Consistency Models

Yang Song, Prafulla Dhariwal,
Mark Chen, and Ilya Sutskever
ICML 2023



Consistency training
One step (top), two steps (bottom)

Same initial noise for both rows.
Trained without a diffusion teacher.

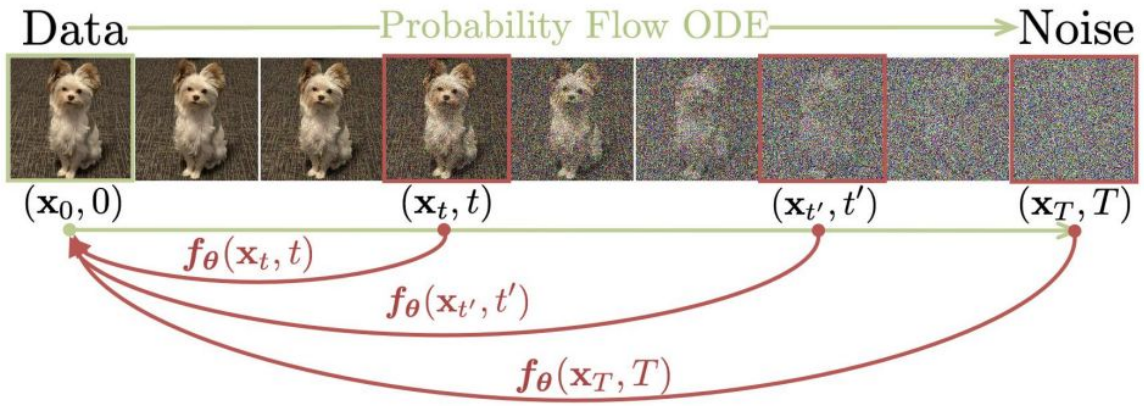
One-step generation through consistency training

Presented by: Zhixuan Zhou

Song et al. (2023), Fig. 5, cropped. arxiv.org/abs/2303.01469

Motivation:

Can generation use just one network evaluation?



A direct map from noise to a sample

The sampling bottleneck

Repeated denoising means repeated compute.

1. **Fast generation**
   Generate with one network evaluation.
2. **Optional refinement**
   Use more steps when quality matters.
3. **Flexible training**
   Learn with or without a diffusion teacher.

One-step sampling

$$\mathbf{x}_0 = f_\theta(\mathbf{x}_T, T)$$

NFE = number of network evaluations.

Song et al. (2023), Fig. 1 and Sec. 1. arxiv.org/abs/2303.01469

Diffusion background:

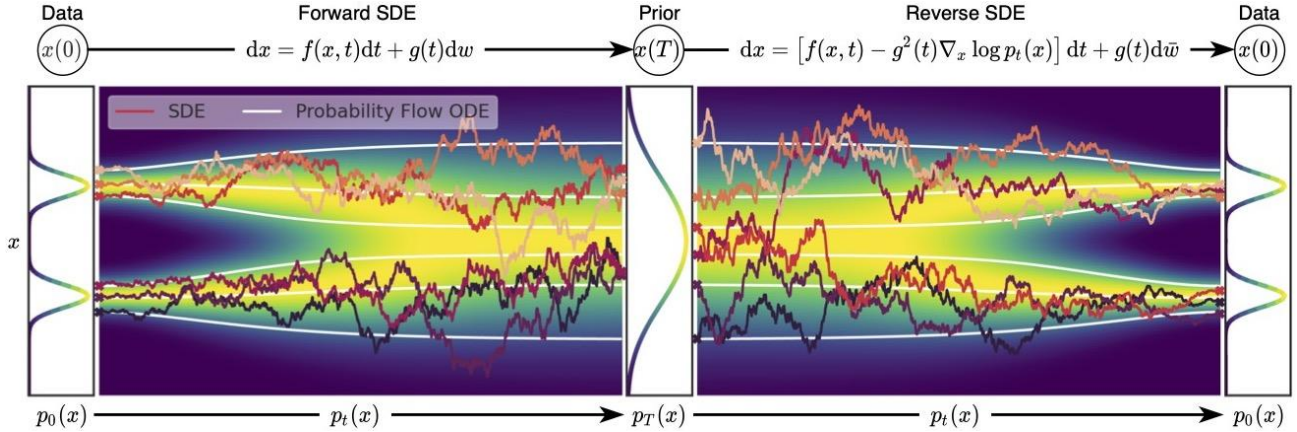
The probability flow ODE defines a trajectory

1 Add Gaussian noise to data

$$\mathbf{x}_t = \mathbf{x}_0 + t\,\mathbf{z}$$

t: noise standard deviation

$$\mathbf{x}_0 \sim p_{\text{data}}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$



2 Learn the score of noisy data

$$\mathbf{s}_\phi(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$$

3 Integrate the probability flow ODE backward

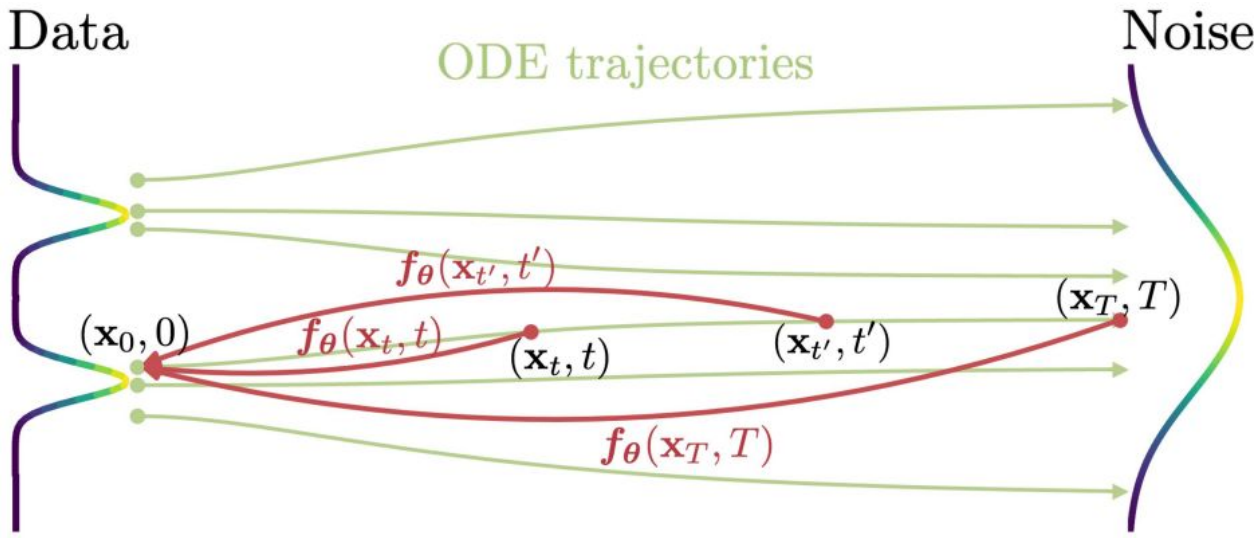
$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = \underbrace{\boldsymbol{\mu}(\mathbf{x}_t, t)}_{\mathbf{0}} - \frac{1}{2}\underbrace{\sigma(t)^2}_{2t}\,\mathbf{s}_\phi(\mathbf{x}_t, t) = -t\,\mathbf{s}_\phi(\mathbf{x}_t, t)$$

Sampling runs from T down to 0, so the time steps are negative.

Equations: Song et al. (2023), Sec. 2, Eqs. (2)–(3). Figure: Yang Song (2021), yang-song.net/blog/2021/score/

Consistency principle:

Changing inputs, same endpoint prediction



Data endpoint: x_0 at t = 0.

Along the same ODE trajectory

Input changes

The noisy state and noise level vary.

Prediction stays the same

$$f_\theta(\mathbf{x}_t, t) = f_\theta(\mathbf{x}_{t'}, t') = \mathbf{x}_0$$

Ideal consistency

Method:

Consistency training with an EMA target

Same image and noise. Two nearby noise levels, s < t.

Online model

$$f_\theta(\mathbf{x}_0 + t\,\mathbf{z}, t)$$

Update by gradient descent

$\approx$

EMA target model

$$f_{\bar{\theta}}(\mathbf{x}_0 + s\,\mathbf{z}, s)$$

Stop gradient: hold this prediction fixed

Make the two endpoint predictions agree.

Online weights

Every gradient update also moves the target, including noise from the current minibatch.

EMA = exponential moving average

Average past weights → a smoother target.

$$\bar{\theta} \leftarrow \mu\bar{\theta} + (1-\mu)\theta$$

Song et al. (2023), Eqs. (8), (10), Algorithm 3. EMA stabilizes the original training recipe.
Prediction distance: Learned Perceptual Image Patch Similarity (LPIPS). arxiv.org/abs/2303.01469

Boundary condition:

At the data end, return the input itself

$$f_\theta(\mathbf{x}, t) = \underbrace{c_{\text{skip}}(t)\,\mathbf{x}}_{\textbf{input path}} + \underbrace{c_{\text{out}}(t)\,g_\theta(\mathbf{x}, t)}_{\textbf{network path}}$$

Scale the input path. **Scale the network output.**

$$t = 0: \qquad c_{\text{skip}}(0) = 1, \qquad c_{\text{out}}(0) = 0$$

$$f_\theta(\mathbf{x}, 0) = 1 \cdot \mathbf{x} + 0 \cdot g_\theta(\mathbf{x}, 0) = \mathbf{x}$$

Song et al. (2023), Sec. 3, Eq. (5). Coefficients are fixed functions of the noise level.
Data end shown at t = 0 for clarity; the paper uses a small positive minimum noise level. arxiv.org/abs/2303.01469

Why it works:

Why not just regress to the original image?

$$\mathbf{x}_t = \mathbf{x}_0 + t\,\mathbf{z}, \quad \mathbf{x}_s = \mathbf{x}_0 + s\,\mathbf{z}, \quad s < t, \qquad D_\theta(\mathbf{x}_t, t) = \mathbf{x}_t - t\,\boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t)$$

D: original-image regression

$$\mathcal{L}_{\text{noise-pred}} = \mathbb{E}\left[t^2 \left\|\boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t) - \mathbf{z}\right\|_2^2\right]$$

$$\mathcal{L}_{\text{x-pred}} = \mathbb{E}\left[\left\|D_\theta(\mathbf{x}_t, t) - \mathbf{x}_0\right\|_2^2\right]$$



$$D^*(\mathbf{x}_t, t) = \mathbf{x}_t - t\,\mathbf{v}_{\text{PF}}(\mathbf{x}_t, t)$$
$$= \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$$

$$\mathbf{v}_{\text{PF}}(\mathbf{x}_t, t) = \mathbb{E}[\mathbf{z} \mid \mathbf{x}_t]$$

f: ODE endpoint prediction

$$\mathcal{L}_{\text{consist}} = \mathbb{E}\, d(f_\theta(\mathbf{x}_t, t), \text{sg}[f_{\bar{\theta}}(\mathbf{x}_s, s)])$$



$$f^*(\mathbf{x}_t, t) = \mathbf{x}_t - \int_0^t \mathbf{v}_{\text{PF}}(\mathbf{x}_r, r)\,\mathrm{d}r$$

sg: stop-gradient. Target network uses EMA. d: training distance (LPIPS).
Song et al. (2023), Secs. 2–5. arxiv.org/abs/2303.01469

Sampling:

Predict once, then optionally re-noise and repeat

Algorithm 1 Multistep consistency sampling

Input: Consistency model $f_\theta(\cdot,\cdot)$, time points $\tau_1 > \tau_2 > \cdots > \tau_{N-1}$, initial noise $\hat{\mathbf{x}}_T$

$\mathbf{x}_0 \leftarrow f_\theta(\hat{\mathbf{x}}_T, T)$

for $n = 1$ **to** $N-1$ **do**

  Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

  $\hat{\mathbf{x}}_{\tau_n} \leftarrow \mathbf{x}_0 + \tau_n \mathbf{z}$

  $\mathbf{x}_0 \leftarrow f_\theta(\hat{\mathbf{x}}_{\tau_n}, \tau_n)$

end for

Output: $\mathbf{x}_0$

T: starting noise level.

τ: chosen noise level for each re-noising step.

Start from Gaussian noise

$$\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, T^2\mathbf{I})$$

1 evaluation

First prediction only.
Skip the loop.

More evaluations

Add fresh noise to the output.
Predict the endpoint again.
Use a lower noise level next.

Each repeat adds one model call.

Source: Song et al. (2023), Algorithm 1, adapted with data end t = 0; Sec. 3. arxiv.org/abs/2303.01469

Method wrap-up:

Two ways to train the same endpoint map

Ingredient	Consistency distillation	Consistency training
Diffusion teacher	Required: pretrained score	No diffusion teacher
Training pair	Noisy sample + ODE step	Same x_0 and z; two scales
Target network	EMA consistency network	EMA consistency network
Main constraint	Endpoint predictions agree	Endpoint predictions agree
Generation	One step or multiple steps	One step or multiple steps

Shared boundary condition: $f_\theta(\mathbf{x}_0, 0) = \mathbf{x}_0$

Consistency training learns directly from data.

Song et al. (2023), Secs. 3–5. arxiv.org/abs/2303.01469

Experimental setup:

Compare quality at a fixed sampling budget

Dataset	Resolution
CIFAR-10	32 × 32
ImageNet	64 × 64
LSUN Bedroom	256 × 256
LSUN Cat	256 × 256

Two training settings

Consistency distill.: pretrained EDM teacher
Consistency training: learn from scratch

Relevant baselines

Progressive distillation
EDM: iterative diffusion
GANs, VAEs, and flows

NFE

Network evaluations per sample

FID

Distributional quality; lower is better

Precision / recall

Fidelity / coverage

Song et al. (2023), Sec. 6 and Appendix C. arxiv.org/abs/2303.01469

Results:

Distillation gives lower FID at the same step count

FID ↓ at one or two network evaluations.

Method	Teacher?	NFE	CIFAR-10	ImageNet 64
Progressive distillation	Yes	1	8.34	15.39
Consistency distillation	Yes	1	3.55	6.20
Consistency training	No	1	8.70	13.0
Progressive distillation	Yes	2	5.58	8.95
Consistency distillation	Yes	2	2.93	4.70
Consistency training	No	2	5.83	11.1

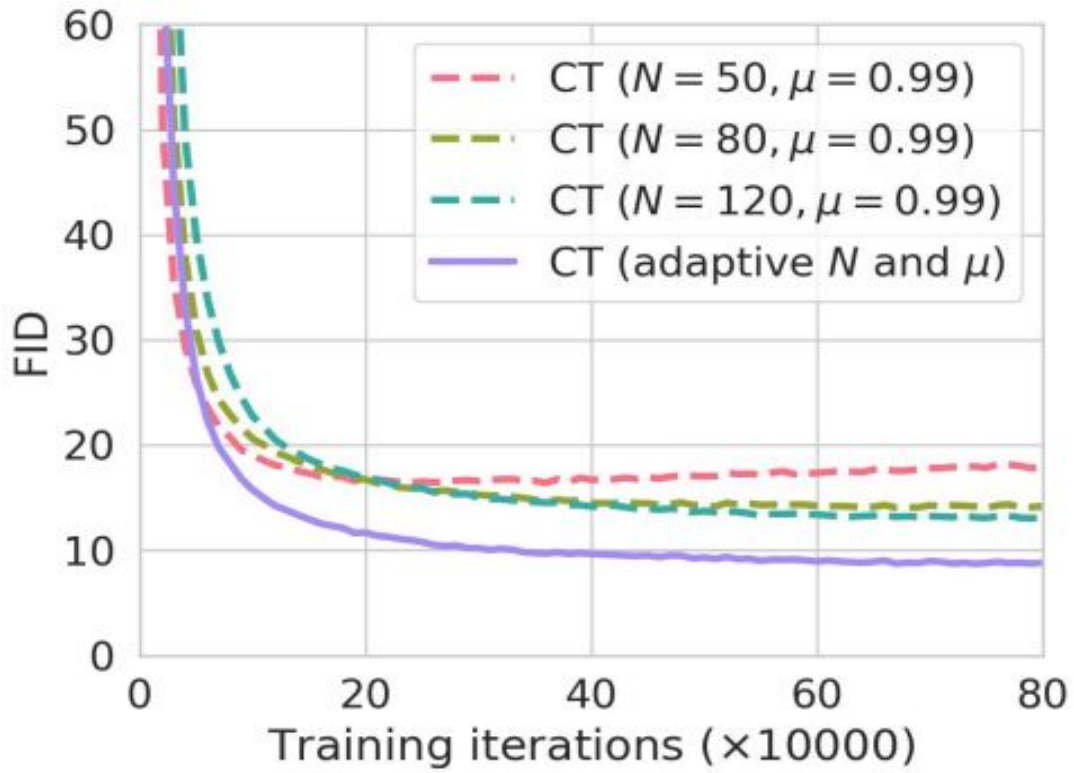
Two steps improve consistency training on both datasets.

Losses: progressive distill. uses squared error; consistency distill. and training use LPIPS.

Song et al. (2023), Tables 1–2; selected published values. arxiv.org/abs/2303.01469

Consistency training ablation:

Increase the number of noise levels during training



All curves: consistency training (original legend abbreviates it as CT).

Coarse to fine training

N: number of discrete noise levels.

1. **Start with fewer levels**
   Faster early convergence; coarser target.
2. **Gradually increase N**
   Finer spacing reduces discretization bias.
3. **Adjust the EMA decay μ**
   Target-network scheduling also matters.

A curriculum over noise scales

Fast early learning → finer consistency.

Song et al. (2023), Fig. 3(d), Sec. 5 and Sec. 6.1. arxiv.org/abs/2303.01469

Applications:

Use observation constraints during iterative sampling

Colorization



Grayscale input → plausible colors

No colorization-specific training.

Super-resolution



32 × 32 input → 256 × 256 outputs

Consistency distillation on LSUN Bedroom.

Song et al. (2023), Fig. 6(a–b), Sec. 6.3. Examples use consistency distillation. arxiv.org/abs/2303.01469

Critical assessment:

Strengths and limitations

Strengths

Direct generation by design

One-step sampling with multistep refinement

Direct training from data

Consistency training needs no diffusion teacher and avoids adversarial training.

Limitations

Direct training still trails distillation

Consistency training has higher FID at the same sampling budget.

Sensitive to training design

Quality depends on the noise schedule, EMA decay, and perceptual loss

Song et al. (2023), Secs. 3–6, Tables 1–2; presenter assessment. arxiv.org/abs/2303.01469

THANK YOU