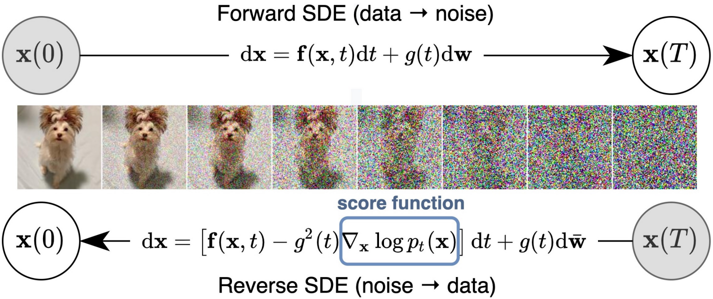
Diffusion Generative Modeling: Making Pictures from Noise with Math

Vasily Ilin

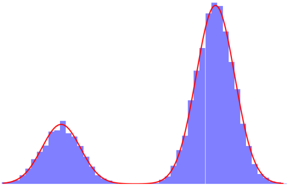


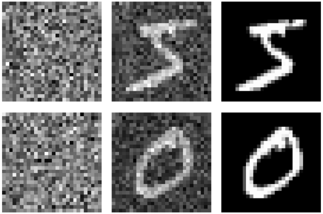
Two Types of Sampling

Two types of sampling:

- model-no-data – classical sampling
- data-no-model – generative modeling.

For example, given millions of pictures on the Internet, how to generate more pictures?





Langevin Dynamics

"Creating noise from data is easy; creating data from noise is generative modeling" (Song et al, 2020)

Langevin Dynamics

"Creating noise from data is easy; creating data from noise is generative modeling" (Song et al, 2020) As $t \to \infty$, the distribution of X_t converges to π:

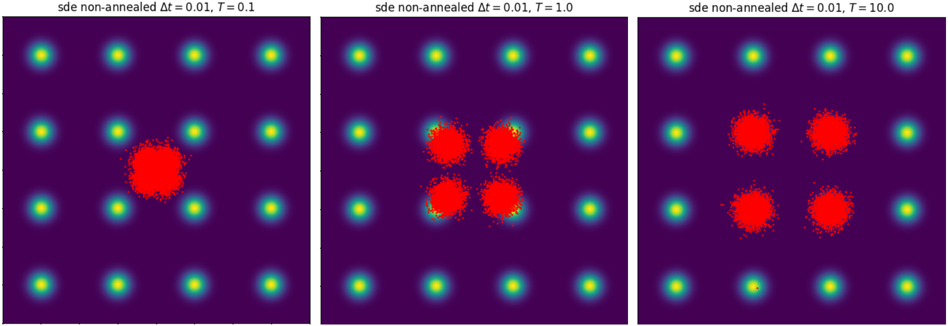
$$dX_t = \nabla \log \pi(X_t) dt + \sqrt{2} dB_t, \quad B_t := \text{Brownian motion}$$

Langevin Dynamics

"Creating noise from data is easy; creating data from noise is generative modeling" (Song et al, 2020) As $t \to \infty$, the distribution of X_t converges to π:

$$dX_t = \nabla \log \pi(X_t)dt + \sqrt{2}dB_t, \quad B_t := \text{Brownian motion}$$

But Langevin dynamics gets stuck when π is multimodal! The mixing time is exponential in distance between modes.



Images are Multimodal

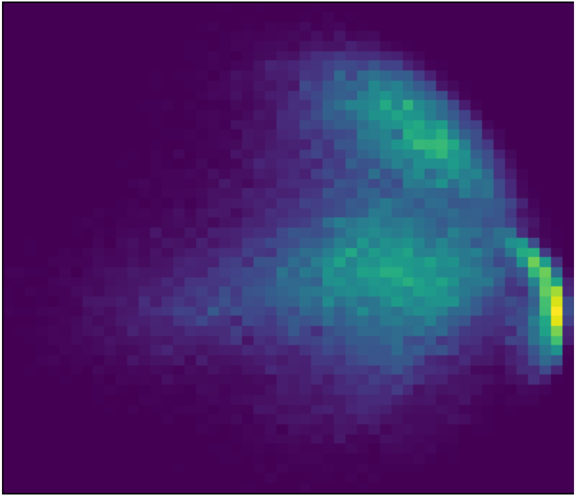


Figure: ICA on MNIST dataset

Reversed SDE

Idea: Reverse the Ornstein-Uhlenbeck process

$$dX_t = -X_t dt + \sqrt{2} dB_t, \quad 0 \leq t \leq T$$

Reversed SDE

Idea: Reverse the Ornstein-Uhlenbeck process

$$dX_t = -X_t dt + \sqrt{2} dB_t, \quad 0 \leq t \leq T$$

The reverse SDE is

$$\begin{aligned} dX_t^{\leftarrow} &= X^{\leftarrow} dt + 2\nabla \log f_{T-t}(X_t^{\leftarrow}) dt + \sqrt{2} dB_t, \\ f_t &:= \mathrm{law}(X_t), \quad X_t^{\leftarrow} := X_{T-t}. \end{aligned}$$

Proof.

On the board... □

Reversed SDE

Idea: Reverse the Ornstein-Uhlenbeck process

$$dX_t = -X_t dt + \sqrt{2} dB_t, \quad 0 \leq t \leq T$$

The reverse SDE is

$$dX_t^{\leftarrow} = X^{\leftarrow} dt + 2\nabla \log f_{T-t}(X_t^{\leftarrow}) dt + \sqrt{2} dB_t,$$
$$f_t := \mathrm{law}(X_t), \quad X_t^{\leftarrow} := X_{T-t}.$$

Proof.
On the board... □

How to estimate the *score* $\nabla \log f_t$?

Score Matching

Approximate $\nabla \log f_t$ with a Neural Network s_t by minimizing the least-squares error.

$$
\begin{aligned}
L(s, f) &= \mathbb{E}_f \|s - \nabla \log f\|^2 \\
&= \mathbb{E}_f \|s\|^2 - 2s \cdot \nabla \log f + const(s) \\
&= \mathbb{E}_f \|s\|^2 + 2\nabla \cdot s + const(s) \\
&= \frac{1}{n} \sum_{i=1}^{n} \|s_t(X_t^i)\|^2 + 2\nabla \cdot s_t(X_t^i) + const(s),
\end{aligned}
$$

where X_t comes from the OU process:

$$
dX_t = -X_t dt + \sqrt{2} dB_t, \quad X_0 \sim \pi
$$

Algorithm

Step 1: Simulate the OU process

$$dX_t = -X_t dt + \sqrt{2} dB_t$$

starting from $X_0^1, \ldots, X_0^n \sim \pi$ for $0 \leq t \leq T$.
Step 2: Train the NN by minimizing

$$\frac{1}{n} \sum_{i=1}^{n} \|s_t(X_t^i)\|^2 + 2\nabla \cdot s_t(X_t^i)$$

Step 3: Simulate the reverse process

$$dX_t^{\leftarrow} = X^{\leftarrow} + 2s_{T-t}(X_t^{\leftarrow})dt + \sqrt{2} dB_t$$

for $0 \leq t \leq T$.
Output: $X_T^{\leftarrow}$.

Fast Convergence

Theorem (Chen et al '23)

Without convexity assumptions on π, convergence is fast.

$$TV(law(X_T^{\leftarrow}), \pi) \lesssim \underbrace{\sqrt{KL(law(X_T) \| \mathcal{N}(0, I))}}_{\text{OU process convergence}} + \underbrace{(\sqrt{dh} + mh)\sqrt{T}}_{\text{time discretization}} + \underbrace{\sqrt{L(s, f)}\sqrt{T}}_{\text{score estimation}}$$

Proof.

By the data-processing inequality and **a lot of** *stochastic calculus.* □

Generating Digits

I trained a NN to generate handwritten digits.

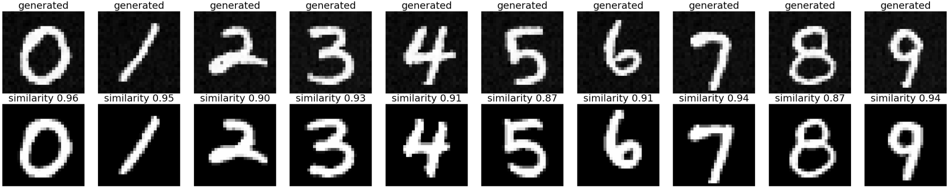


Figure: Generated digits (top) and their closest neighbors (bottom)

Conditional Generation

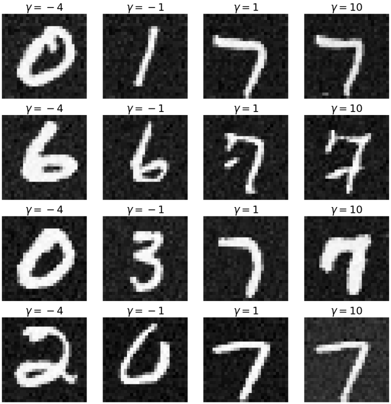
How to generate specific pictures on demand?

Conditional Generation

How to generate specific pictures on demand? Sample from the conditional distribution $\pi(x|c)$, e.g. $c =$ "digit 7". Control the strength of conditioning with γ:

$$f_{t,\gamma}(x|c) \propto f_t(x)^{-\gamma} f_t(x|c)^{1+\gamma}$$

$$\nabla \log_x f_{t,\gamma}(x|c) = -\gamma \nabla \log_x f_t(x) + (1+\gamma) \nabla \log_x f_t(x|c)$$



Mode Capturing

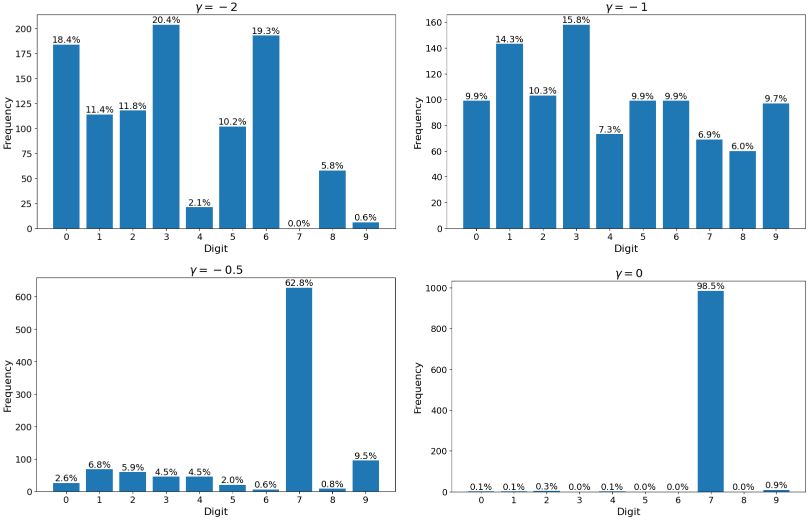


Figure: Digit frequencies conditioned on “7”, anti-conditional ($\gamma = -2$), unconditional ($\gamma = -1$) and conditional ($\gamma = -0.5, 0$).

Image Editing – Conditioning on Image+Text

Change his racing suit to red

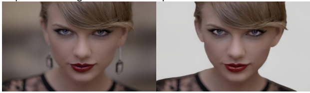
Make her hair pink

Colorize this photo

Make her clothes black and pink

Give Simba a crown

Replace the background with blank space



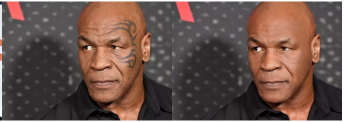
Fix his tooth

Make his hair messy

Give him a Christmas hat

Give him Joker's makeup

Remove his tattoo



Resources

- Yang Song's blog '21: "Generative Modeling by Estimating Gradients of the Data Distribution"
- Convergence paper, Chen et al '23: "Sampling is as easy as learning the score"