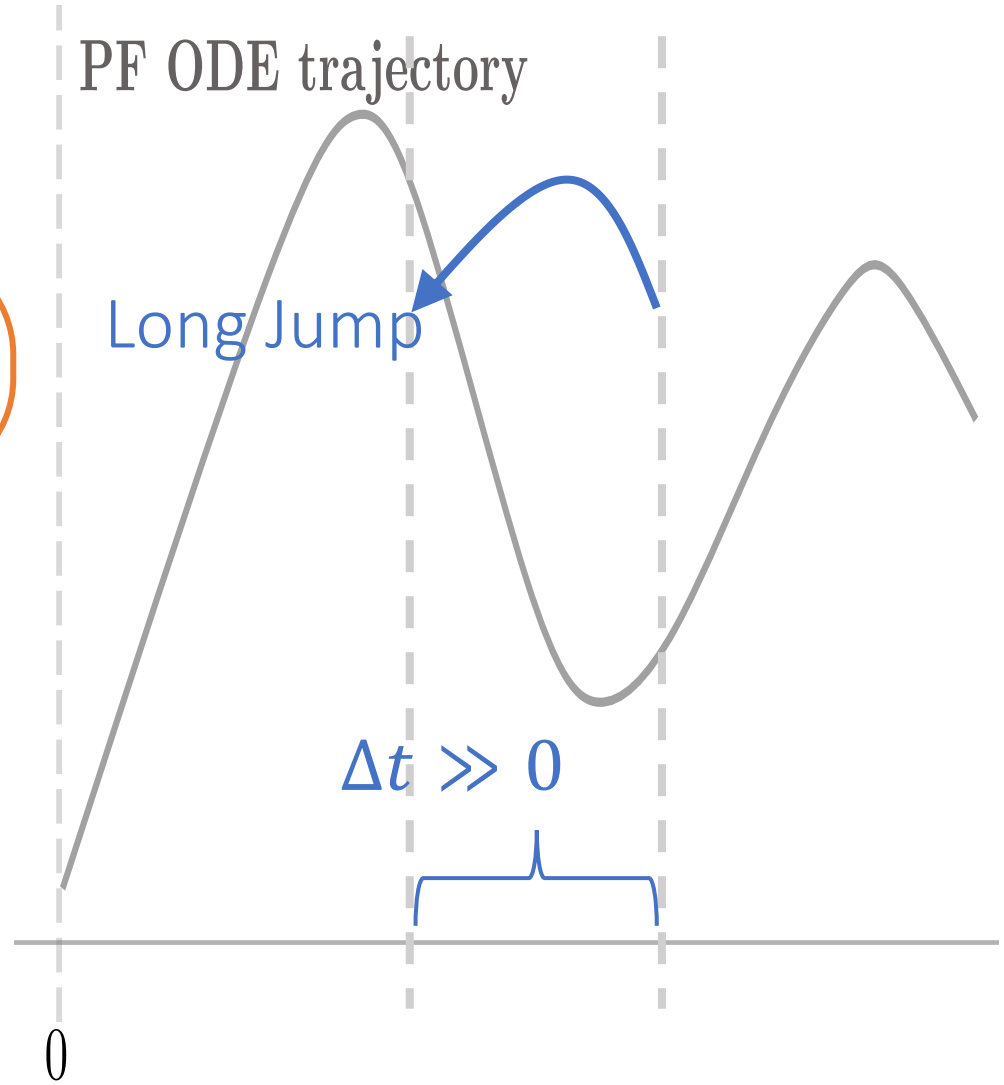
CTM enables both long "jumps" along the solution trajectory and score evaluation!

$$G_\theta(\boldsymbol{x}_t, t, t-\Delta t)$$

Long jump ($\Delta t \gg 0$)

CTM enables long jump along the trajectory

→ Allow our new sampling method



CTM's Sampling Flexibility

CTM enables both long "jumps" along the solution trajectory and score evaluation!

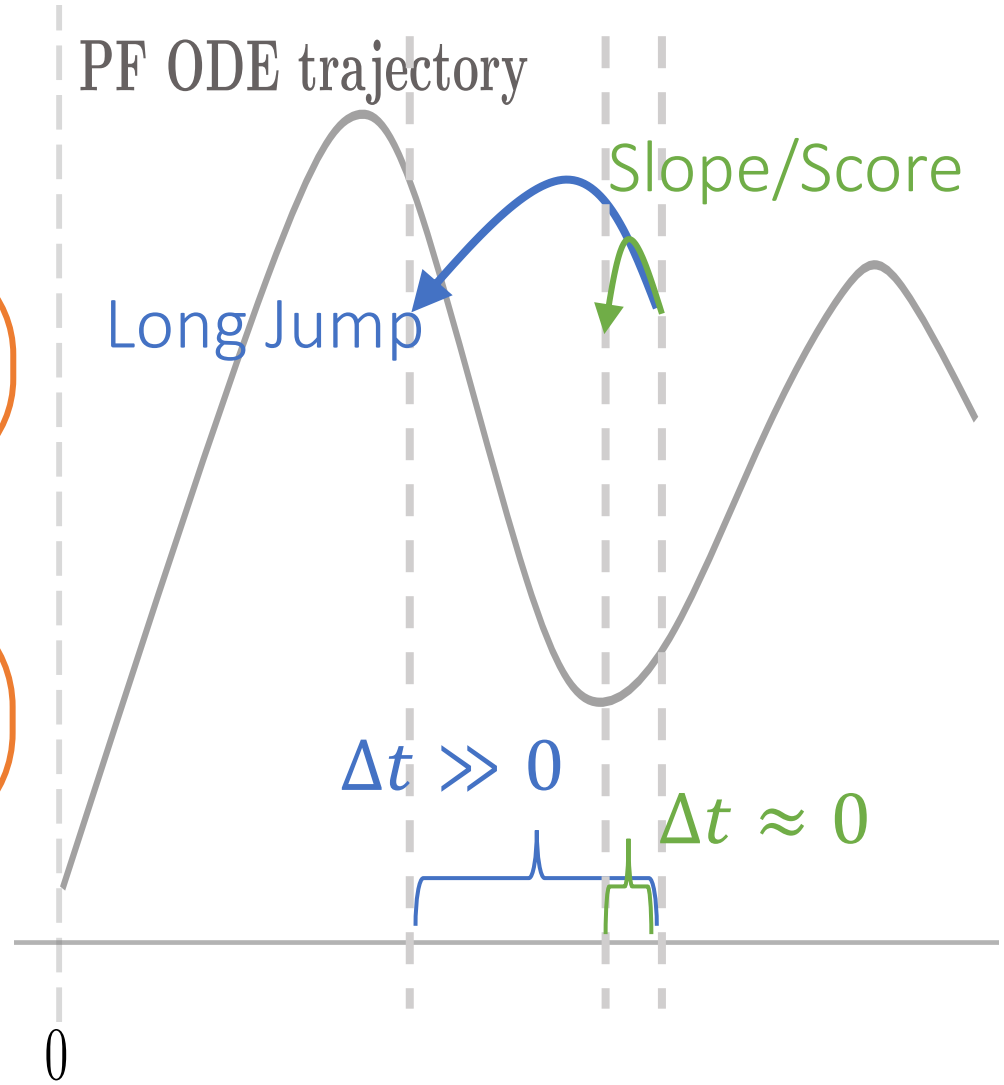
$$G_\theta(\boldsymbol{x}_t, t, t-\Delta t)$$

Long jump ($\Delta t \gg 0$)

CTM enables long jump along the trajectory
→ Allow our new sampling method

Slope (infinitesimal jump $\Delta t \approx 0$)

CTM enables score evaluations
→ Allow score-based sampling & likelihood computation



One-step Generation with CTM

Solution Trajectory

$$G_\theta(\ \text{Noise}\ , t_{\text{start}} = T,\ t_{\text{end}} = 0)$$

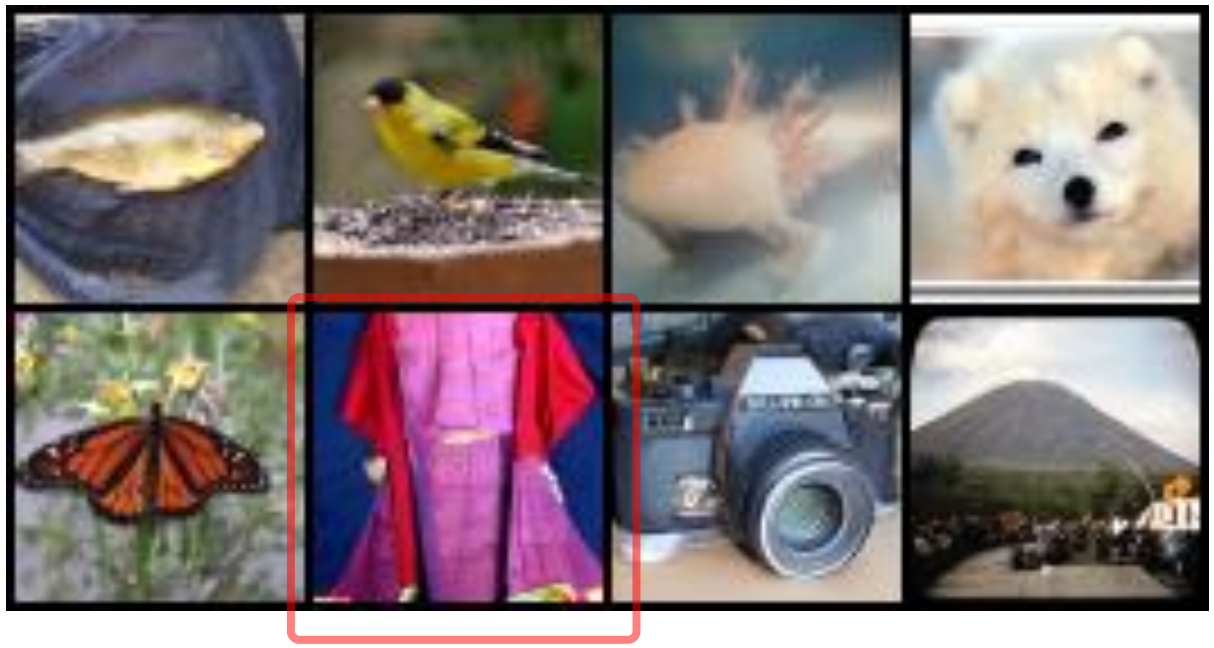
Clean

Noise

0

T

SOTA (beats DM) with One-Step



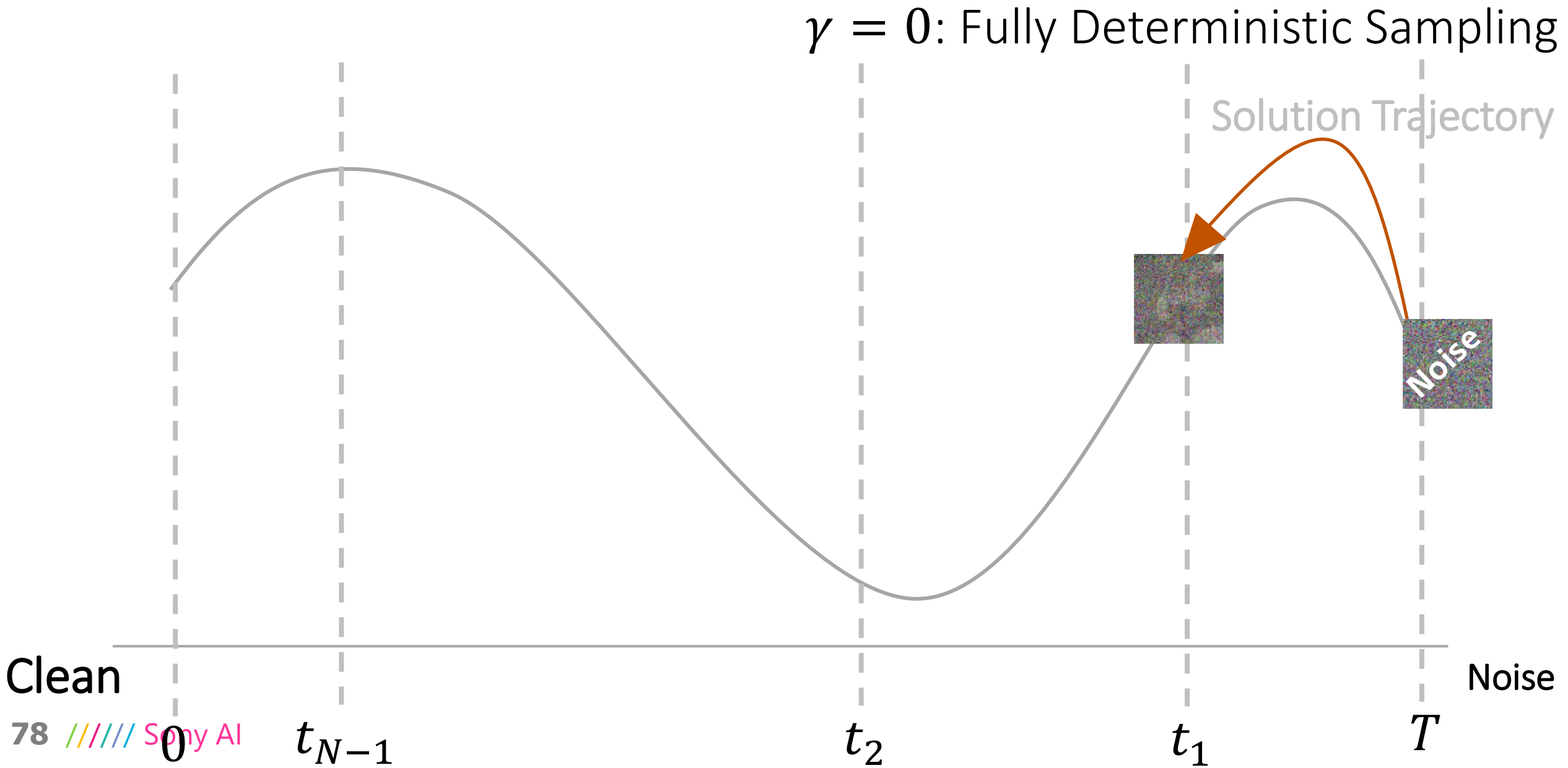
Teacher: Sampling steps = 79

Takes minutes

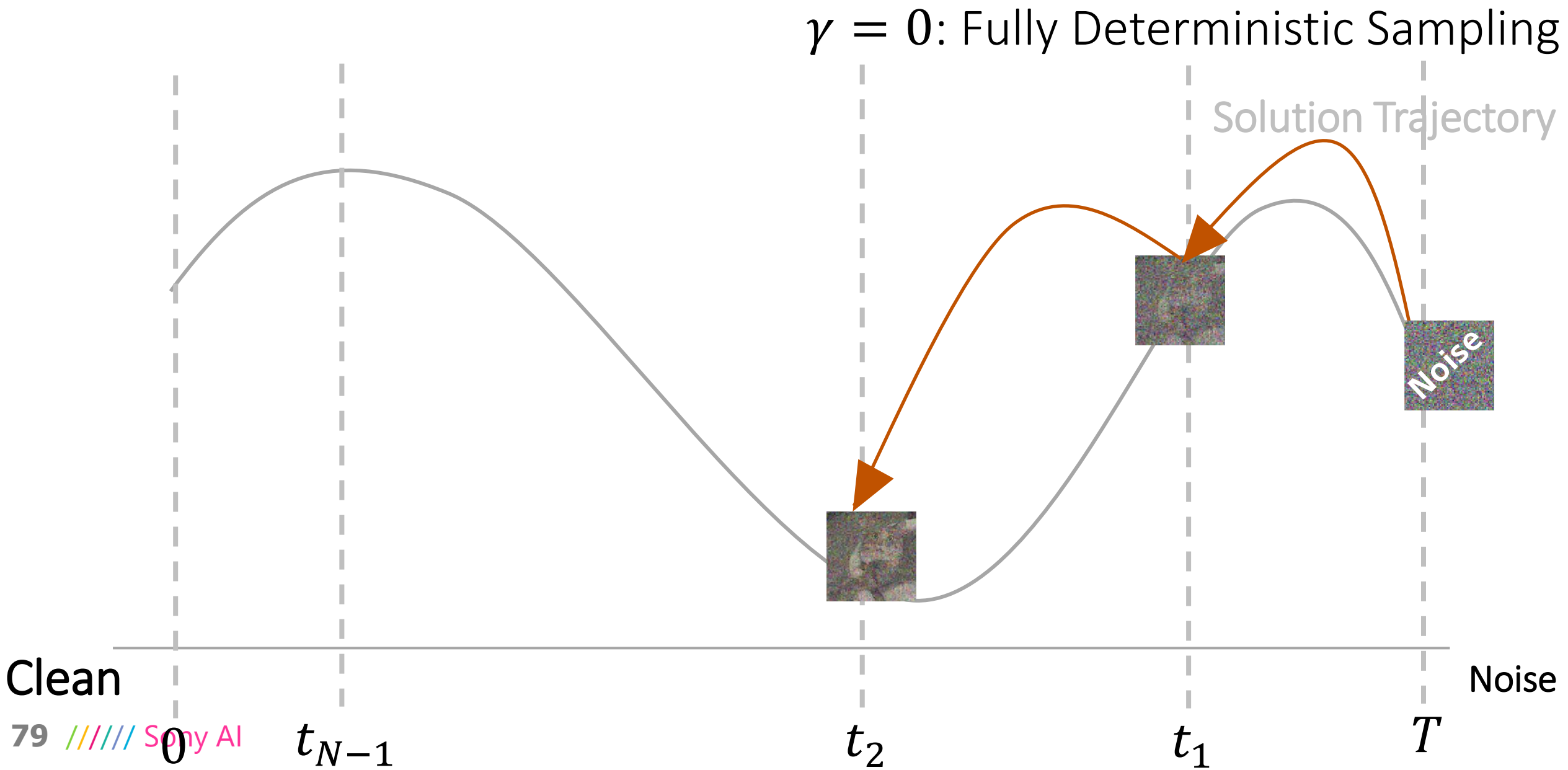
Student (CTM): Sampling step = 1

Within a second

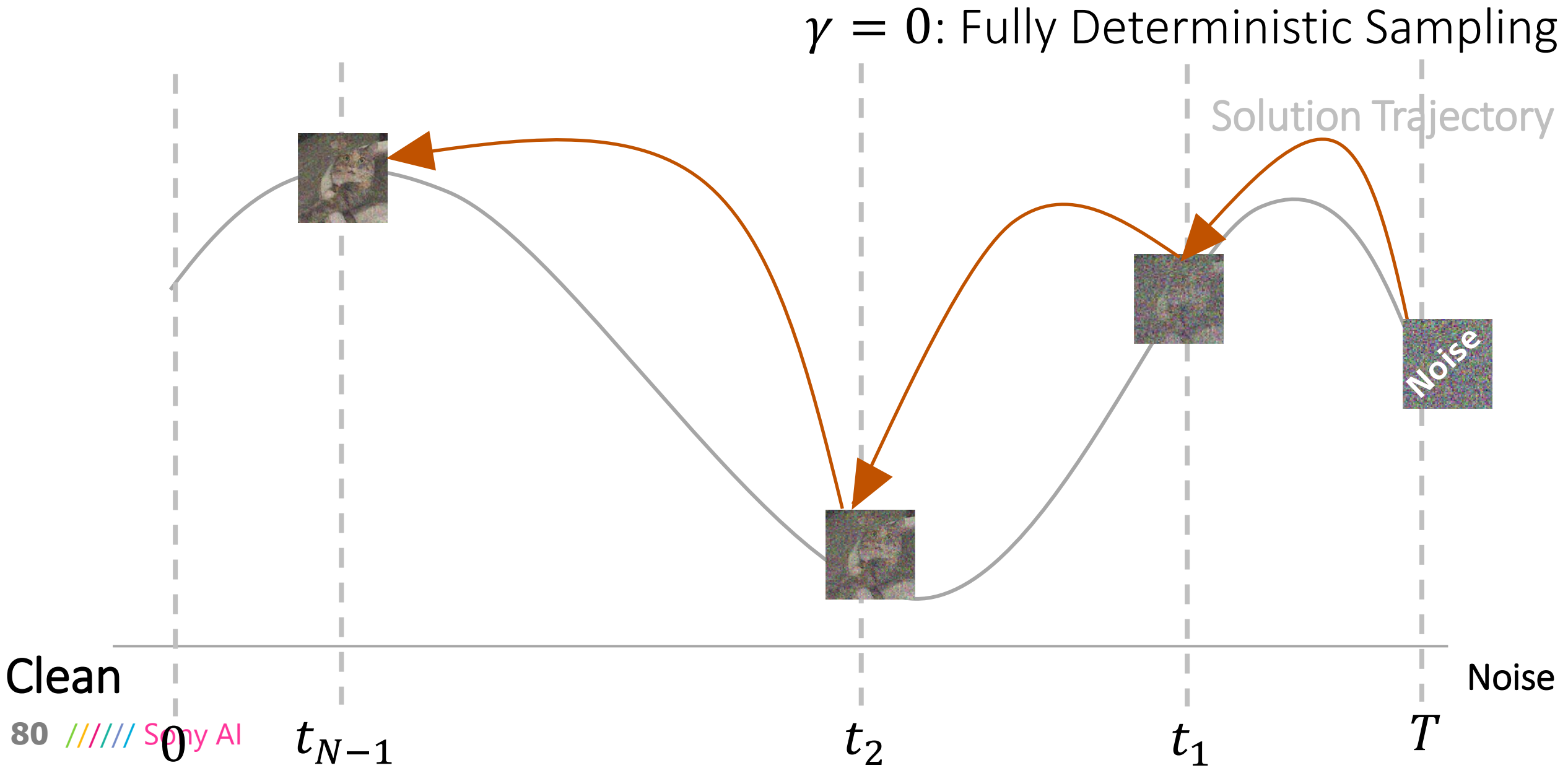
Multistep Generation with CTM: γ-sampling



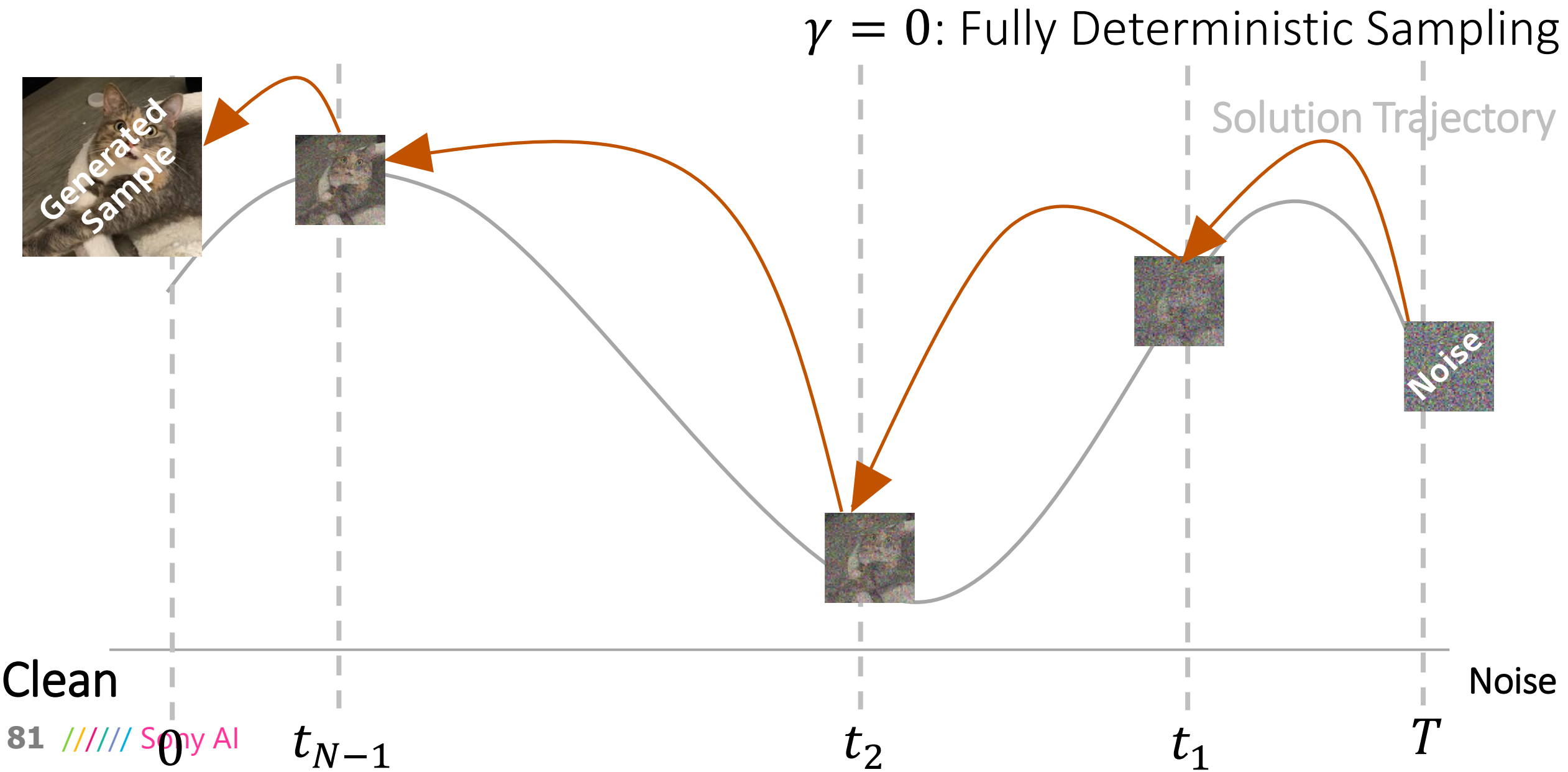
Multistep Generation with CTM: γ-sampling



Multistep Generation with CTM: γ-sampling

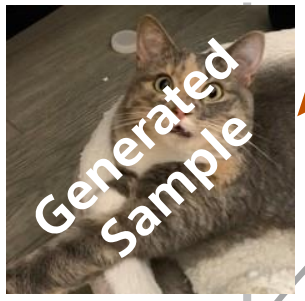


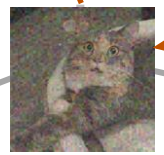
Multistep Generation with CTM: γ-sampling



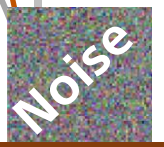
Multistep Generation with CTM: γ-sampling

$\gamma = 0$: Fully Deterministic Sampling





Solution Trajectory



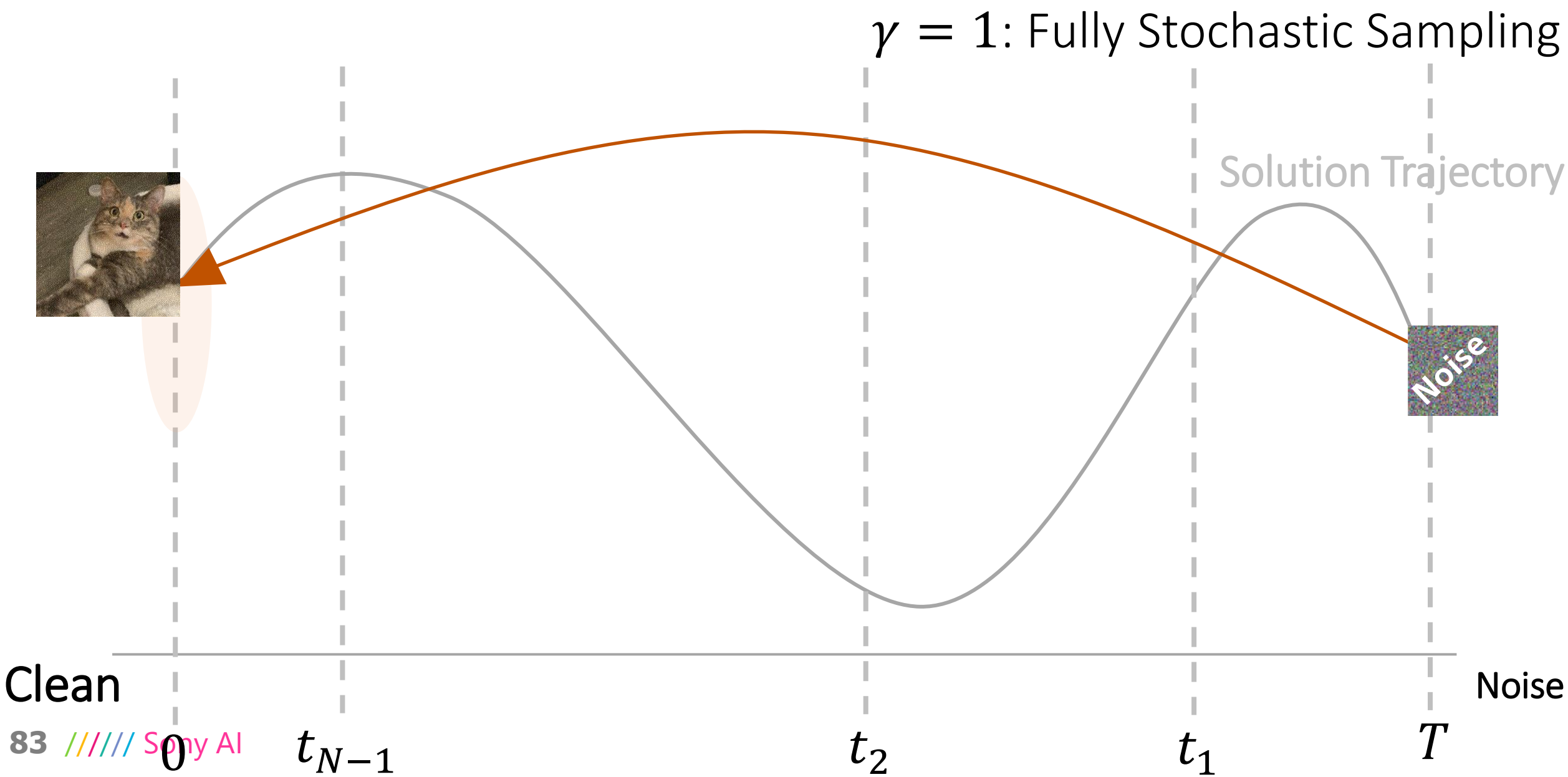
$\mathcal{O}(T)$

Clean

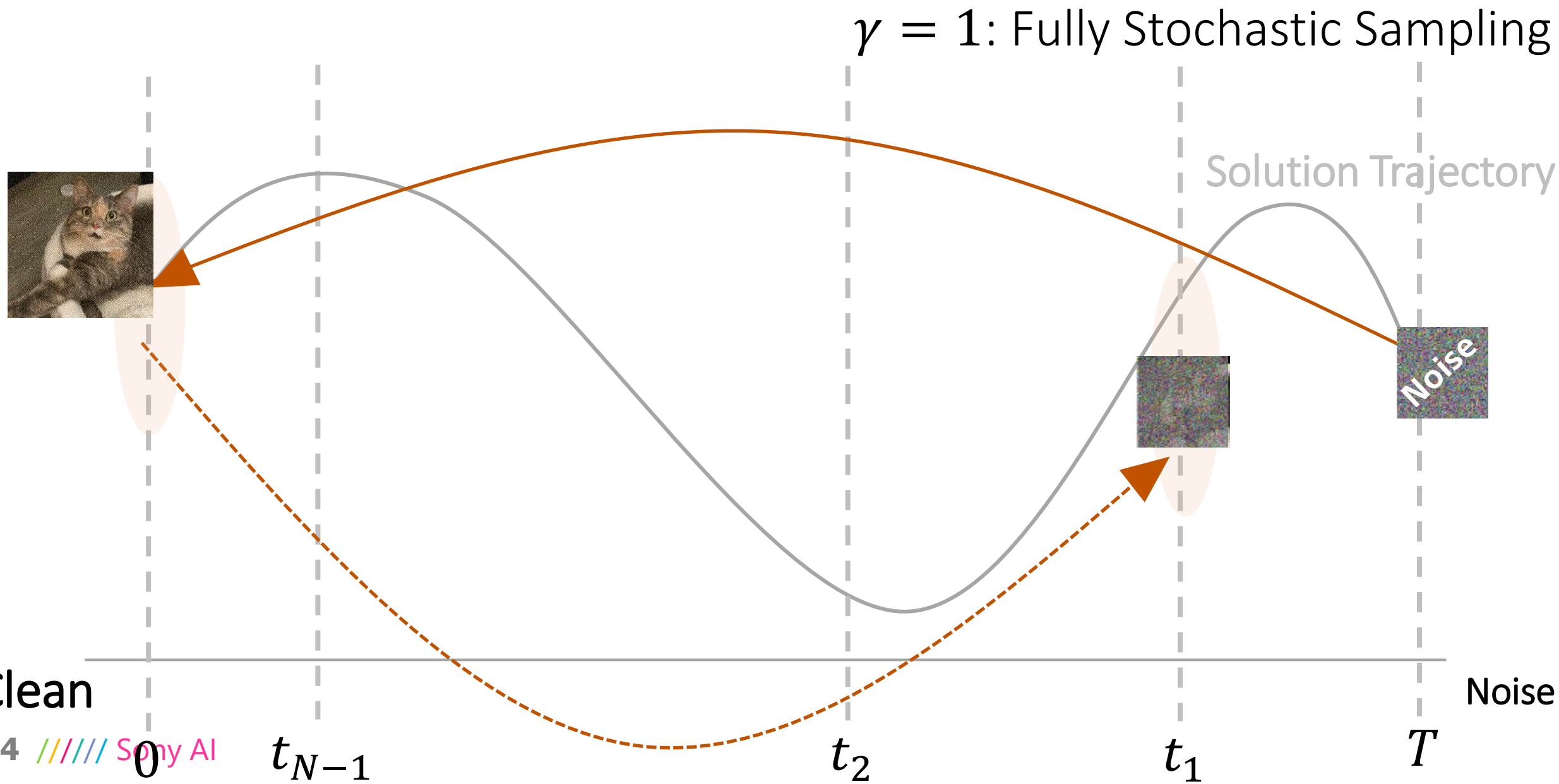
Noise

0 t_{N-1} t_2 t_1 T

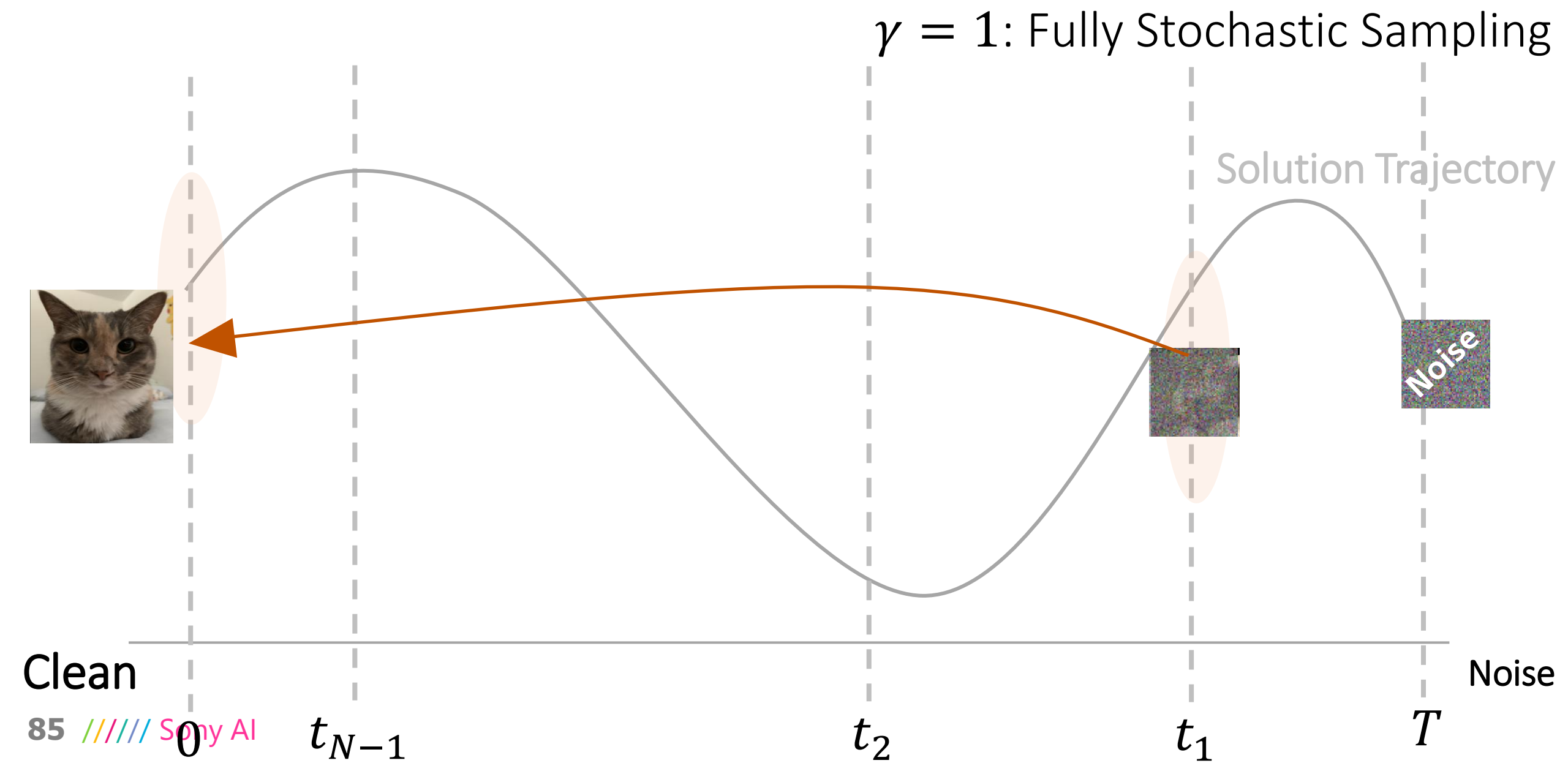
Multistep Generation with CTM: γ-sampling



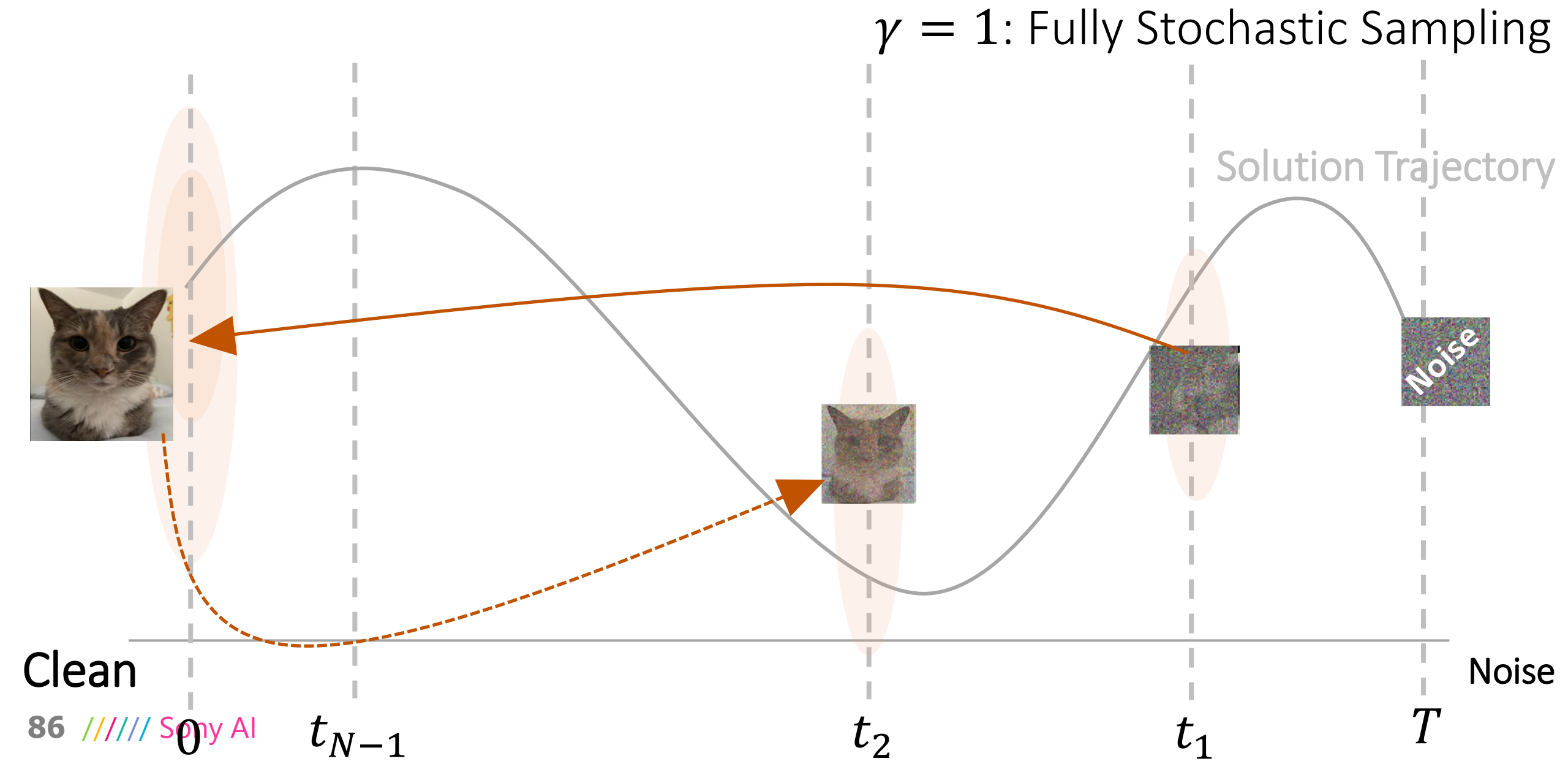
Multistep Generation with CTM: γ-sampling



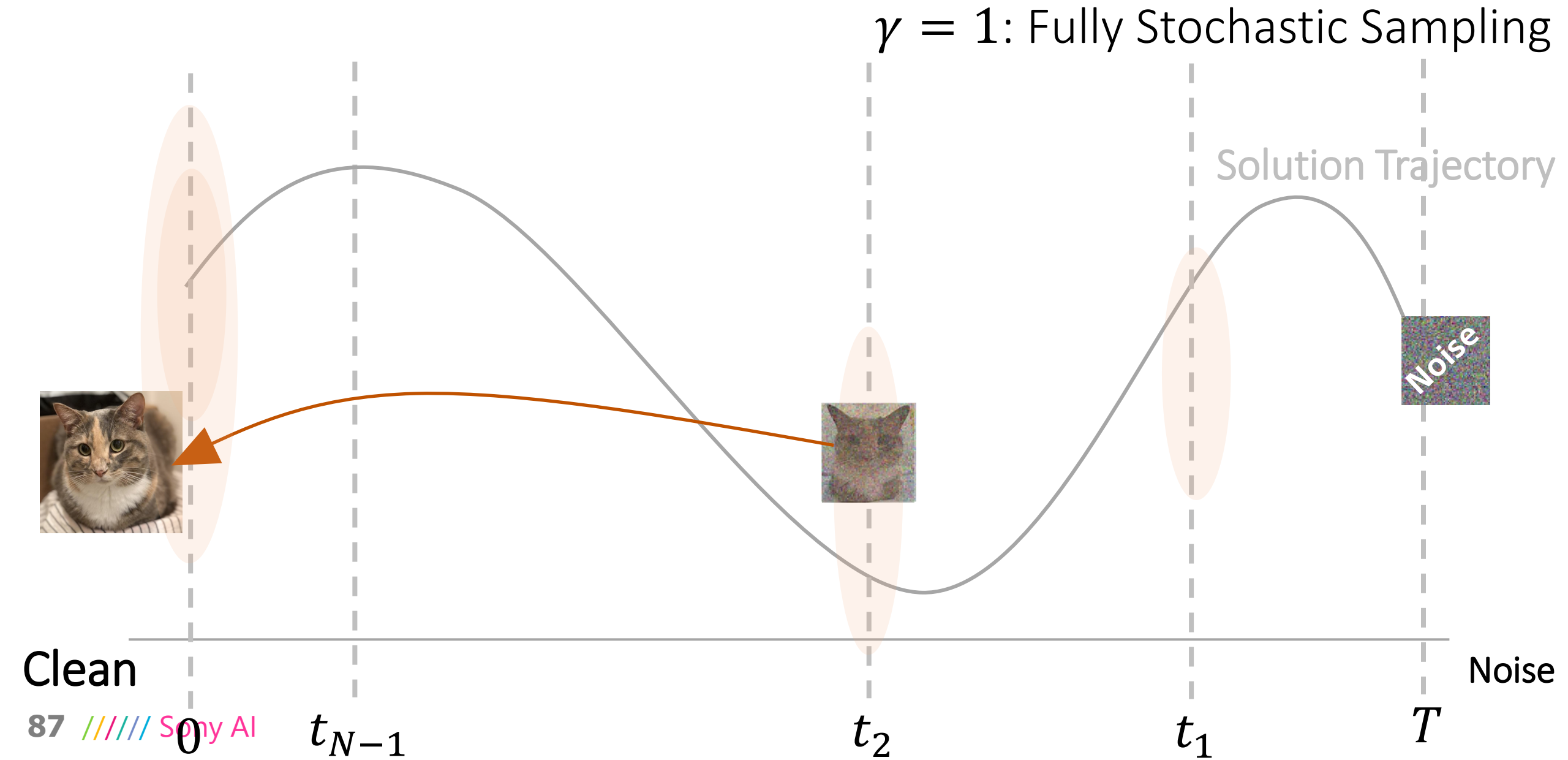
Multistep Generation with CTM: γ-sampling



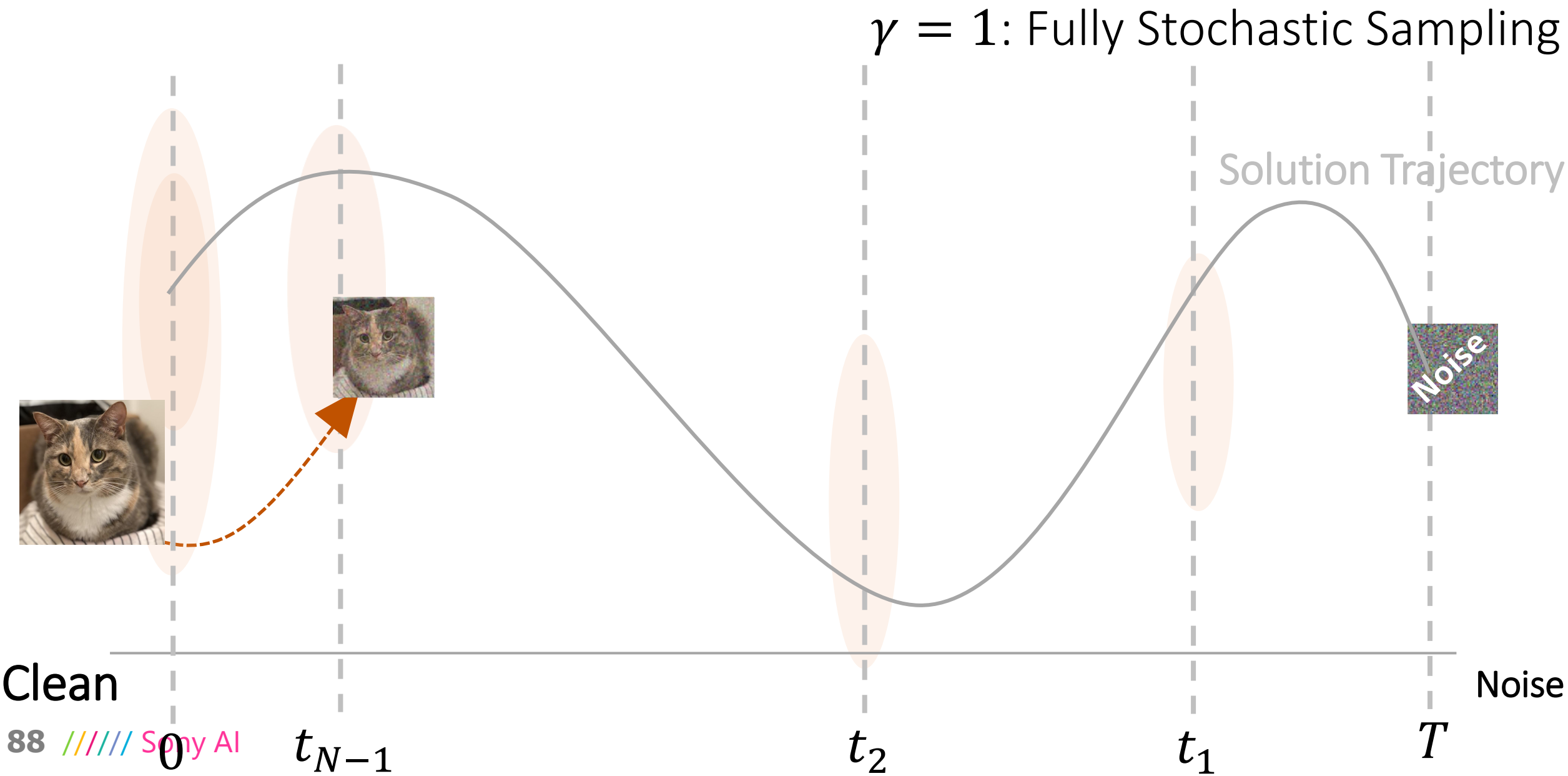
Multistep Generation with CTM: γ-sampling



Multistep Generation with CTM: γ-sampling



Multistep Generation with CTM: γ-sampling



Multistep Generation with CTM: γ-sampling

$\gamma = 1$: Fully Stochastic Sampling

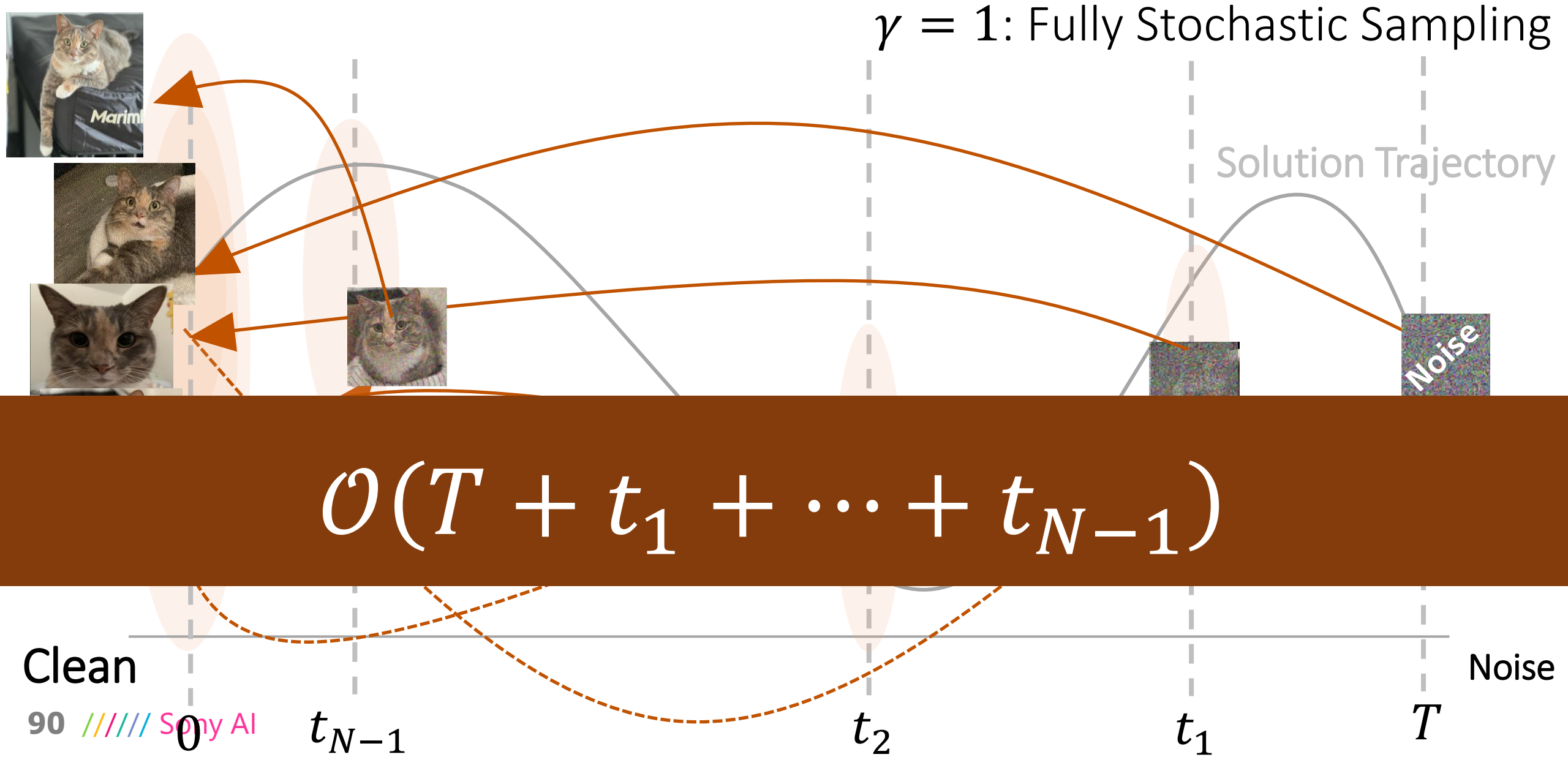
Solution Trajectory

Clean

Noise

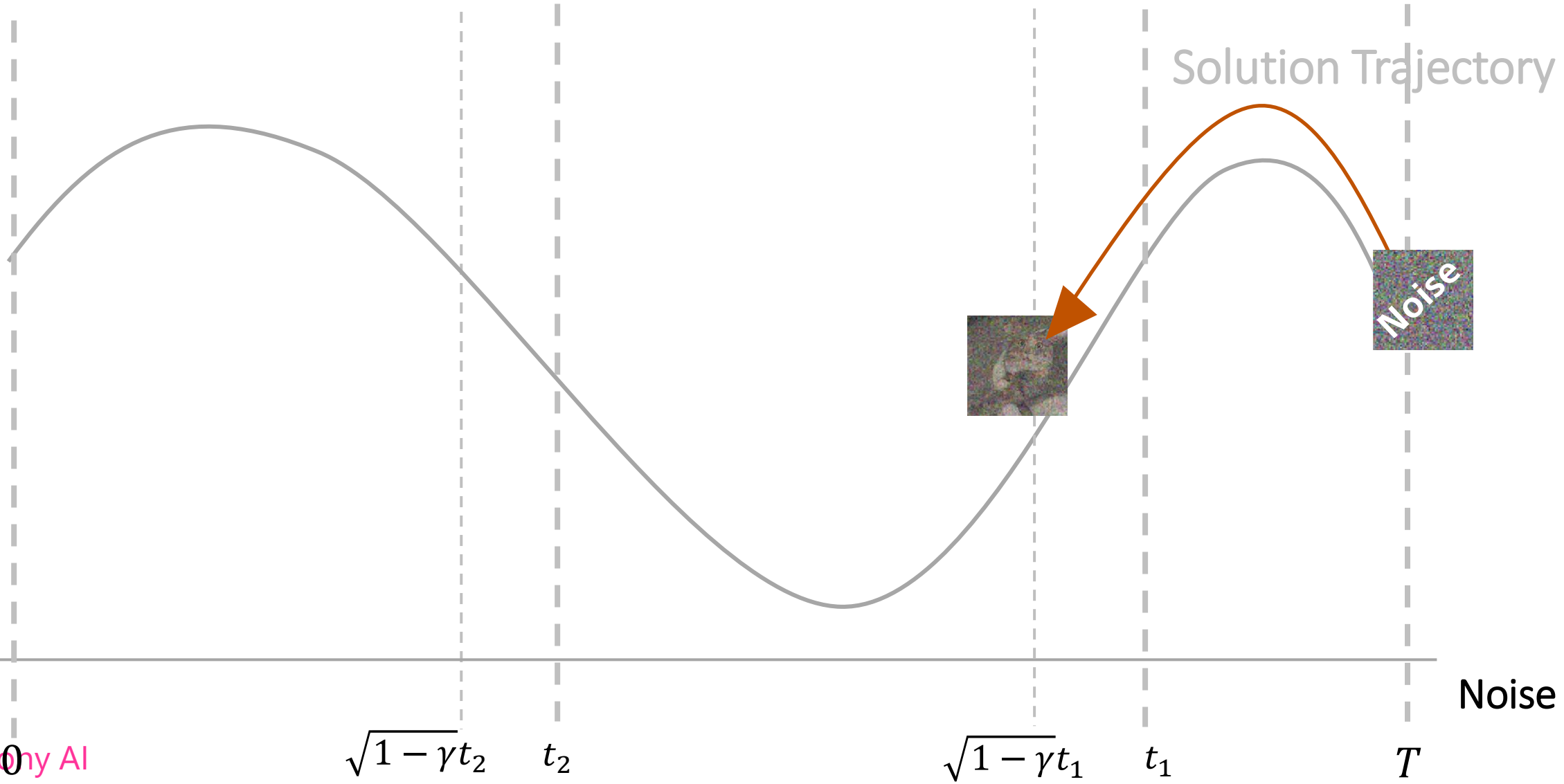
0 t_{N-1} t_2 t_1 T

Multistep Generation with CTM: γ-sampling



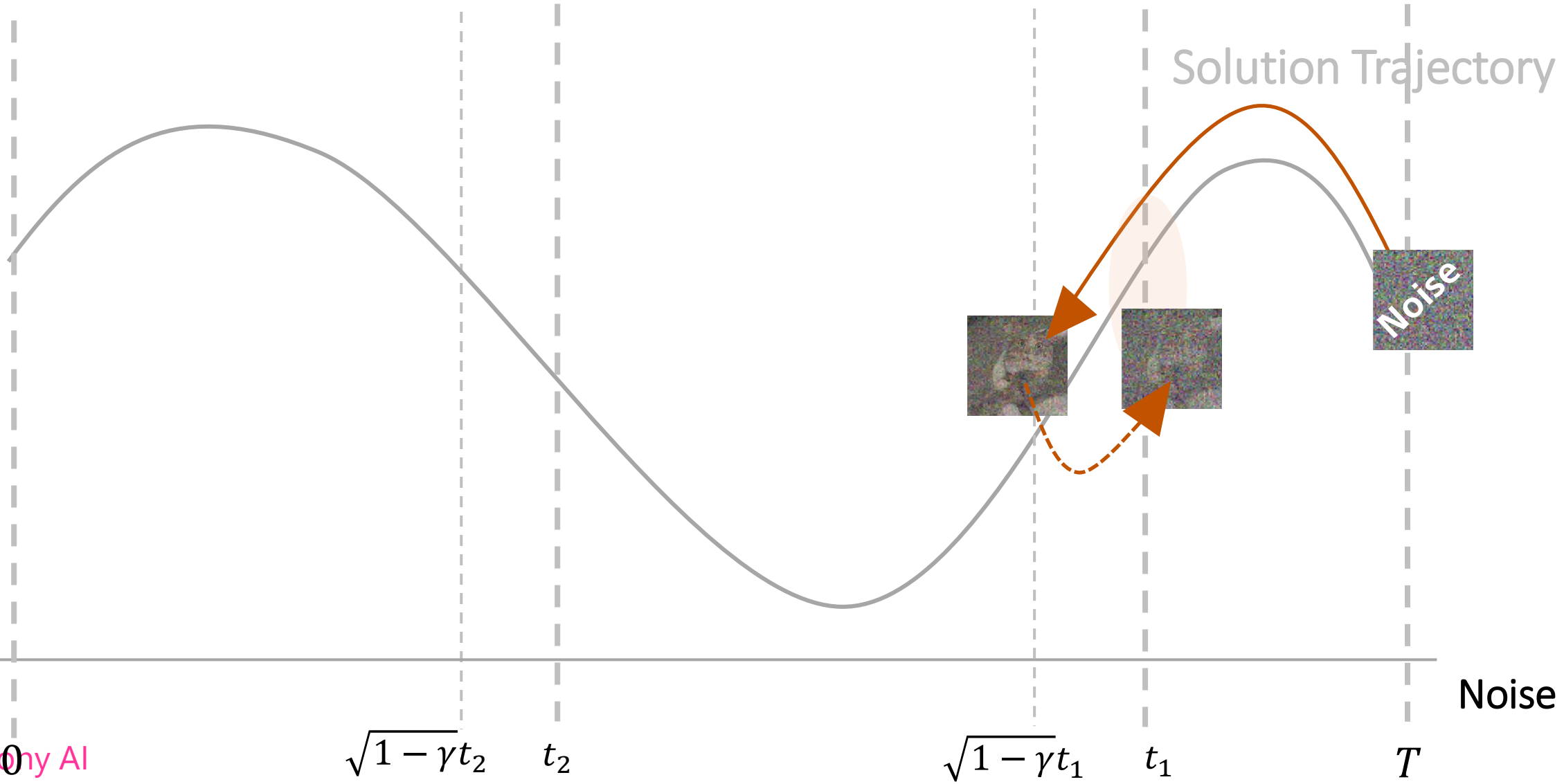
Multistep Generation with CTM: γ-sampling

$\gamma \in (0,1)$: Stochastic Sampling



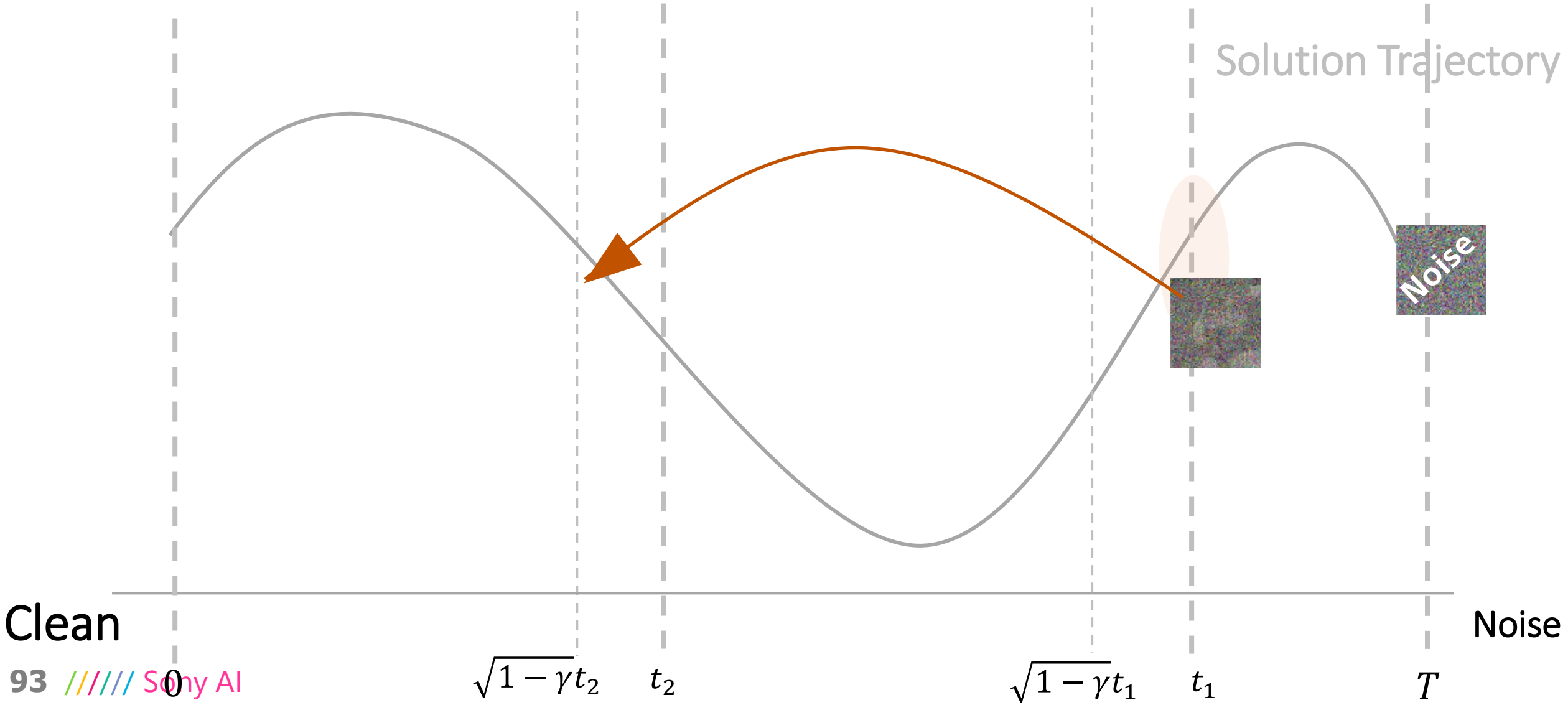
Multistep Generation with CTM: γ-sampling

$\gamma \in (0,1)$: Stochastic Sampling



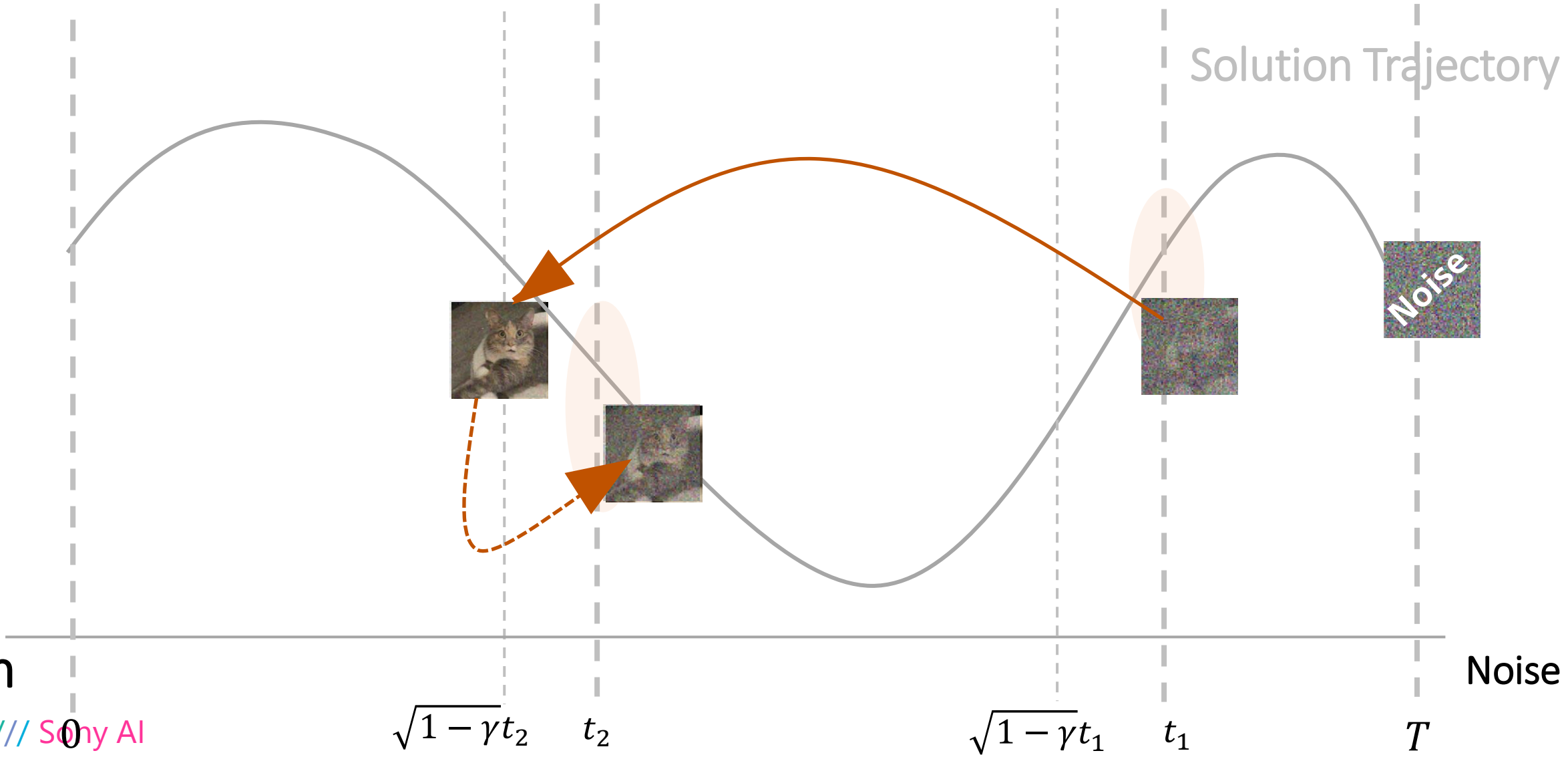
Multistep Generation with CTM: γ-sampling

$\gamma \in (0,1)$: Stochastic Sampling



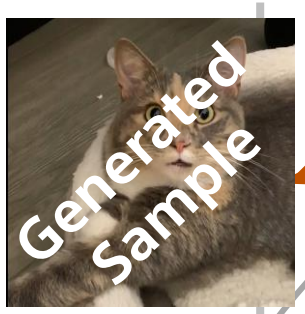
Multistep Generation with CTM: γ-sampling

$\gamma \in (0,1)$: Stochastic Sampling

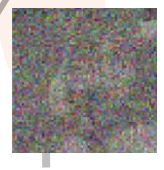


Multistep Generation with CTM: γ-sampling

$\gamma \in (0,1)$: Stochastic Sampling



Solution Trajectory



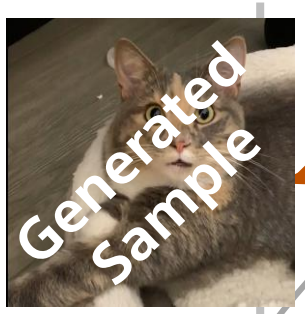
Clean

Noise

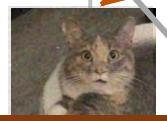
0 $\sqrt{1-\gamma}t_2$ t_2 $\sqrt{1-\gamma}t_1$ t_1 T

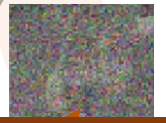
Multistep Generation with CTM: γ-sampling

$\gamma \in (0,1)$: Stochastic Sampling



Solution Trajectory





$$\mathcal{O}\left(\sum_{n=0}^{N-1}\sqrt{t_n - \sqrt{1-\gamma^2}\,t_{n+1}}\right)$$

0 $\sqrt{1-\gamma}t_2$ t_2 $\sqrt{1-\gamma}t_1$ t_1 T

CTM is a General Recipe to Many Modalities

SoundCTM: Uniting Score-based and Consistency Models for Text-to-Sound Generation

Koichi Saito
Sony AI
NY, USA
koichi.saito@sony.com

Dongjun Kim
Stanford University
CA, USA

Takashi Shibuya
Sony AI
Tokyo, Japan

Chieh-Hsin Lai
Sony AI
Tokyo, Japan

Zhi Zhong
Sony Group Corporation
Tokyo, Japan

Yuhta Takida
Sony AI
Tokyo, Japan

Yuki Mitsufuji
Sony AI, Sony Group Corporation
NY, USA

DITTO-2: Distilled Diffusion Inference-Time T-Optimization for Music Generation

Zachary Novack[1] **Julian McAuley**[1] **Taylor Berg-Kirkpatrick**[1] **Nicholas Bryan**[2]

[1] University of California – San Diego
[2] Adobe Research

znovack@ucsd.edu, njb@ieee.org

SoundCTM's Generation with γ-sampling

SoundCTM: Uniting Score-based and Consistency Models for Text-to-Sound Generation

Koichi Saito[1], Dongjun Kim[2], Takashi Shibuya[1], Chieh-Hsin Lai[1],
Zhi Zhong[3], Yuhta Takida[1], Yuki Mitsufuji[1,3],
[1]Sony AI, [2]Stanford University, [3]Sony Group Corporation

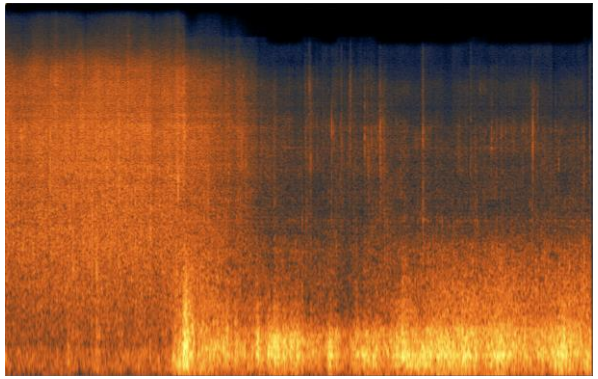
Paper PDF | arXiv | Code | Hugging Face

[Saito+ '24] SoundCTM

SoundCTM's Generation with γ-sampling

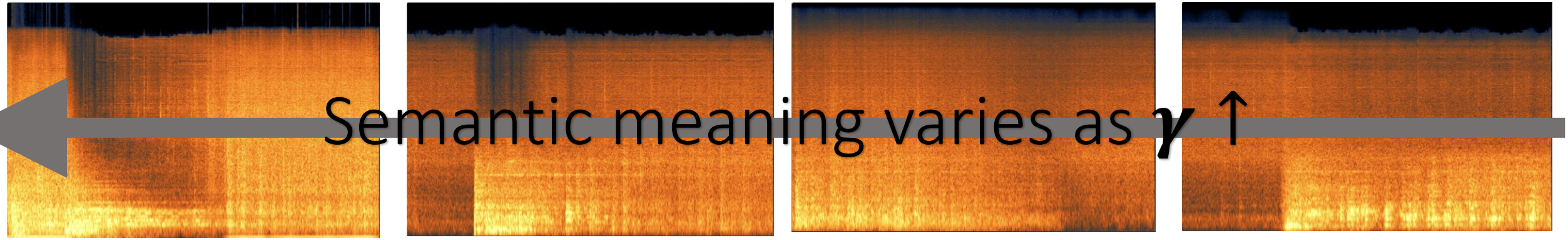
User specified Text prompt: *Thunderclaps, and hard rain falls and splashes on surfaces.*

One step



Semantically similar

16 steps

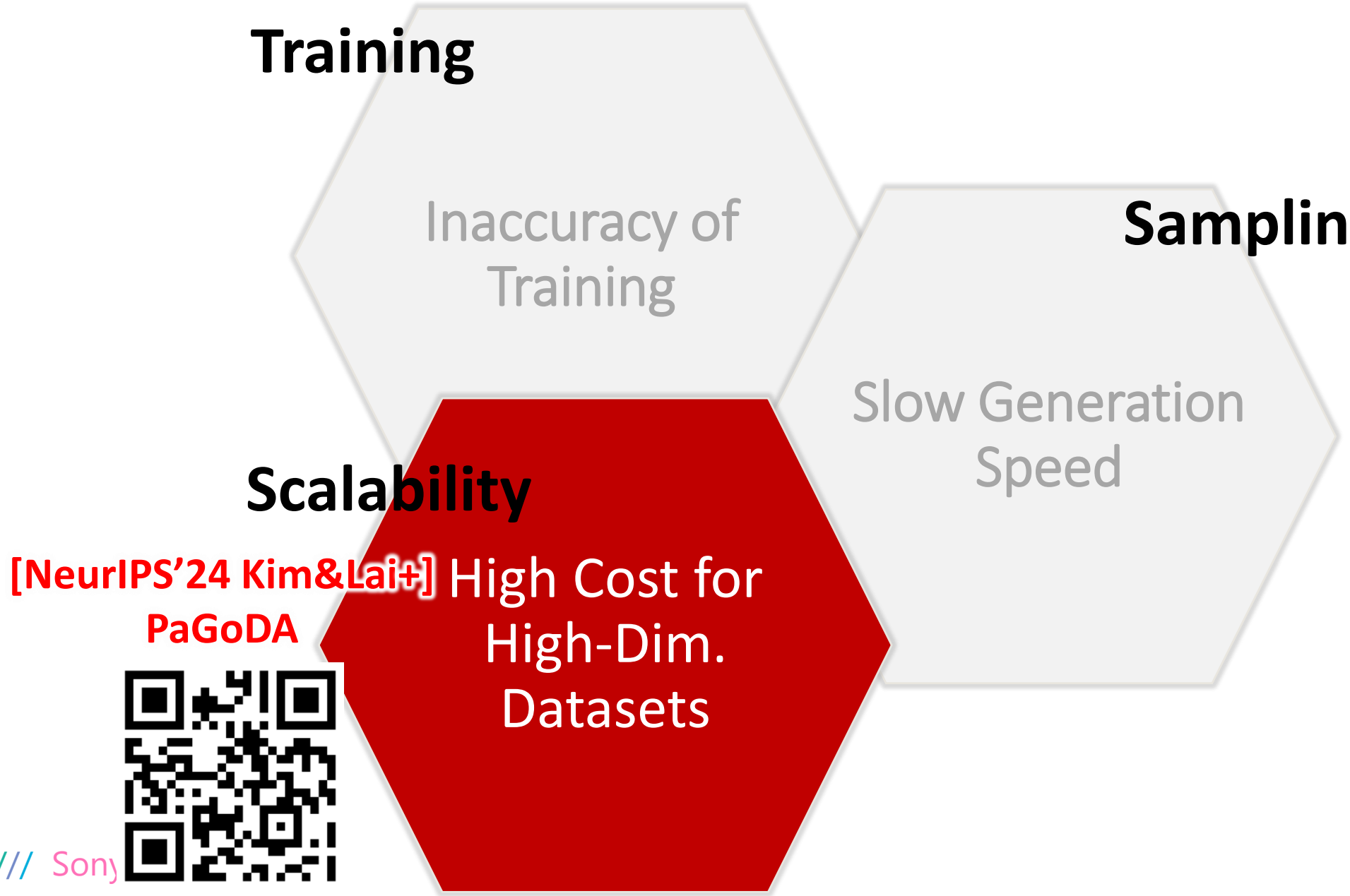


Semantic meaning varies as $\gamma \uparrow$

γ=1 γ=0.5 γ=0.25 γ=0

[Saito+ '24] SoundCTM

Making Diffusion Model More Powerful



[NeurIPS'24 Kim&Lai+] PaGoDA

Motivation of PaGoDA

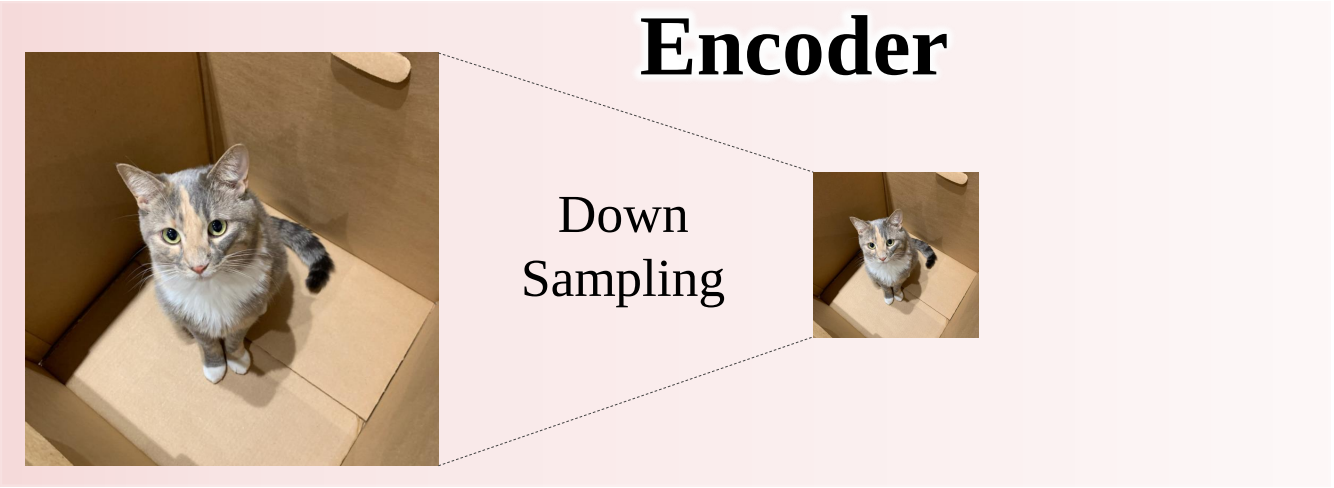
- DMs need to stay in the same dimension
- When image resolution ↑, training and sampling will be costly…
- Can we distill low-resolution DM for a **high-resolution 1-step** generation?

—— Training and Sampling Efficiency

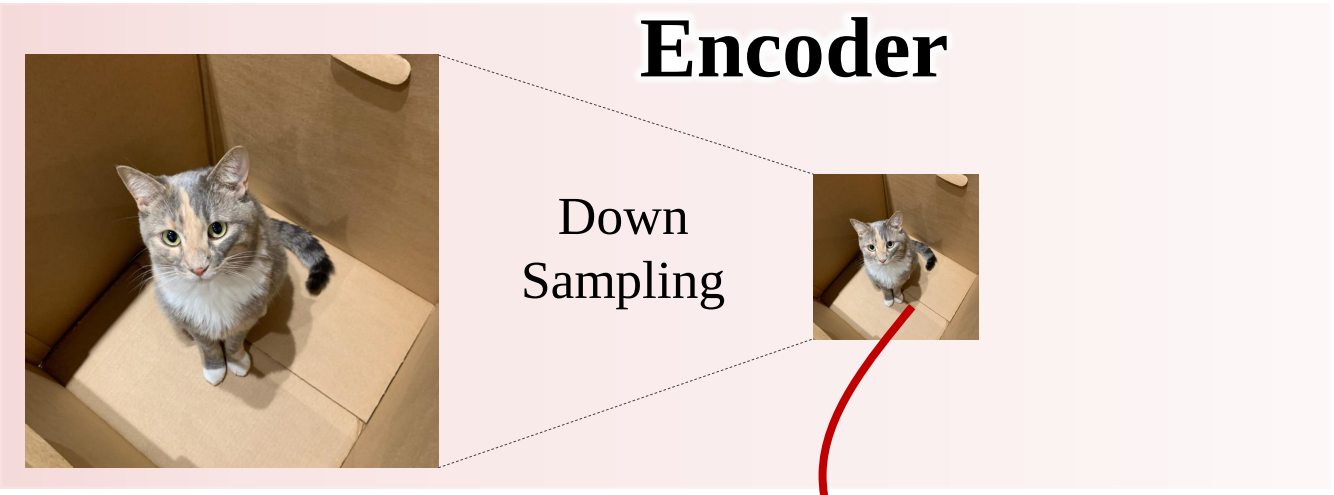
PaGoDA

Progressive
Growing **o**f
Diffusion
Autoencoder

Stage 1. **Pretrain** DM on low-dim data

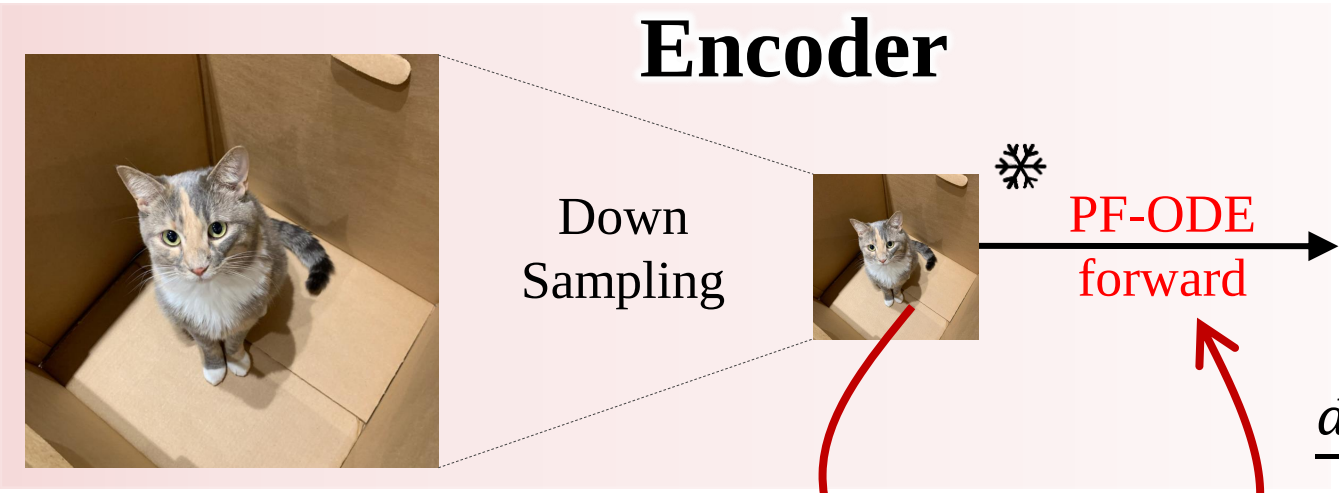


Stage 1. **Pretrain** DM on low-dim data



Train DM $\boldsymbol{s}_{\text{teacher}}(\boldsymbol{x}_t, t)$
on low dim.

Stage 1. **Pretrain** DM on low-dim data

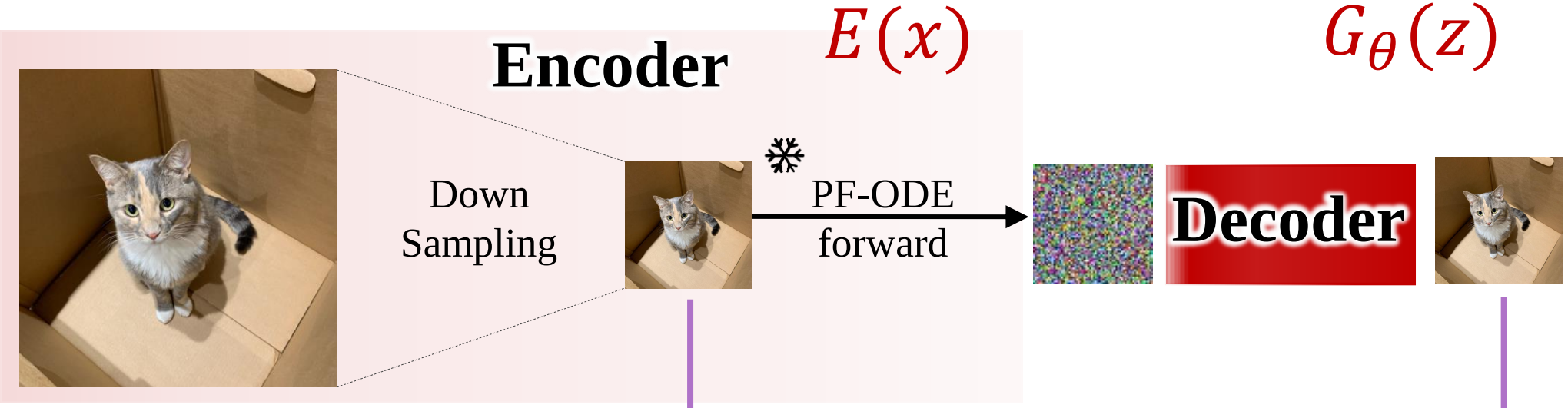


$$\frac{d\boldsymbol{x}_t}{dt} = -t\boldsymbol{s}_{\text{teacher}}(\boldsymbol{x}_t, t),$$

starting from $\boldsymbol{x}_{\text{low}}$

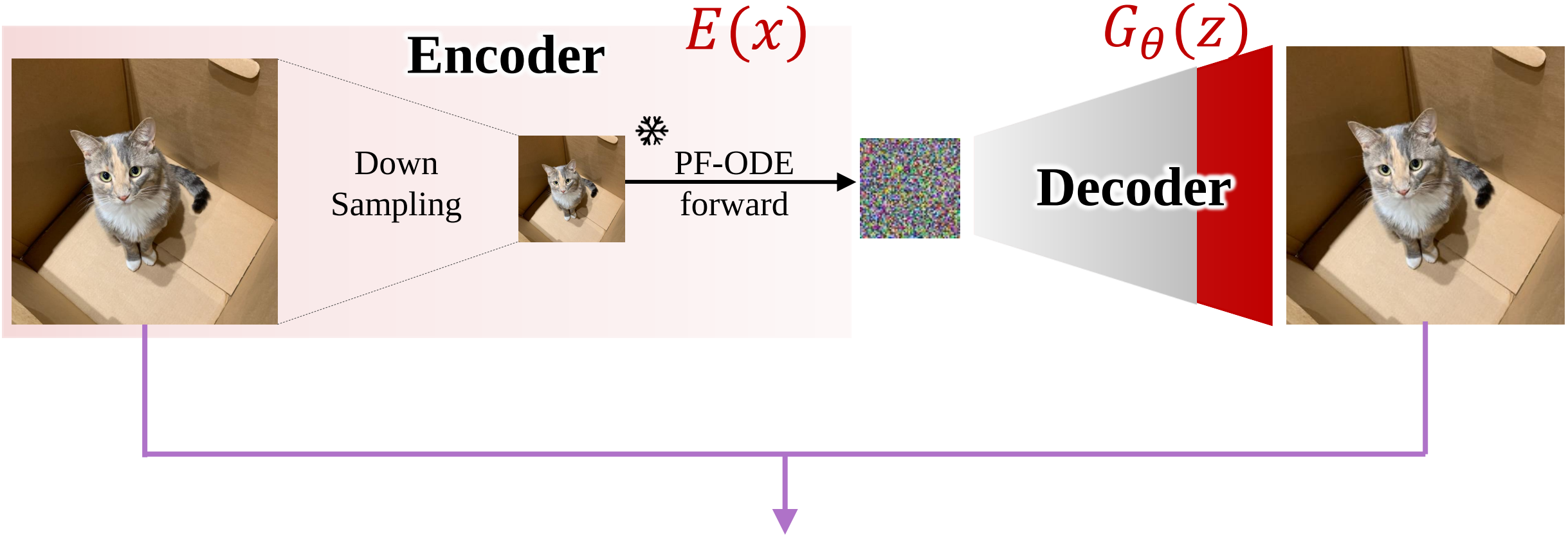
Train DM $\boldsymbol{s}_{\text{teacher}}(\boldsymbol{x}_t, t)$ on low dim.

Stage 2. **Distill** DM with ***Forward*** PF-ODE



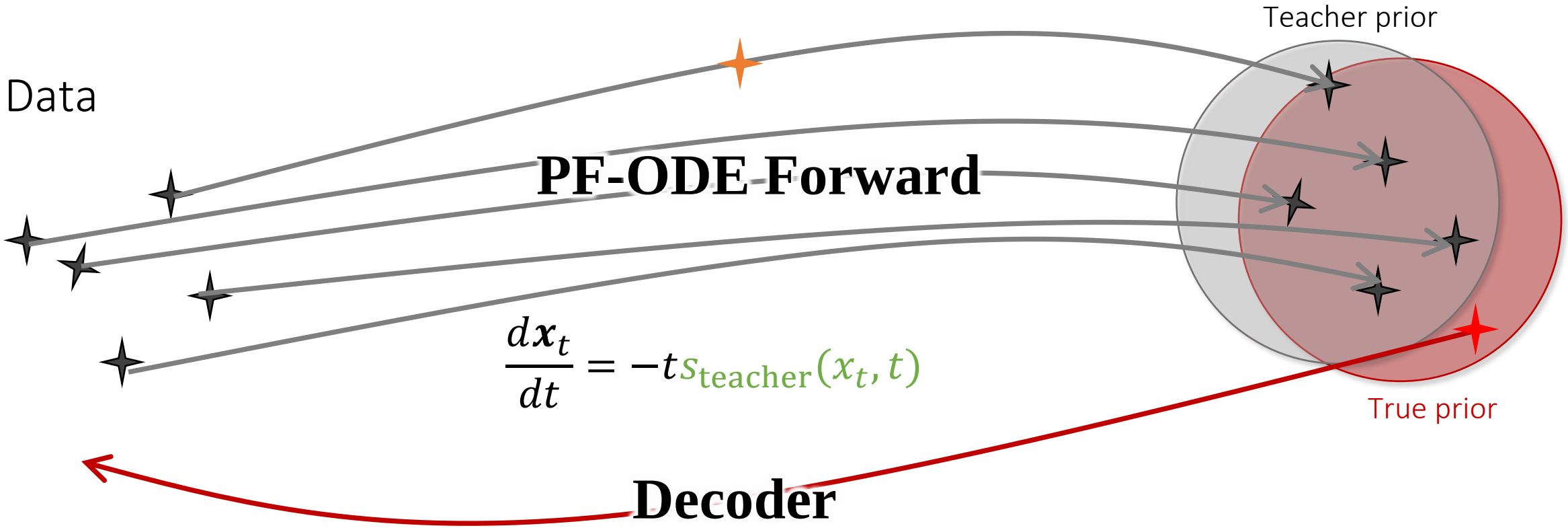
$$\mathcal{L}_{\text{rec}}(\theta) := \mathbb{E}_{p_{\text{data}}(x_{\text{low}})}\left[\left\|x_{\text{low}} - G_\theta\left(E(x_{\text{low}})\right)\right\|_2^2\right]$$

Stage 3. Progressively Grows Generator's Output Dim.



$$\mathcal{L}_{\text{rec}}(\theta) \coloneqq \mathbb{E}_{p_{\text{data}}(x_{\text{high}})}\left[\left\|x_{\text{high}} - G_\theta\left(E\left(x_{\text{high}}\right)\right)\right\|_2^2\right]$$

Issues of Solely Learning with Rec.

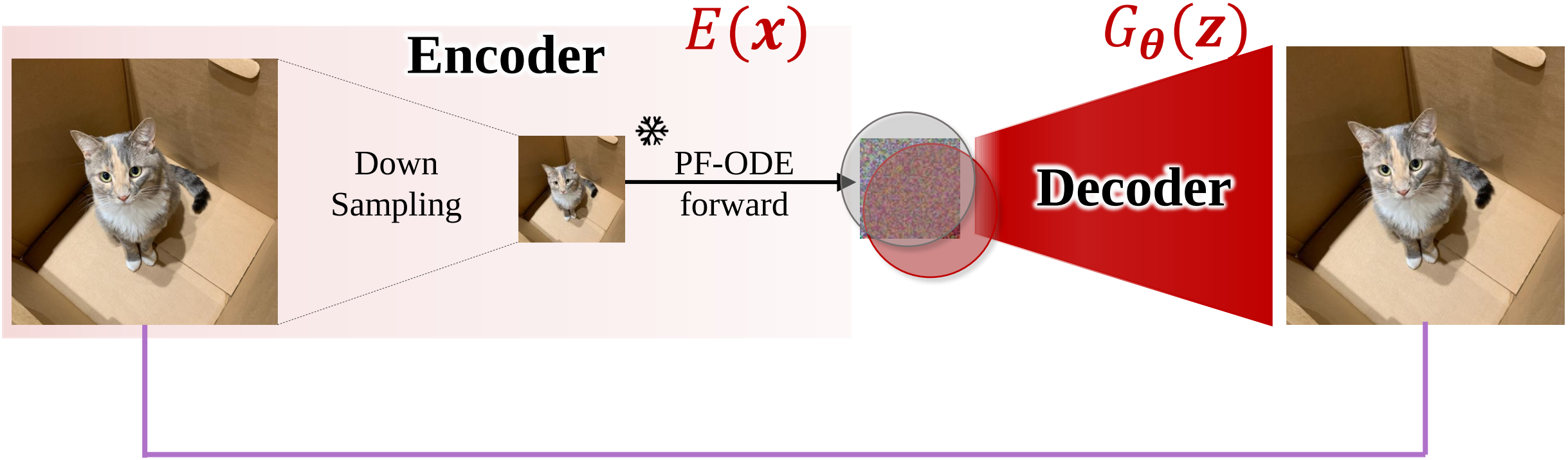


Theorem 1 (Informal) [Density can be improved with enhanced teacher]

$$W_q\left(p_{\text{PaGoDA\_noGAN},\theta}, p_{\text{data}}\right) \lesssim \mathcal{L}_{\text{rec}}(\theta) + \epsilon_{\text{DSM}}, q = 1,2$$

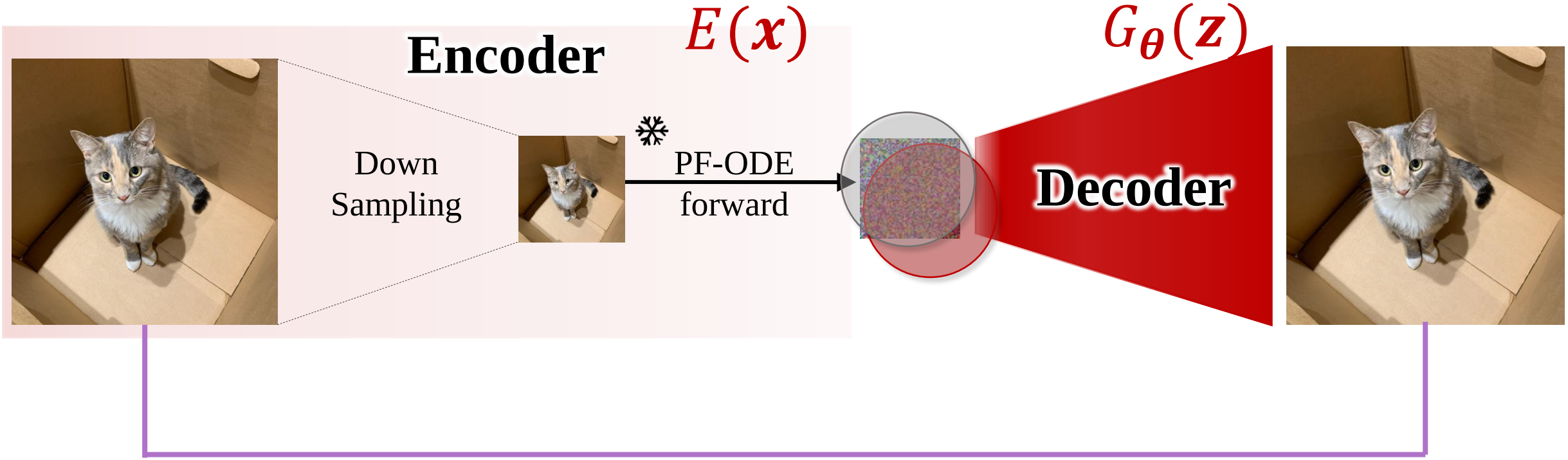
$$\mathcal{L}_{\text{rec}}(\theta) \coloneqq \mathbb{E}_{p_{\text{data}}(x)}\left[\left\|x - G_\theta\left(E(x)\right)\right\|_2^2\right]$$

Overall PaGoDA's Objective



$$\mathcal{L}_{\text{rec}}(G_\theta) := \mathbb{E}_{p_{\text{data}}(x_{\text{high}})}\left[\left\| x_{\text{high}} - G_\theta\left(E\left(x_{\text{high}}\right)\right)\right\|_2^2\right]$$

Overall PaGoDA's Objective

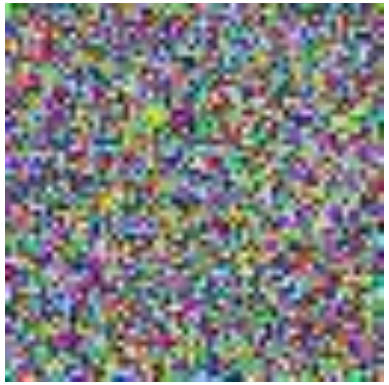


$$\mathcal{L}_{\text{rec}}(G_\theta) := \mathbb{E}_{p_{\text{data}}(x_{\text{high}})}\left[\left\|x_{\text{high}} - G_\theta\left(E(x_{\text{high}})\right)\right\|_2^2\right]$$

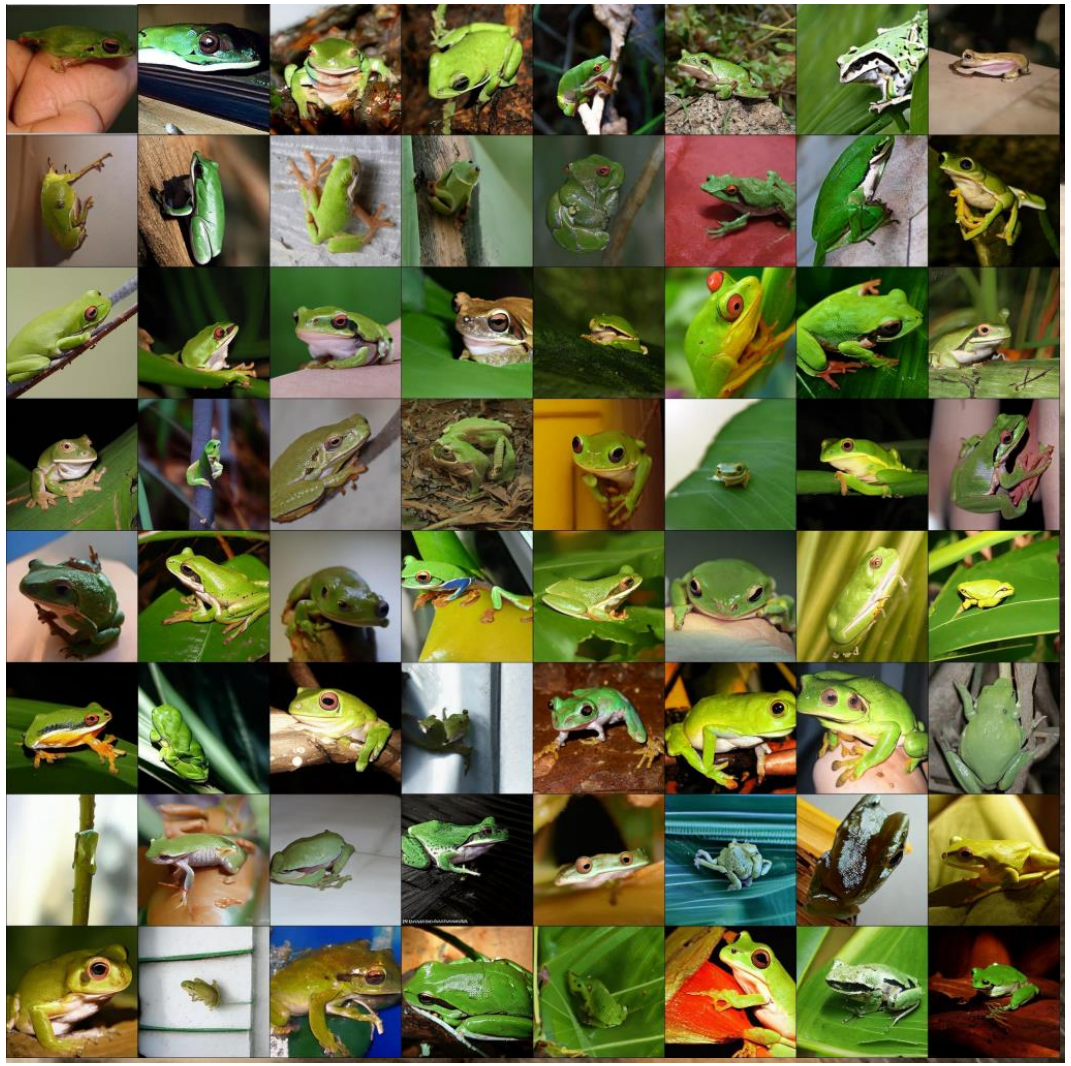
$$\mathcal{L}_{\text{GAN}}(G_\theta, D_\psi) := \mathbb{E}_{p_{\text{data}}(x_{\text{high}})}[\log D_\psi(x)] + \mathbb{E}_{p_{\text{prior}}(z_{\text{low}})}\left[\log\left(1 - D_\psi(G_\theta(z_{\text{low}}))\right)\right]$$

$$\mathcal{L}_{\text{PaGoDA}}(G_\theta, D_\psi) := \mathcal{L}_{\text{rec}}(G_\theta) + \lambda\mathcal{L}_{\text{GAN}}(G_\theta, D_\psi)$$

SOTA 1-step “Res.-Agnostic” Generation



64 ⟶ 64, 128, 256, 512



- ✓ Don’t need pre-training DM on high-dim.
- ✓ Don’t need expensive inference

111

SOTA 1-step “Res.-Agnostic” Generation

Text-to-Image generation

64 → 64, 128, 256, 512

- ✓ Don’t need pre-training DM on high-dim.
- ✓ Don’t need expensive inference

112

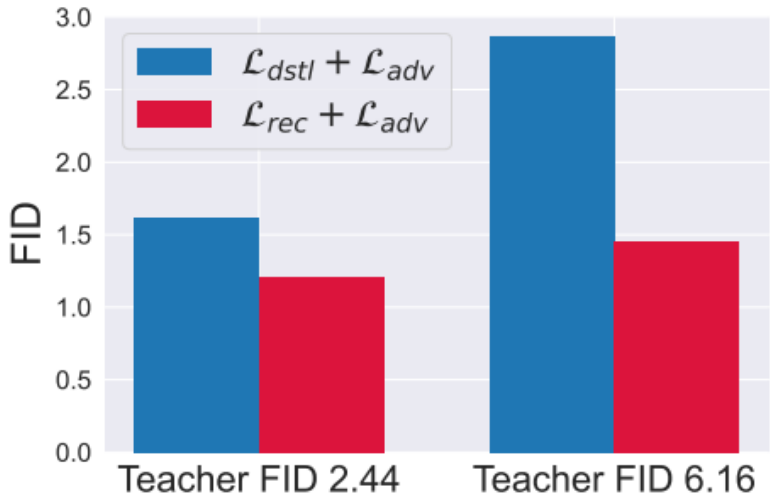
Insight I. PaGoDA can learn the true p_{data}

$$\mathcal{L}_{\text{PaGoDA}}(G_{\boldsymbol{\theta}}, D_{\boldsymbol{\psi}}) := \mathcal{L}_{\text{rec}}(G_{\boldsymbol{\theta}}) + \lambda \mathcal{L}_{\text{GAN}}(G_{\boldsymbol{\theta}}, D_{\boldsymbol{\psi}})$$

Theorem 2 (Informal) [PaGoDA learns the true data] Assume an optimal discriminator $\boldsymbol{\psi}^*$ in the above loss. Then with some conditions, the following holds:

$$p_{\text{PaGoDA},\boldsymbol{\theta}^*} = p_{\text{data}},$$

where $p_{\text{PaGoDA},\boldsymbol{\theta}^*} := G_{\boldsymbol{\theta}^*} \# p_{\text{prior}}$. Not the case for conventional distillation-based method.



Insight II. PaGoDA is not GAN

$$\mathcal{L}_{\text{PaGoDA}}(G_{\boldsymbol{\theta}}, D_{\boldsymbol{\psi}}) := \mathcal{L}_{\text{rec}}(G_{\boldsymbol{\theta}}) + \lambda \mathcal{L}_{\text{GAN}}(G_{\boldsymbol{\theta}}, D_{\boldsymbol{\psi}})$$

(e) Upsampling of PaGoDA

(f) Upsampling of StyleGAN-XL

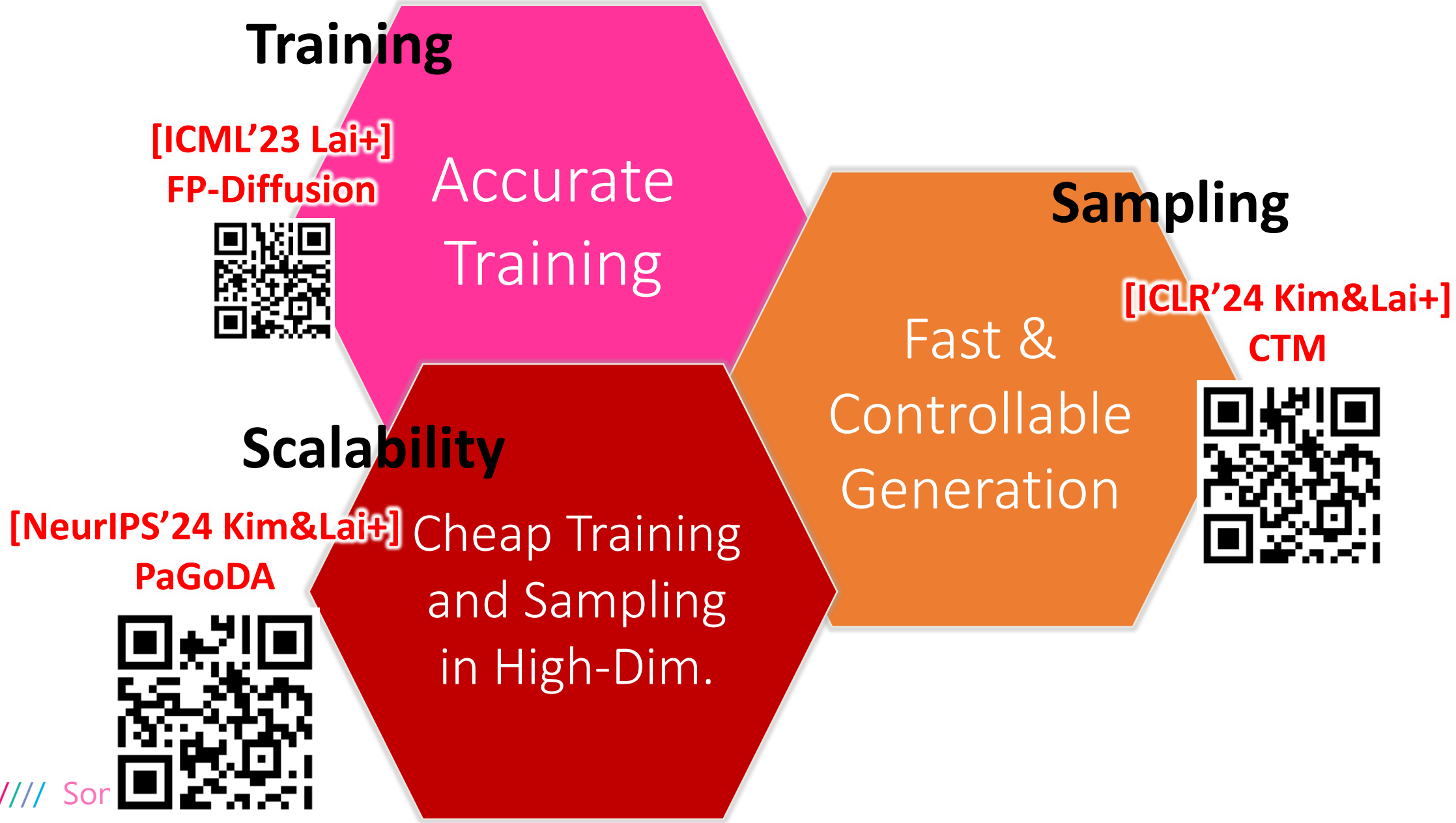
Theorem 3 (Informal) [PaGoDA is STABLE!] Let E be fixed ODE-based encoder. Assume at optimal generator $\boldsymbol{\theta}^*$ that

$$\boldsymbol{x} = G_{\boldsymbol{\theta}^*}\big(E(\boldsymbol{x})\big) \text{ on } \boldsymbol{x} \in \text{supp}(p_{\text{data}})$$

I. Then under some conditions, PaGoDA's alternative gradient descent of $(\boldsymbol{\theta}, \boldsymbol{\psi})$ is stable at $(\boldsymbol{\theta}^*, \boldsymbol{\psi}^*)$ in Lyapunov sense.

II. Potentially not the case for GAN.

Making Diffusion Models More Powerful

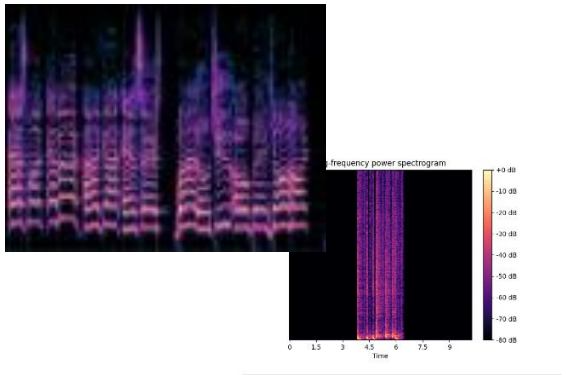


More than that…

Pre-trained Diffusion Models

Image Diffusion Model

Music/audio Diffusion Model

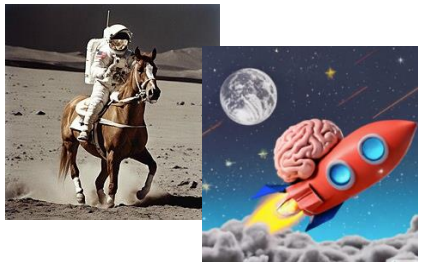


0:00 / 0:22

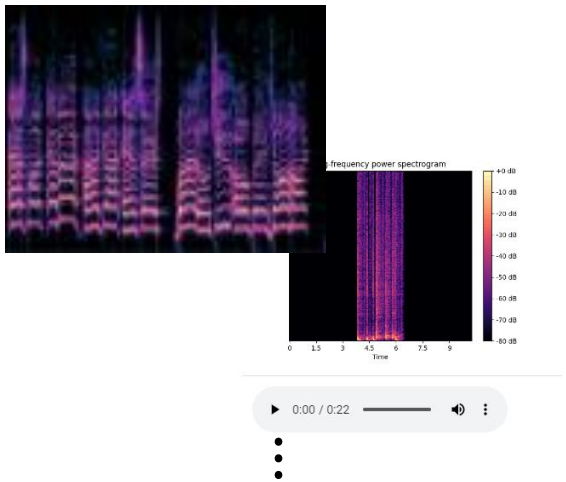
More than that…

Pre-trained Diffusion Models

Image Diffusion Model



Music/audio Diffusion Model



Pre-trained Diffusion for various applications

Without any extra training!!

ICML'23 GibbsDDRM

ICLR'24 MPGD

Various applications

Inverse Problems/Restoration

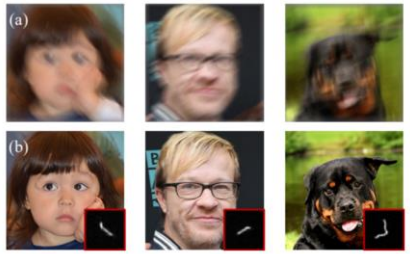
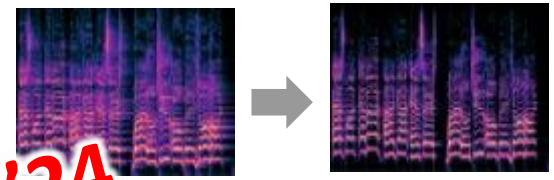


Image Restoration

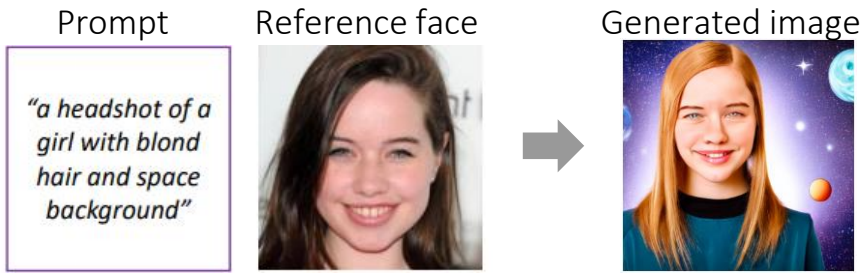
- GibbsDDRM [ICML'23]



Audio Restoration

- DiffDereverb [ICASSP'23]
- Diffiner in [Interspeech'23]
- VRDMG in [ICASSP'24]

Guided (Controllable) Generation



Missing Collaborator…



MawMaw (貓貓)

…literally means Cat Cat

THANK YOU!


Chieh-Hsin.Lai@sony.com


@JCJesseLai