Maximum Likelihood Training of Score-Based Diffusion Models

Yang Song*, Conor Durkan*, Iain Murray, Stefano Ermon
Stanford University and University of Edinburgh

ABSTRACT

- Score-based diffusion models synthesize images by reversing a stochastic process that diffuses data to noise, and are trained by minimizing a weighted combination of score matching losses.
- We show that by choosing a special weighting function, called the likelihood weighting, minimizing the weighted combination of score matching losses amounts to maximum likelihood training.
- Our theoretical results enable ScoreFlow, a continuous normalizing flow model trained with a variational objective, which is much more efficient than neural ODEs. We report the state-of-the-art likelihood on CIFAR-10 and ImageNet 32x32 among all flow models, achieving comparable performance to cutting-edge autoregressive models.

Code:

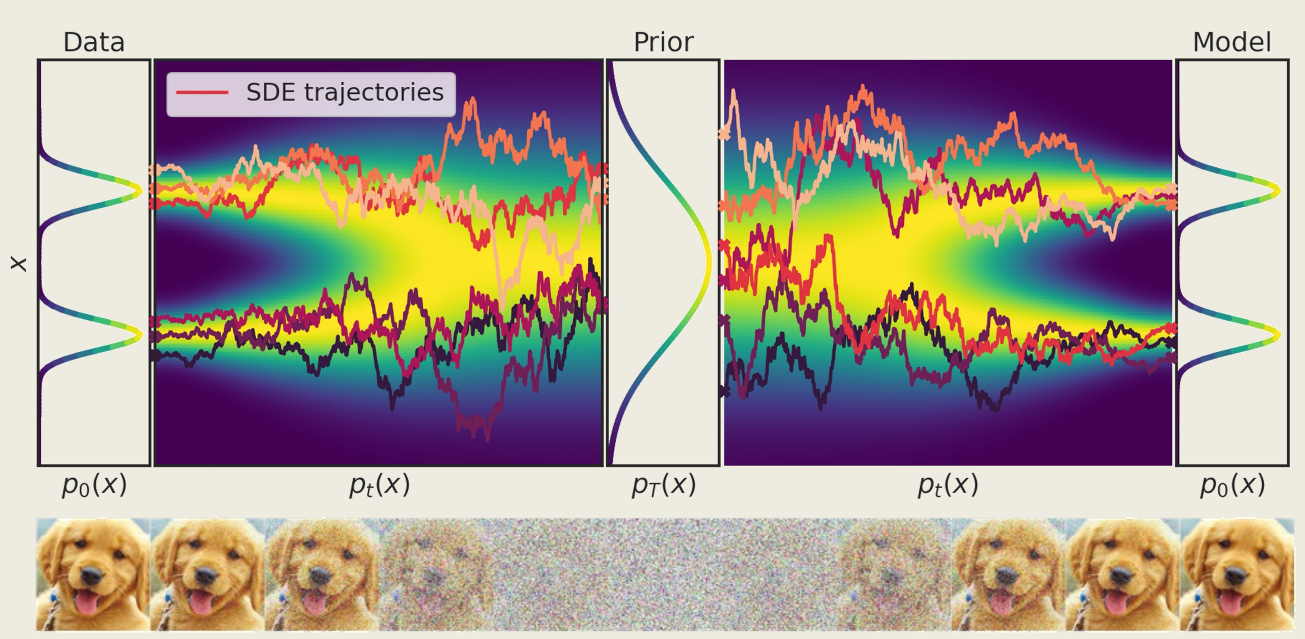
Score-Based Diffusion Models

Stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{x} = \underbrace{\boldsymbol{f}(\mathbf{x},t)\,\mathrm{d}t}_{\text{Deterministic drift}} + \underbrace{g(t)\,\mathrm{d}\mathbf{w}}_{\text{Stochastic diffusion}}$$

Brownian motion

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



The reverse-time SDE:

$$\mathrm{d}\mathbf{x} = [\boldsymbol{f}(\mathbf{x},t) - g^2(t)\underbrace{\nabla_{\mathbf{x}}\log p_t(\mathbf{x})}_{\text{Score function of } p_t(\mathbf{x})}]\,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}$$

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

Estimating the score function for the reverse 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)

Two Ways to Define Likelihoods

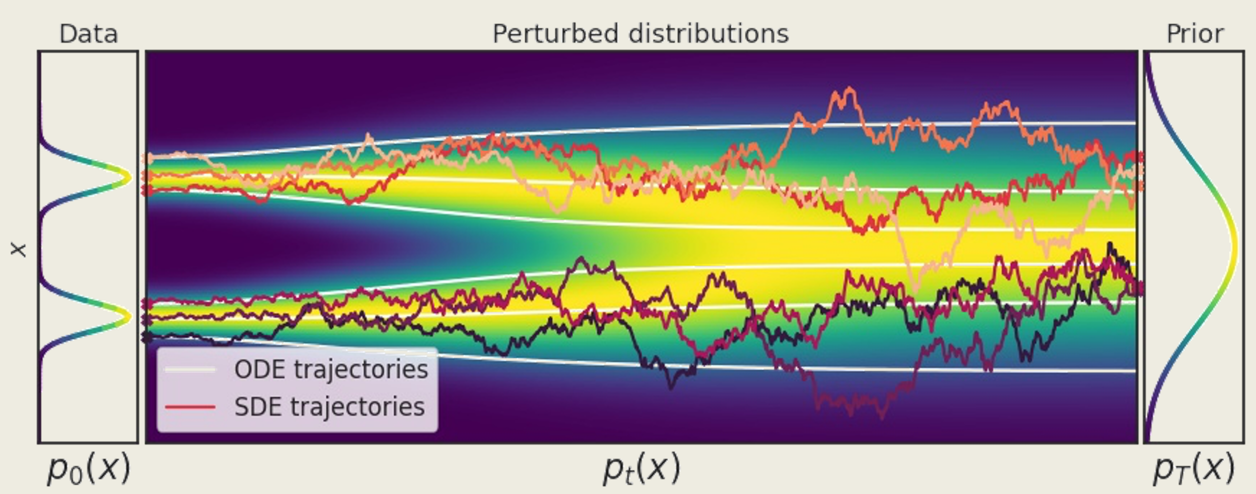
Probability flow ODE:

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

$$\{p_t(\mathbf{x})\}_{t\in[0,T]}$$

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

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



Prior distribution: $\pi(\mathbf{x}) \approx p_T(\mathbf{x})$

Model 1: Reversing the probability flow ODE

$$\mathrm{d}\tilde{\mathbf{x}} = \left[\boldsymbol{f}(\tilde{\mathbf{x}},t) - \frac{1}{2}g(t)^2\boldsymbol{s}_{\boldsymbol{\theta}^*}(\tilde{\mathbf{x}},t)\right]\mathrm{d}t, \quad \tilde{\mathbf{x}}(T)\sim\pi$$

$$\tilde{\mathbf{x}}(0)\sim p_{\boldsymbol{\theta}^*}^{\mathrm{ODE}}$$

Model 2: Reversing the SDE

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

$$\hat{\mathbf{x}}(0)\sim p_{\boldsymbol{\theta}^*}^{\mathrm{SDE}}$$

$\boldsymbol{s}_{\boldsymbol{\theta}^*}(\mathbf{x},t)$, π $\xrightarrow{\text{Reverse ODE}}$ $p_{\boldsymbol{\theta}^*}^{\mathrm{ODE}}(\mathbf{x})$

$\boldsymbol{s}_{\boldsymbol{\theta}^*}(\mathbf{x},t)$, π $\xrightarrow{\text{Reverse SDE}}$ $p_{\boldsymbol{\theta}^*}^{\mathrm{SDE}}(\mathbf{x})$

Maximum Likelihood Training

Theorem: KL divergence is upper-bounded by the weighted combination of score matching.

"Likelihood weighting"

$$\mathrm{KL}(p_{\mathrm{data}}\,\|\,p_{\boldsymbol{\theta}^*}^{\mathrm{SDE}}) \leq \frac{1}{2}\mathbb{E}_{t\sim\mathrm{Uniform}[0,T]}[\boxed{g(t)^2}\,\mathbb{E}_{p_t(\mathbf{x})}[\|\nabla_{\mathbf{x}}\log p_t(\mathbf{x}) - \boldsymbol{s}_{\boldsymbol{\theta}^*}(\mathbf{x},t)\|_2^2]] + \boxed{\mathrm{KL}(p_T\,\|\,\pi)} \approx 0$$

Theorem: There exists an efficiently computable variational lower bound to $\log p_{\boldsymbol{\theta}^*}^{\mathrm{SDE}}(\mathbf{x})$, analogous to the evidence lower bound of variational inference or Variational Auto-Encoders (VAEs).

ScoreFlow

$\pi(\mathbf{x})$	$\pi(\mathbf{x})$
Weighted combination of score matching with our likelihood weighting → Efficient maximum likelihood Training	Instantaneous change-of-variable formula (Chen et al. 2018) → Slow but accurate testing
$\leq \log p_{\boldsymbol{\theta}^*}^{\mathrm{SDE}}(\mathbf{x})$	$\log p_{\boldsymbol{\theta}^*}^{\mathrm{ODE}}(\mathbf{x})$

Remarks:

- Under ideal conditions: $\log p_{\boldsymbol{\theta}^*}^{\mathrm{SDE}}(\mathbf{x}) \approx \log p_{\boldsymbol{\theta}^*}^{\mathrm{ODE}}(\mathbf{x})$
- Importance sampling w.r.t. variable t to reduce the variance when estimating expectations.
- Variational dequantization for comparing with models trained on discrete data.

Empirical Results

Negative log-likelihood (bits/dim) and sample quality (FID scores) on CIFAR-10 and ImageNet 32ˆ32.

Model	SDE	CIFAR-10 Uni. deq. NLL↓	CIFAR-10 Uni. deq. Bound↓	CIFAR-10 Var. deq. NLL↓	CIFAR-10 Var. deq. Bound↓	CIFAR-10 FID↓	ImageNet 32×32 Uni. deq. NLL↓	ImageNet 32×32 Uni. deq. Bound↓	ImageNet 32×32 Var. deq. NLL↓	ImageNet 32×32 Var. deq. Bound↓	ImageNet 32×32 FID↓
Baseline	VP	3.16	3.28	3.04	3.14	3.98	3.90	3.96	3.84	3.91	**8.34**
Baseline + LW	VP	3.06	3.18	2.94	3.03	5.18	3.91	3.96	3.86	3.92	17.75
Baseline + LW + IS	VP	2.95	3.08	2.83	2.94	6.03	3.86	3.92	3.80	3.88	11.15
Deep	VP	3.13	3.25	3.01	3.10	3.09	3.89	3.95	3.84	3.90	8.40
Deep + LW	VP	3.06	3.17	2.93	3.02	7.88	3.91	3.96	3.86	3.92	17.73
Deep + LW + IS	VP	2.93	3.06	**2.80**	2.92	5.34	3.85	3.92	3.79	3.88	11.20
Baseline	subVP	2.99	3.09	2.88	2.98	3.20	3.87	3.92	3.82	3.88	8.71
Baseline + LW	subVP	2.97	3.07	2.86	2.96	7.33	3.87	3.92	3.82	3.88	12.99
Baseline + LW + IS	subVP	2.94	3.05	2.84	2.94	5.58	3.84	3.91	3.79	3.87	10.57
Deep	subVP	2.96	3.06	2.85	2.95	**2.86**	3.86	3.91	3.81	3.87	8.87
Deep + LW	subVP	2.95	3.05	2.85	2.94	6.57	3.88	3.93	3.83	3.88	16.55
Deep + LW + IS	subVP	**2.90**	**3.02**	2.81	**2.90**	5.40	**3.82**	**3.90**	**3.76**	**3.86**	10.18

NLLs on CIFAR-10 and ImageNet 32x32 without any data augmentation

Model	CIFAR-10	ImageNet
FFJORD [15]	3.40	-
Flow++ [18]	3.08	3.86
Gated PixelCNN [35]	3.03	3.83
VFlow [4]	2.98	3.83
PixelCNN++ [40]	2.92	-
NVAE [54]	2.91	3.92
Image Transformer [36]	2.90	3.77
Very Deep VAE [8]	2.87	3.80
PixelSNAIL [7]	2.85	3.80
δ-VAE [38]	2.83	3.77
Sparse Transformer [9]	**2.80**	-
ScoreFlow (Ours)	2.83	**3.76**