Score-Based Generative Modeling through Stochastic Differential Equations

Yang Song[2], Jascha Sohl-Dickstein[1], Diederik P. Kingma[1], Abhishek Kumar[1], Stefano Ermon[2], and Ben Poole[1]
[1]Google Brain, [2]Stanford University

ABSTRACT

- We perturb data to noise with a fixed stochastic differential equation (SDE), and learn to reverse it for sample generation. The reverse SDE can be obtained by estimating the time-dependent gradient field (aka. score function) of the perturbed data distribution.
- We achieve outstanding sample quality: state-of-the-art FID and Inception scores on CIFAR-10, high fidelity generation of 1024x1024 images.
- The SDE framework allows exact likelihood computation. We obtain the state-of-the-art likelihood on uniformly dequantized CIFAR-10 images, without maximum likelihood training.
- We can perform controllable generation without re-training models, and demonstrate applications in class-conditional generation, image inpainting and colorization.

Code:

Generative Modeling with SDEs

Stochastic differential equation (SDE):

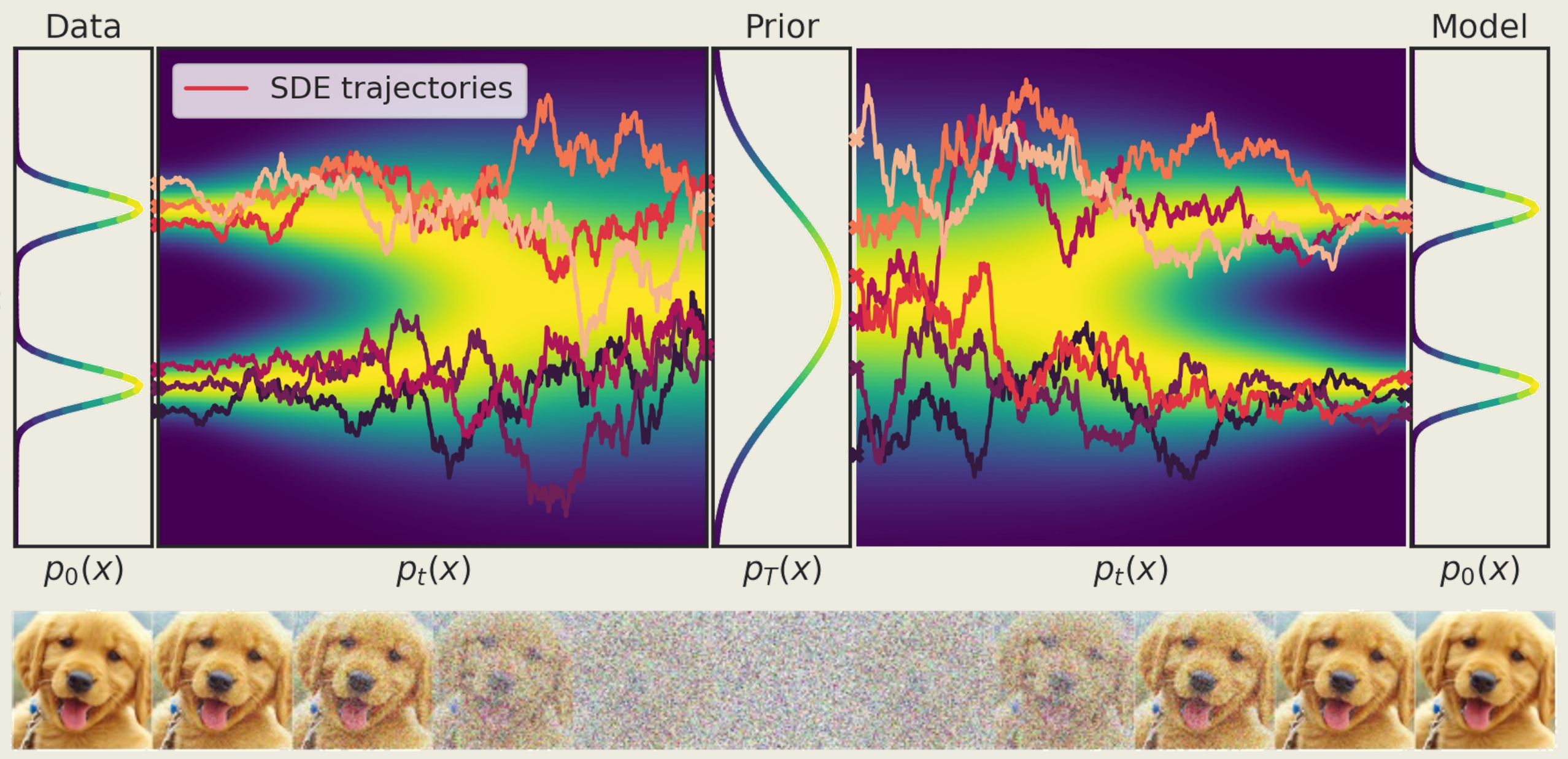
$$\mathrm{d}\mathbf{x} = \boxed{\boldsymbol{f}(\mathbf{x},t)\,\mathrm{d}t} + \boxed{g(t)\,\mathrm{d}\mathbf{w}}$$

Brownian motion

Deterministic drift

Stochastic diffusion

Perturbing data with a fixed SDE, and reverse it for generative modeling



The reverse-time SDE:

$$\mathrm{d}\mathbf{x} = [\boldsymbol{f}(\mathbf{x},t) - g^2(t)\boxed{\nabla_\mathbf{x}\log p_t(\mathbf{x})}]\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}$$

Score function of $p_t(\mathbf{x})$

- Must be solved in the reverse time direction
- Requires estimating score functions at all time steps.

Learning to reverse the SDE:

- **Time-dependent score-based model**

$$\boldsymbol{s}_{\boldsymbol{\theta}}(\cdot,t):\mathbb{R}^d\to\mathbb{R}^d$$

- **Goal:**

$$\boldsymbol{s}_{\boldsymbol{\theta}^*}(\mathbf{x},t)\approx\nabla_\mathbf{x}\log p_t(\mathbf{x})$$

- **Training objective:**

$$\min_{\boldsymbol{\theta}}\mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\mathbf{x}\sim p_t(\mathbf{x})}[\lambda(t)\,\|\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x},t)-\nabla_\mathbf{x}\log p_t(\mathbf{x})\|_2^2]$$

?

- **Score matching (Hyvarinen 2005)**
  - Denoising score matching (Vincent 2010)
  - Sliced score matching (Song et al., 2019)

- **Estimated reverse-time SDE**

$$\mathrm{d}\mathbf{x} = [\boldsymbol{f}(\mathbf{x},t) - g^2(t)\boldsymbol{s}_{\boldsymbol{\theta}^*}(\mathbf{x},t)]\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}$$

Solving Reverse SDEs for Sampling

Numerical SDE solvers:

- **Example: Euler-Maruyama method**

Initialize $t = T, \quad \mathbf{x}\sim p_T(\mathbf{x})$

Repeat until $t=0$

$$\begin{cases}\Delta\mathbf{x}\leftarrow[\boldsymbol{f}(\mathbf{x},t)-g^2(t)\boldsymbol{s}_{\boldsymbol{\theta}^*}(\mathbf{x},t)]\Delta t+g(t)\mathbf{z}\\ \mathbf{x}\leftarrow\mathbf{x}+\Delta\mathbf{x}\\ t\leftarrow t+\Delta t\end{cases}$$

$$\mathbf{z}\sim\mathcal{N}(\mathbf{0},|\Delta t|\boldsymbol{I})$$

- **Example: Reverse diffusion method** (see paper)

Predictor-Corrector methods:

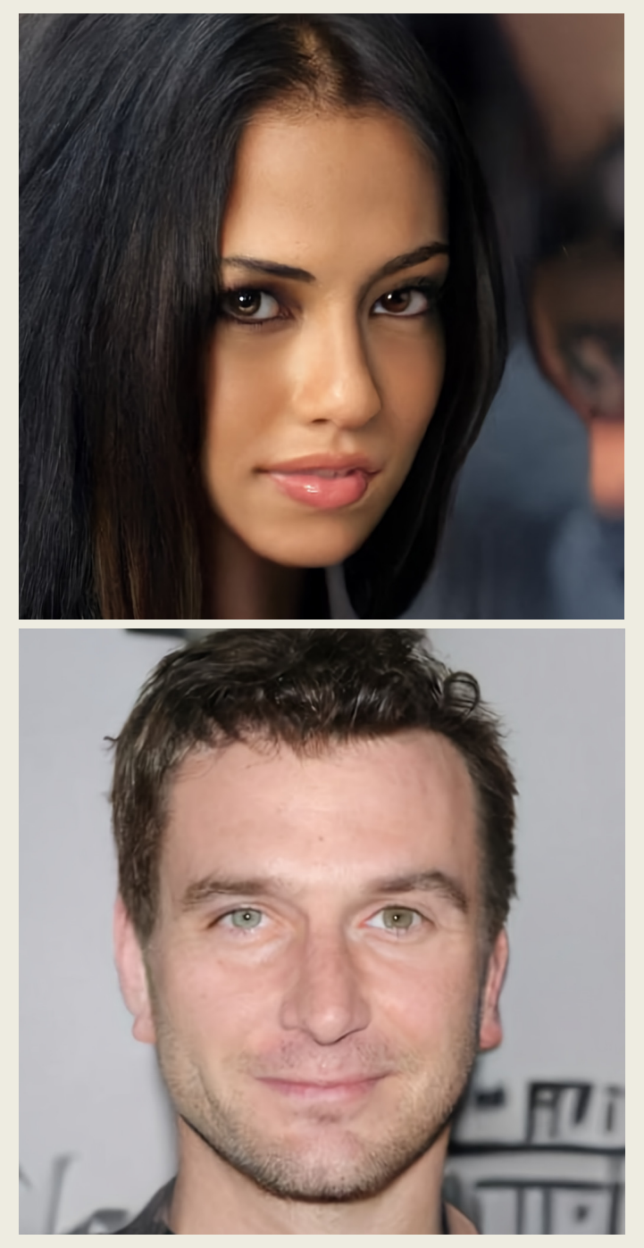
- Improves numerical SDE solvers with MCMC, at the cost of more computation and more hyperparameters.

Experimental results:

CIFAR-10 sample quality

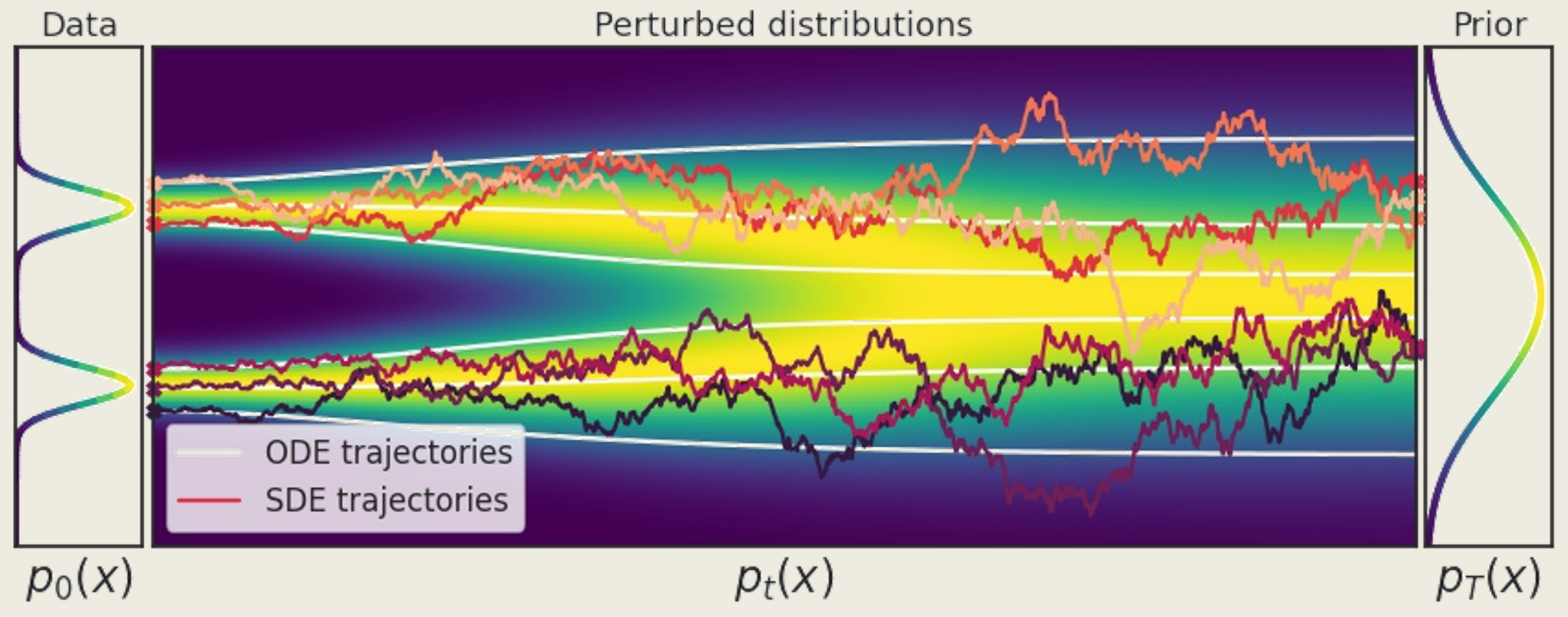
Model	FID↓	IS↑
Conditional		
BigGAN (Brock et al., 2018)	14.73	9.22
StyleGAN2-ADA (Karras et al., 2020a)	**2.42**	**10.14**
Unconditional		
StyleGAN2-ADA (Karras et al., 2020a)	2.92	9.83
NCSN (Song & Ermon, 2019)	25.32	8.87 ± .12
NCSNv2 (Song & Ermon, 2020)	10.87	8.40 ± .07
DDPM (Ho et al., 2020)	3.17	9.46 ± .11
DDPM++	2.78	9.64
DDPM++ cont. (VP)	2.55	9.58
DDPM++ cont. (sub-VP)	2.61	9.56
DDPM++ cont. (deep, VP)	2.41	9.68
DDPM++ cont. (deep, sub-VP)	2.41	9.57
NCSN++	2.45	9.73
NCSN++ cont. (VE)	2.38	9.83
NCSN++ cont. (deep, VE)	**2.20**	**9.89**

CelebA-HQ 1024px samples



Probability Flow ODEs

Turning the SDE into an ODE with the same $p_t(\mathbf{x})$



- **Probability flow ODE:**

$$\mathrm{d}\mathbf{x} = \left[\boldsymbol{f}(\mathbf{x},t) - \frac{1}{2}g^2(t)\boxed{\nabla_\mathbf{x}\log p_t(\mathbf{x})}\right]\mathrm{d}t$$

Score function of $p_t(\mathbf{x}) \approx \boldsymbol{s}_{\boldsymbol{\theta}^*}(\mathbf{x},t)$

Can sample from the same distribution by solving the ODE instead of the SDE.

Exact likelihood computation:

Model	NLL Test ↓	FID ↓
RealNVP (Dinh et al., 2016)	3.49	-
iResNet (Behrmann et al., 2019)	3.45	-
Glow (Kingma & Dhariwal, 2018)	3.35	-
MintNet (Song et al., 2019b)	3.32	-
Residual Flow (Chen et al., 2019)	3.28	46.37
FFJORD (Grathwohl et al., 2018)	3.40	-
Flow++ (Ho et al., 2019)	3.29	-
DDPM (L) (Ho et al., 2020)	$\leqslant 3.70^*$	13.51
DDPM (L_{simple}) (Ho et al., 2020)	$\leqslant 3.75^*$	3.17
DDPM	3.28	3.37
DDPM cont. (VP)	3.21	3.69
DDPM cont. (sub-VP)	3.05	3.56
DDPM++ cont. (VP)	3.16	3.93
DDPM++ cont. (sub-VP)	3.02	3.16
DDPM++ cont. (deep, VP)	3.13	3.08
DDPM++ cont. (deep, sub-VP)	**2.99**	**2.92**

$\log p_T(\mathbf{z})$

Instantaneous change-of-variable formula (Chen et al. 2018)

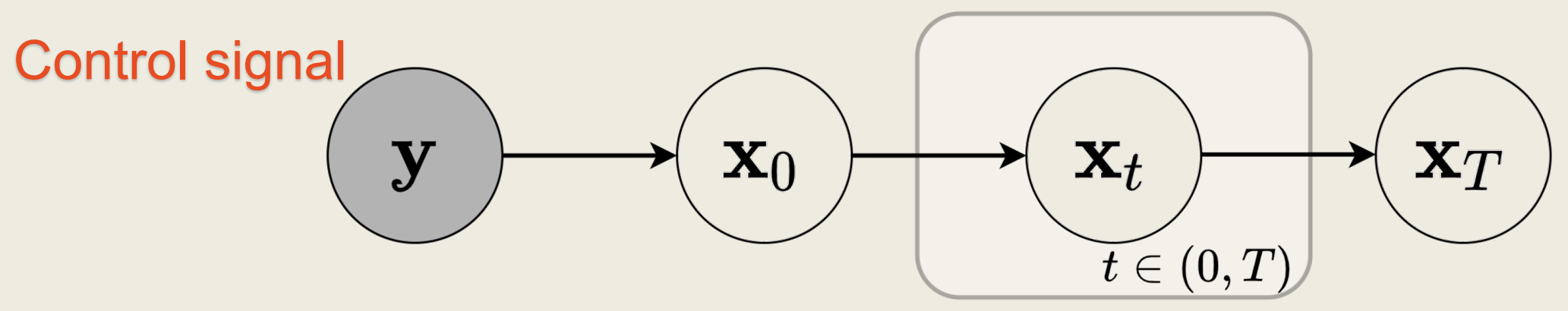
$\int$

$\log p_0(\mathbf{x})$

Controllable Generation

We can perform conditional generation with an unconditional score-based model. No need of re-training.

Control signal



Reverse for controllable generation

$$\nabla_\mathbf{x}\log p_t(\mathbf{x}\mid\mathbf{y}) = \boxed{\nabla_\mathbf{x}\log p_t(\mathbf{x})} + \boxed{\nabla_\mathbf{x}\log p_t(\mathbf{y}\mid\mathbf{x})}$$

unconditional score, Trained w/o y

specified with domain knowledge

Image inpainting and colorization results: