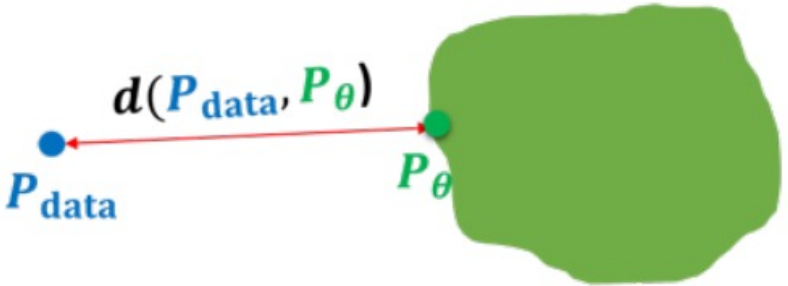
Score-Based Models

LECTURE 14

CS236: Deep Generative Models

Summary

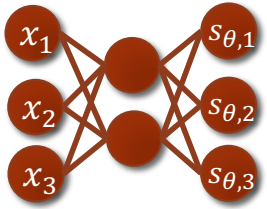
$$x_i \sim P_{\text{data}}$$
$$i = 1, 2, \ldots, n$$

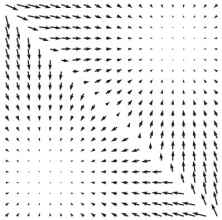


Score-based models

- A model that represents the score function

$$s_\theta(\mathbf{x})$$



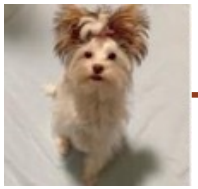


- **Score estimation:** training the score-based model from datapoints
- Score matching

$$\frac{1}{2}E_{\mathbf{x}\sim p_{\text{data}}}[\|\nabla_{\mathbf{x}}\log p_{\text{data}}(\mathbf{x}) - \boldsymbol{s}_\theta(\mathbf{x})\|_2^2]$$
$$= \frac{1}{2}E_{\mathbf{x}\sim p_{\text{data}}}\left[\frac{1}{2}\|\boldsymbol{s}_\theta(\mathbf{x})\|_2^2 + \text{tr}(\nabla_{\mathbf{x}}\boldsymbol{s}_\theta(\mathbf{x}))\right] + \text{const.}$$

- Not scalable for training deep score-based models.

Denoising score matching



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

Data distribution

$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$

Perturbation distribution/kernel

$\tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}})$

Noise-perturbed data distribution

$$\frac{1}{2} E_{\tilde{\mathbf{x}} \sim p_{\text{data}}}[\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) \|_2^2]$$

$$= \frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x}), \tilde{\mathbf{x}} \sim q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})}[\| \boldsymbol{s}_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x}) \|_2^2] + \text{const.}$$

$$= \frac{1}{2} E_{\mathbf{x} \sim p_{\text{data}}, \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\left\| \boldsymbol{s}_\theta(\mathbf{x} + \sigma \mathbf{z}) + \frac{\mathbf{z}}{\sigma} \right\|_2^2 \right] + \text{const.}$$

- **Pros:**
  - Much more scalable than score matching;
  - connection to optimal denoising.
- **Con:**

$$\boldsymbol{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log q_\sigma(\mathbf{x}) \neq \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$$

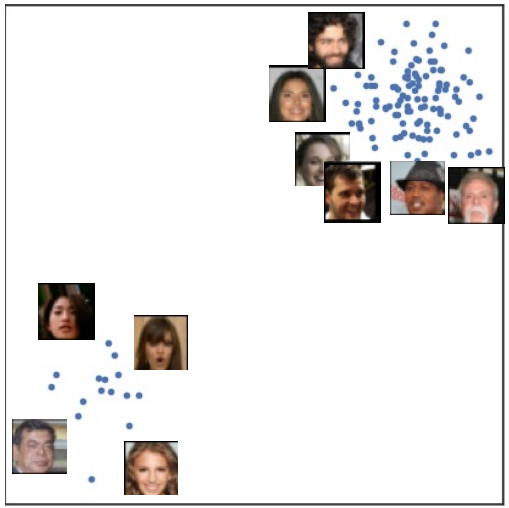
Sliced score matching

$$\frac{1}{2} E_{\mathbf{v}\sim p_{\mathbf{v}}} E_{\mathbf{x}\sim p_{\text{data}}} [(\mathbf{v}^\intercal \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \mathbf{v}^\intercal \boldsymbol{s}_\theta(\mathbf{x}))^2]$$

$$= E_{\mathbf{v}\sim p_{\mathbf{v}}, \mathbf{x}\sim p_{\text{data}}} \left[\mathbf{v}^\intercal \nabla_{\mathbf{x}} \boldsymbol{s}_\theta(\mathbf{x}) \mathbf{v} + \frac{1}{2} (\mathbf{v}^\intercal \boldsymbol{s}_\theta(\mathbf{x}))^2 \right] + \text{const.}$$

- Projection distribution $p_{\mathbf{v}}$ can be Gaussian or Rademacher
- **Pros:**
  - Much more scalable than score matching;
  - Estimates the true data score.
- **Con**: Slower than denoising score matching.

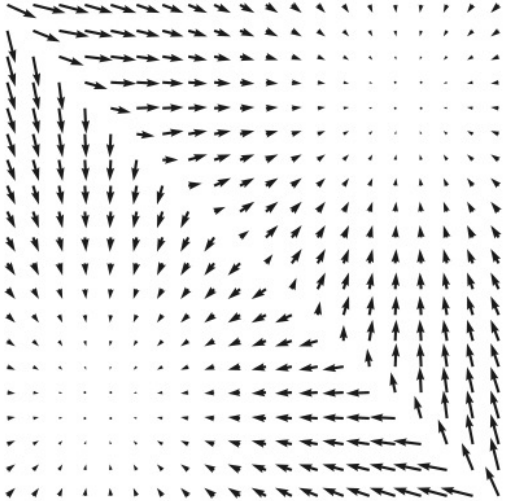
Score-based generative modeling



Data samples

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \sim p_{\text{data}}(\mathbf{x})$$

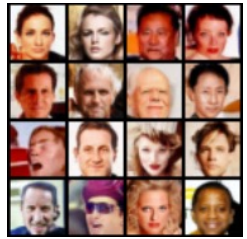
score matching



Scores

$$\boldsymbol{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$$

Langevin dynamics

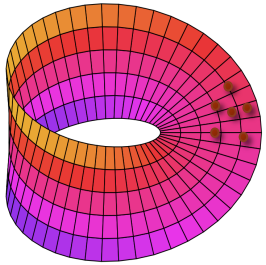


New samples

Pitfalls

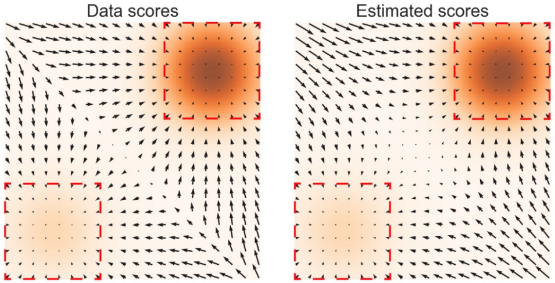
- Manifold hypothesis. Data score is undefined.

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

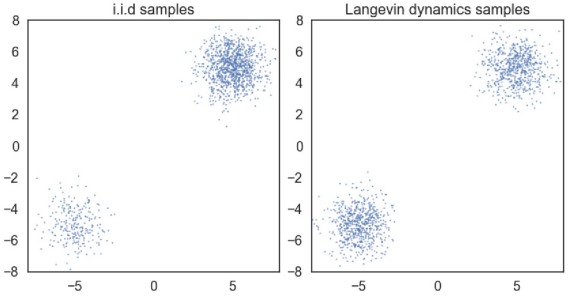


Data points

- Score matching fails in low data density regions

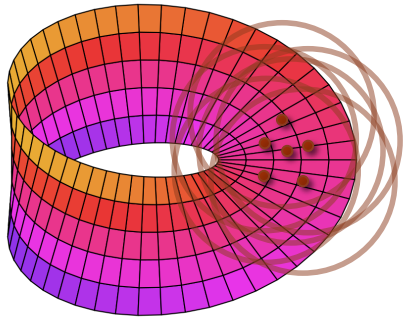


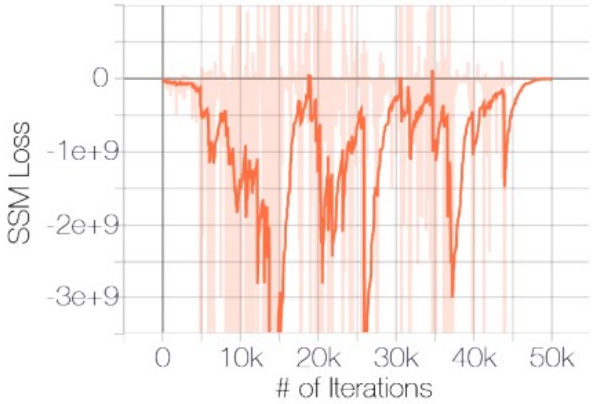
- Langevin dynamics fail to weight different modes correctly



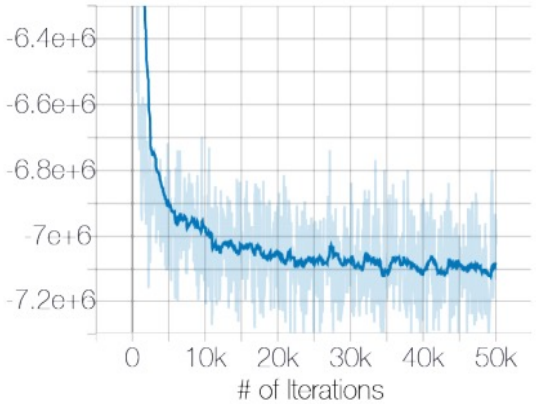
Gaussian perturbation

- The solution to all pitfalls: **Gaussian perturbation!**
- Manifold + noise
- Score matching on noisy data.





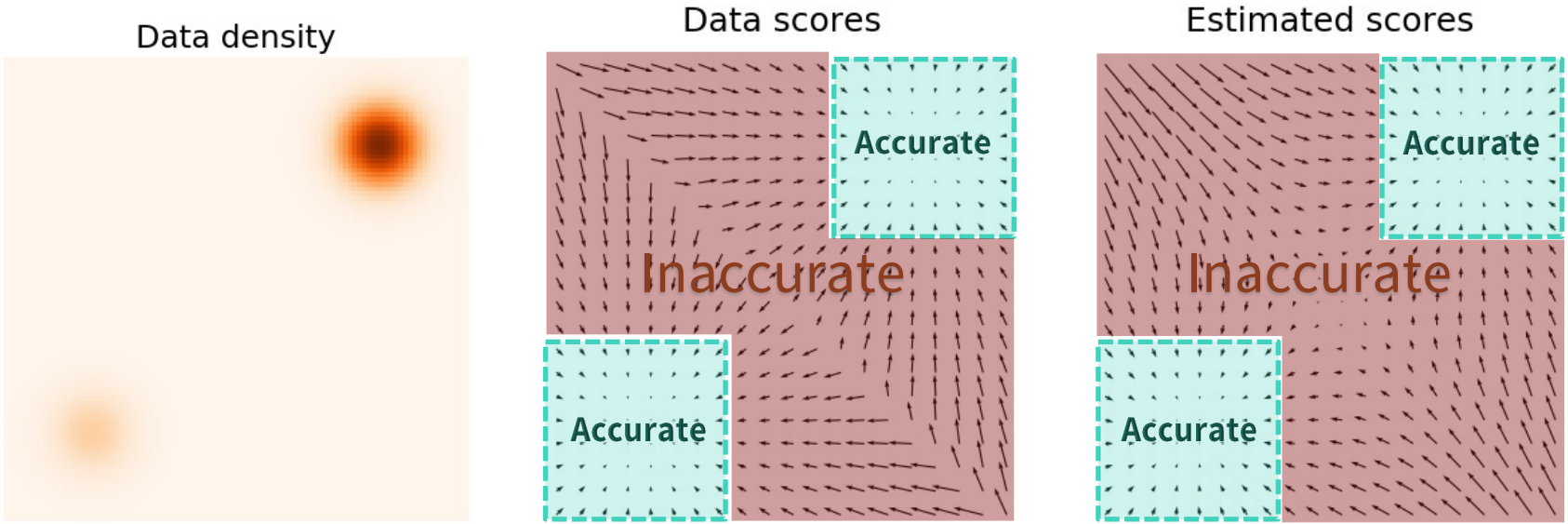
CIFAR-10



Noisy CIFAR-10

$$\mathcal{N}(0; 0.0001)$$

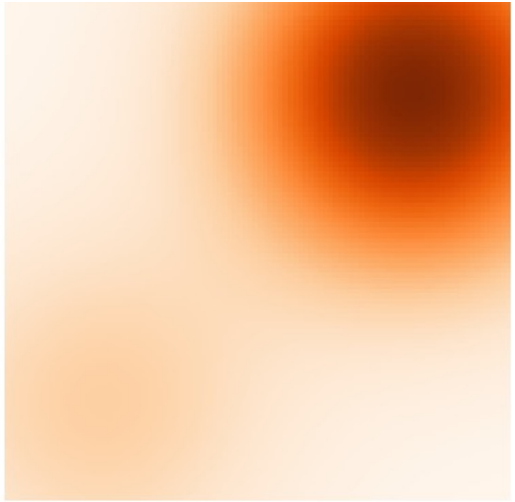
Challenge in low data density regions



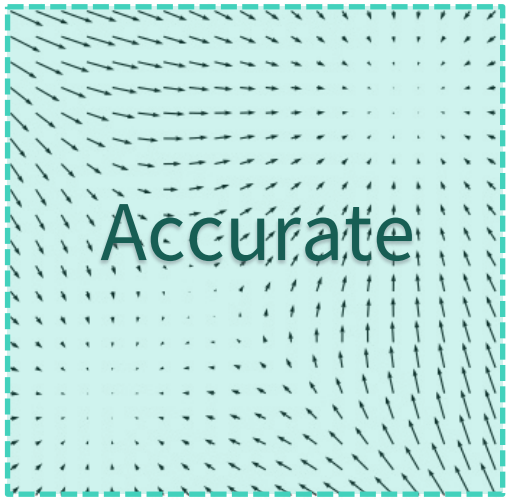
Song and Ermon. “Generative Modeling by Estimating Gradients of the Data Distribution.” NeurIPS 2019.

Adding noise to data

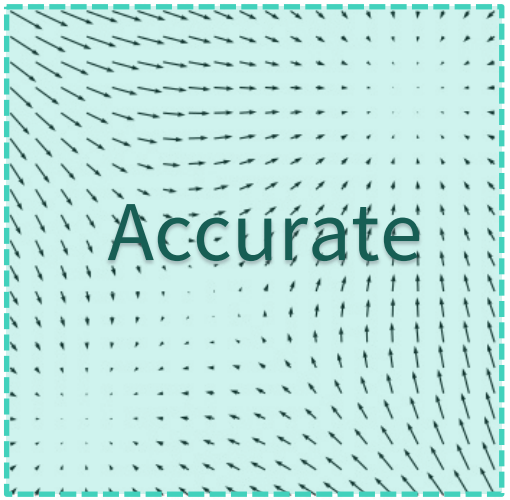
Perturbed density



Perturbed scores



Estimated scores

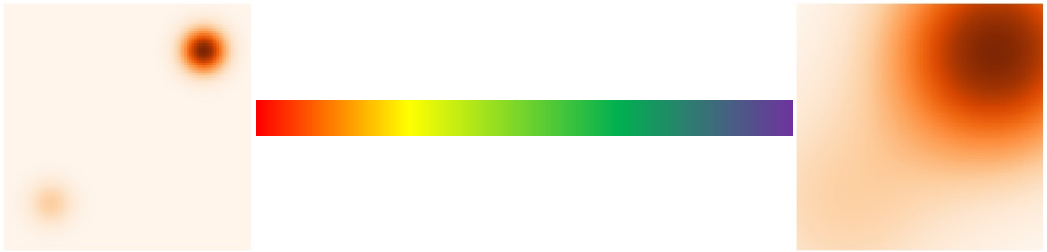


Provide useful directional information for Langevin MCMC.

Song and Ermon. “Generative Modeling by Estimating Gradients of the Data Distribution.” NeurIPS 2019.

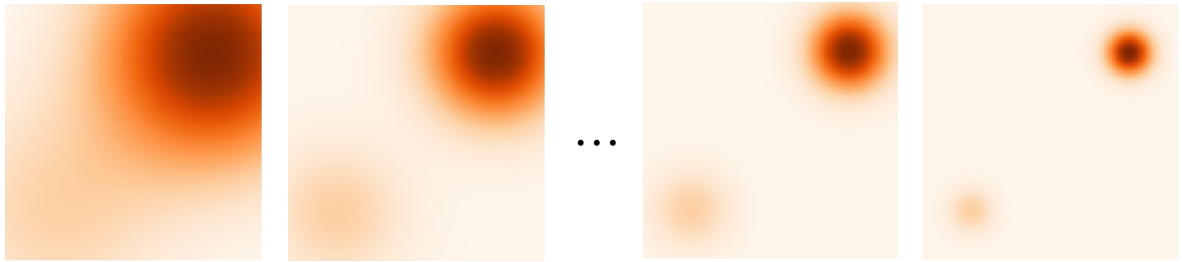
Multi-scale Noise Perturbation

- Trade-off

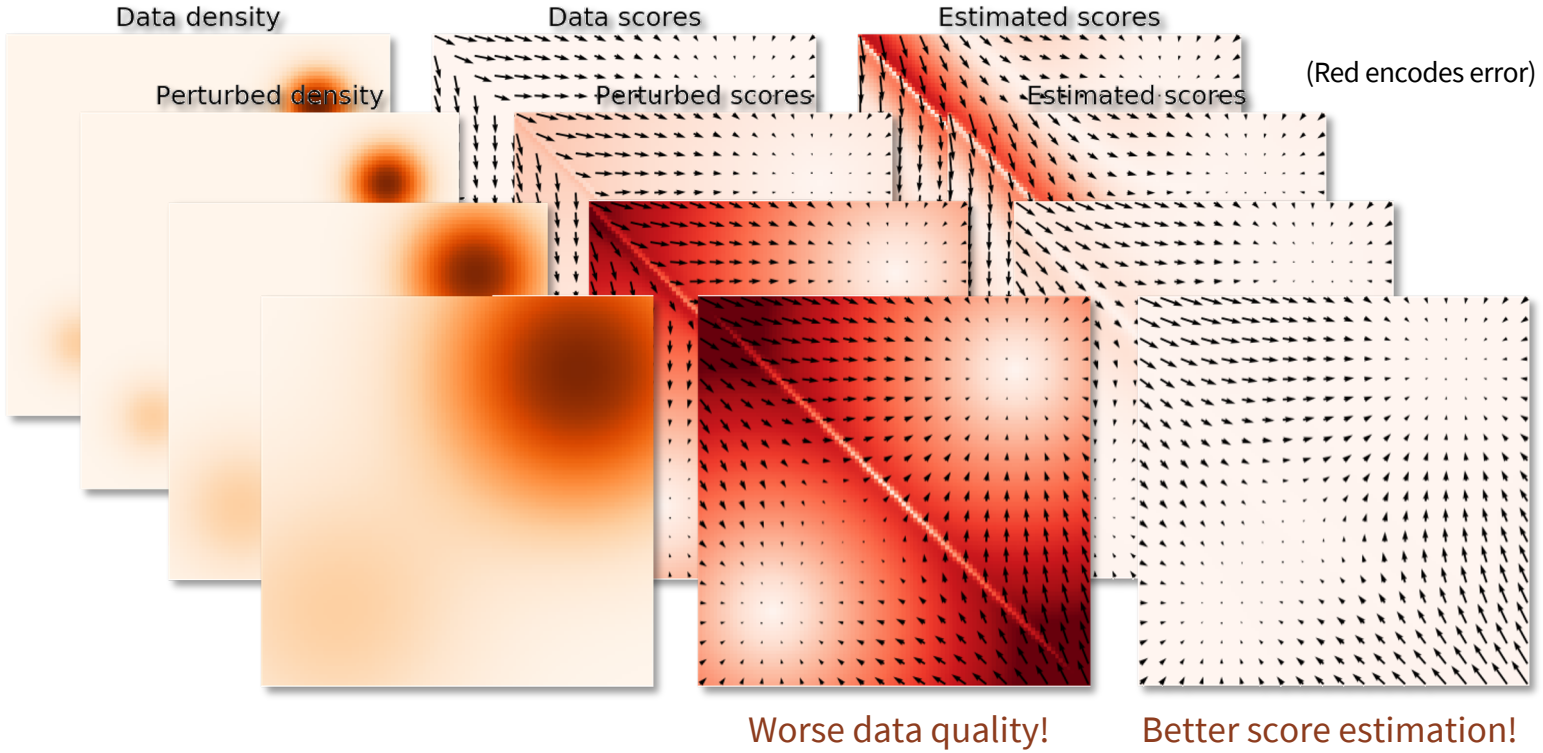


- Multi-scale noise perturbations.

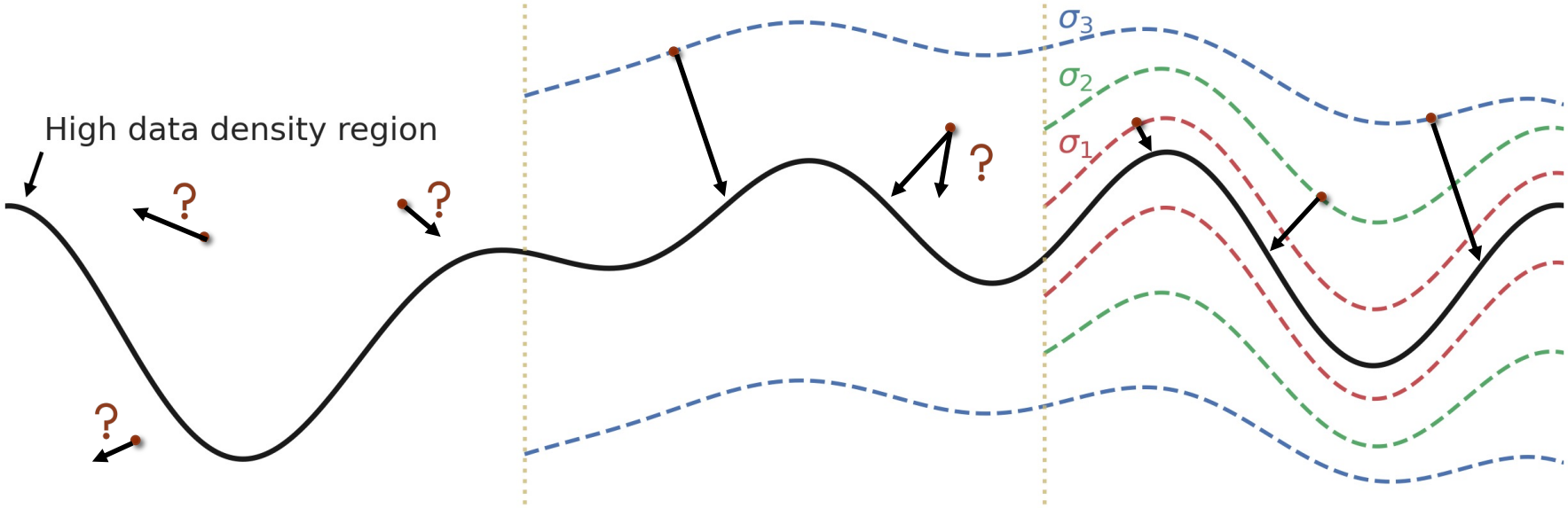
$$\sigma_1 > \sigma_2 > \cdots > \sigma_{L-1} > \sigma_L$$



Trading off Data Quality and Estimation Accuracy

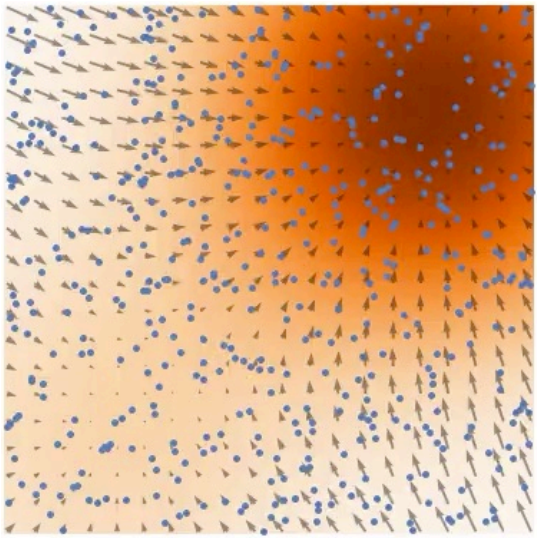


Using multiple noise scales

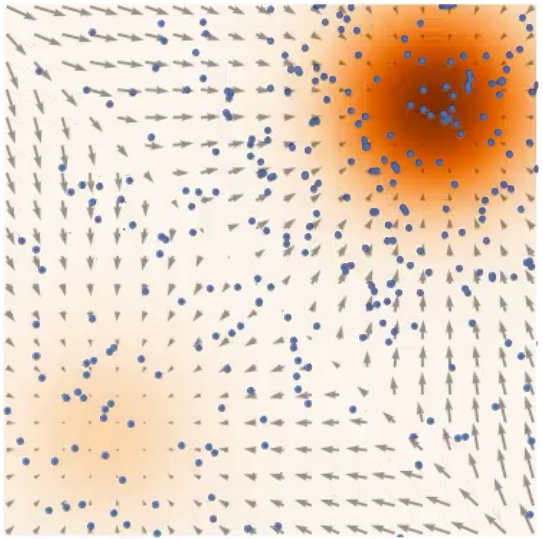


Annealed Langevin Dynamics: Joint Scores to Samples

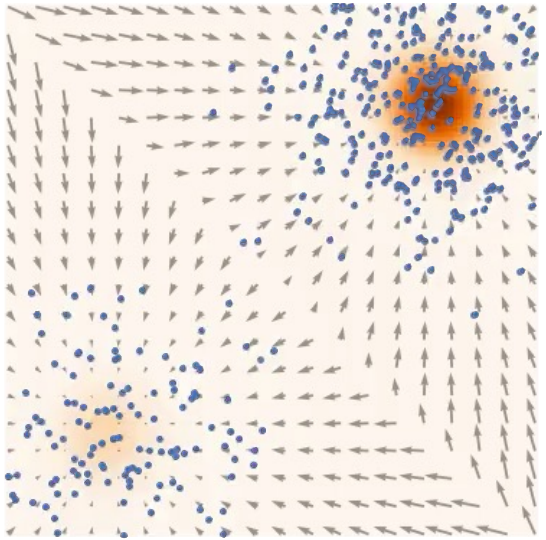
- Sample using $\sigma_1, \sigma_2, \cdots, \sigma_L$ sequentially with Langevin dynamics.
- Anneal down the noise level.
- Samples used as initialization for the next level.



σ_1



σ_2



σ_3

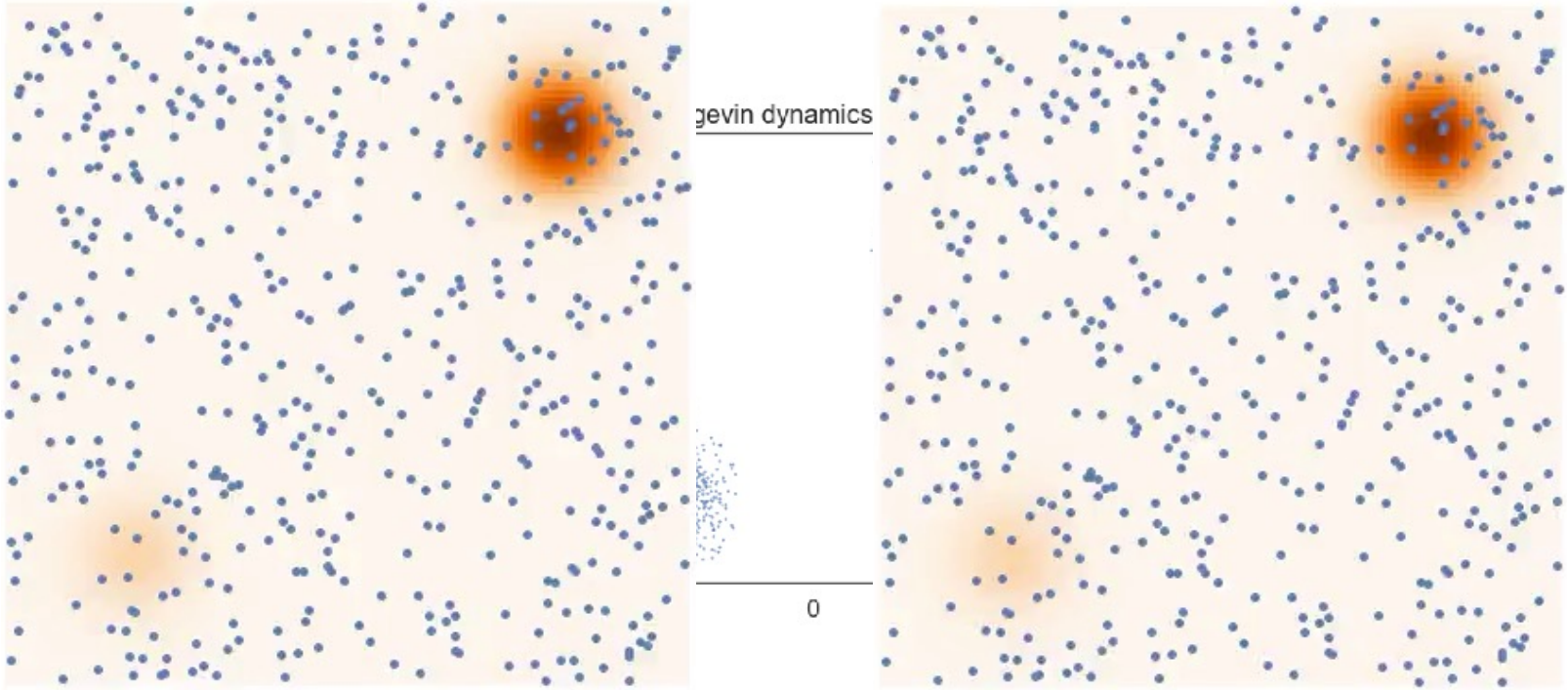
Annealed Langevin dynamics

Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$.

Initialize $\tilde{\mathbf{x}}_0$
for $i \leftarrow 1$ to L **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$ ▷ α_i is the step size.
  for $t \leftarrow 1$ to T **do**
    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$
  end for
  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
end for
return $\tilde{\mathbf{x}}_T$

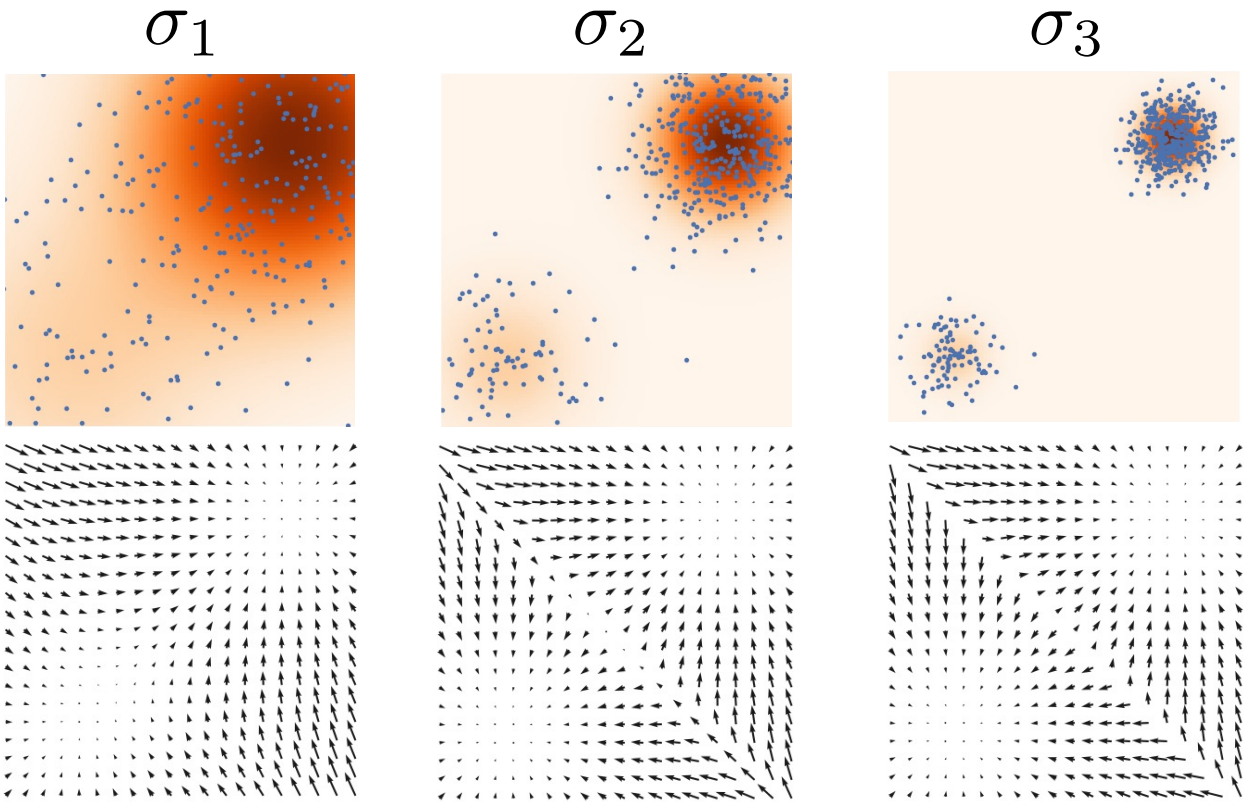
Comparison to the vanilla Langevin dynamics

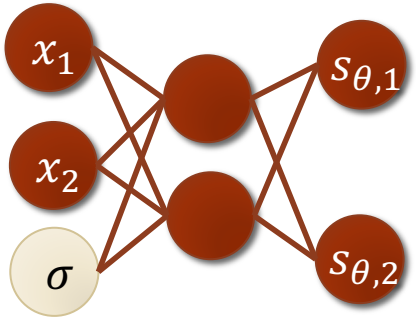


Langevin dynamics

Annealed Langevin dynamics

Joint Score Estimation via Noise Conditional Score Networks





Noise Conditional Score Network (NCSN)

Stanford University

Training noise conditional score networks

- Which score matching loss?
  - Sliced score matching?
  - Denoising score matching?
- Denoising score matching is naturally suitable, since the goal is to estimate the score of perturbed data distributions.
- Weighted combination of denoising score matching losses

$$\frac{1}{L}\sum_{i=1}^{L}\lambda(\sigma_i)E_{q_{\sigma_i}(\mathbf{x})}[\|\nabla_{\mathbf{x}}\log q_{\sigma_i}(\mathbf{x}) - \boldsymbol{s}_\theta(\mathbf{x},\sigma_i)\|_2^2]$$

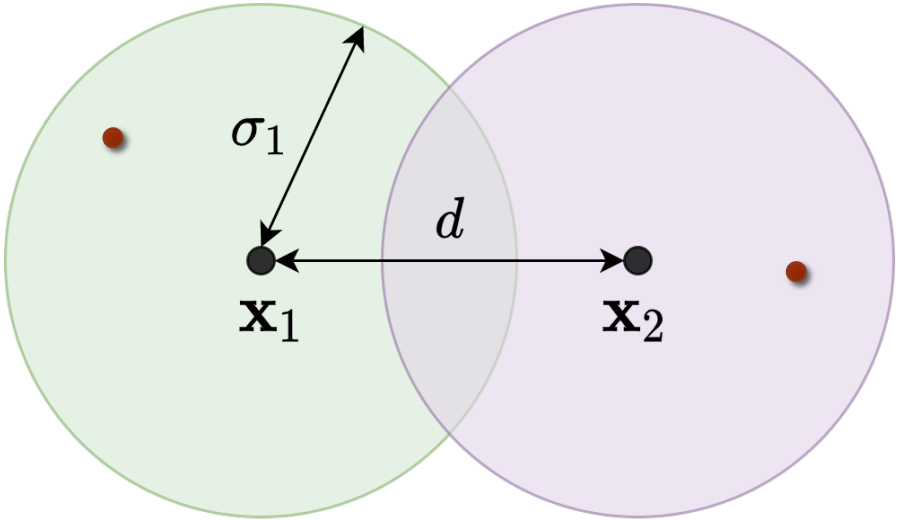
$$= \frac{1}{L}\sum_{i=1}^{L}\lambda(\sigma_i)E_{\mathbf{x}\sim p_{\text{data}},\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\|\nabla_{\tilde{\mathbf{x}}}\log q_{\sigma_i}(\tilde{\mathbf{x}}\mid\mathbf{x}) - \boldsymbol{s}_\theta(\tilde{\mathbf{x}},\sigma_i)\|_2^2] + \text{const.}$$

$$= \frac{1}{L}\sum_{i=1}^{L}\lambda(\sigma_i)E_{\mathbf{x}\sim p_{\text{data}},\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\left\|\boldsymbol{s}_\theta(\mathbf{x}+\sigma_i\mathbf{z},\sigma_i) + \frac{\mathbf{z}}{\sigma_i}\right\|_2^2\right] + \text{const.}$$

Choosing noise scales

- Maximum noise scale

$$\sigma_1 \approx \text{maximum pairwise distance between datapoints}$$



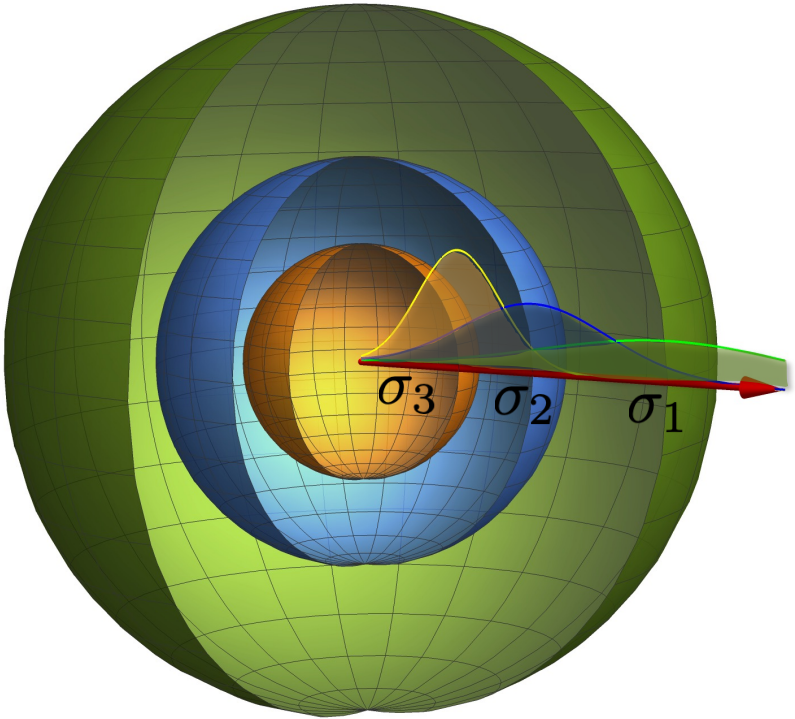
- Minimum noise scale: σ_L should be sufficiently small to control the noise in final samples.

Choosing noise scales

- **Key intuition:** adjacent noise scales should have sufficient overlap to facilitate transitioning across noise scales in annealed Langevin dynamics.
- A geometric progression with sufficient length.

$$\sigma_1 > \sigma_2 > \sigma_3 > \cdots > \sigma_{L-1} > \sigma_L$$

$$\frac{\sigma_1}{\sigma_2} = \frac{\sigma_2}{\sigma_3} = \cdots = \frac{\sigma_{L-1}}{\sigma_L}$$



Choosing the weighting function

- Weighted combination of denoising score matching losses

$$\frac{1}{L}\sum_{i=1}^{L}\lambda(\sigma_i)E_{\mathbf{x}\sim p_{\text{data}},\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\left\|\boldsymbol{s}_\theta(\mathbf{x}+\sigma_i\mathbf{z},\sigma_i)+\frac{\mathbf{z}}{\sigma_i}\right\|_2^2\right]$$

- How to choose the weighting function $\lambda : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$?
- **Goal:** balancing different score matching losses in the sum → $\lambda(\sigma_i) = \sigma_i^2$

$$\begin{aligned}
&\frac{1}{L}\sum_{i=1}^{L}\sigma_i^2 E_{\mathbf{x}\sim p_{\text{data}},\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\left\|\boldsymbol{s}_\theta(\mathbf{x}+\sigma_i\mathbf{z},\sigma_i)+\frac{\mathbf{z}}{\sigma_i}\right\|_2^2\right] \\
=&\frac{1}{L}\sum_{i=1}^{L}E_{\mathbf{x}\sim p_{\text{data}},\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\left\|\sigma_i\boldsymbol{s}_\theta(\mathbf{x}+\sigma_i\mathbf{z},\sigma_i)+\mathbf{z}\right\|_2^2\right] \\
=&\frac{1}{L}\sum_{i=1}^{L}E_{\mathbf{x}\sim p_{\text{data}},\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\left\|\boldsymbol{\epsilon}_\theta(\mathbf{x}+\sigma_i\mathbf{z},\sigma_i)+\mathbf{z}\right\|_2^2\right] \qquad \left[\ \boldsymbol{\epsilon}_\theta(\cdot,\sigma_i) := \sigma_i\boldsymbol{s}_\theta(\cdot,\sigma_i)\ \right]
\end{aligned}$$

Training noise conditional score networks

- Sample a mini-batch of datapoints $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \sim p_{\text{data}}$
- Sample a mini-batch of noise scale indices

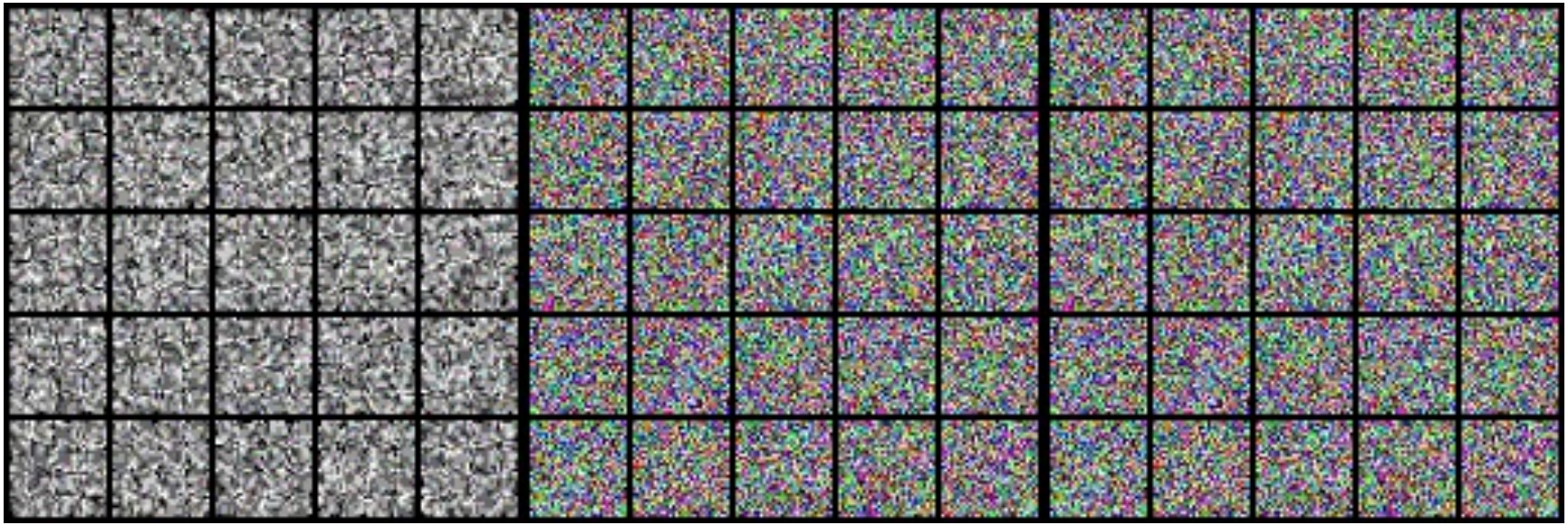
$$\{i_1, i_2, \cdots, i_n\} \sim \mathcal{U}\{1, 2, \cdots, L\}$$

- Sample a mini-batch of Gaussian noise $\{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n\} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
- Estimate the weighted mixture of score matching losses

$$\frac{1}{n} \sum_{k=1}^{n} \left[\left\| \sigma_{i_k} \boldsymbol{s}_\theta(\mathbf{x}_k + \sigma_{i_k} \mathbf{z}_k, \sigma_{i_k}) + \mathbf{z}_k \right\|_2^2 \right]$$

- Stochastic gradient descent
- As efficient as training one single non-conditional score-based model

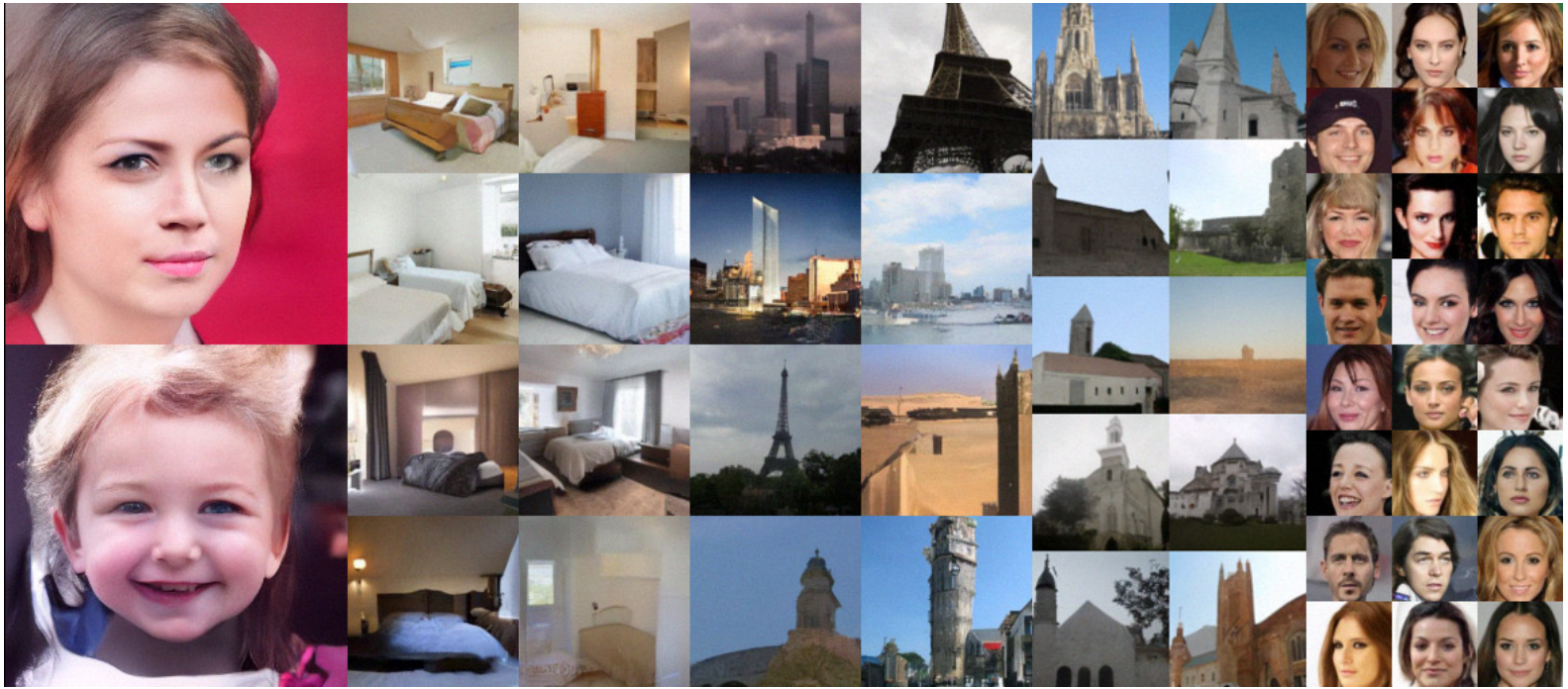
Experiments: Sampling



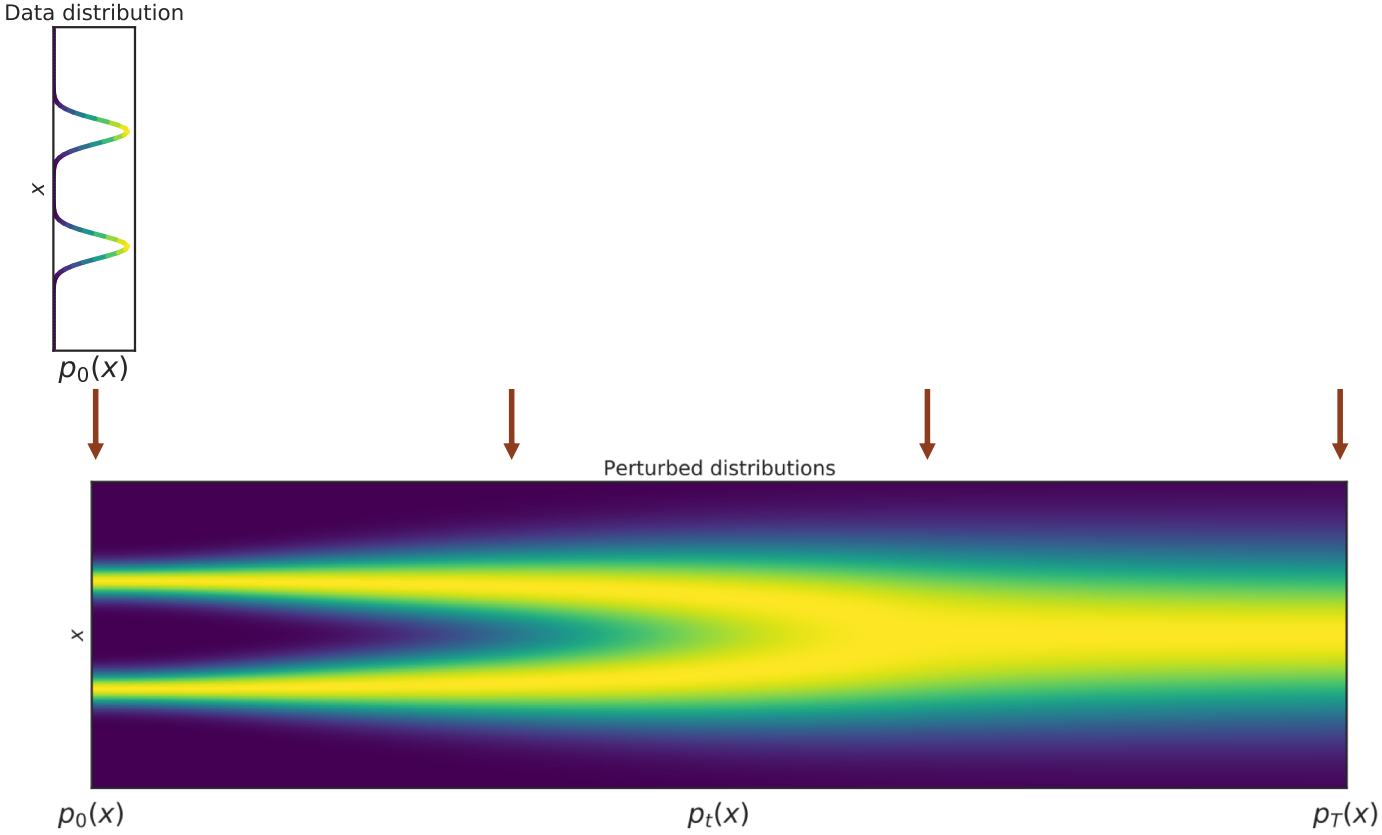
Experiments: sampling

Model	Inception	FID
CIFAR-10 Unconditional		
PixelCNN [59]	4.60	65.93
PixelIQN [42]	5.29	49.46
EBM [12]	6.02	40.58
WGAN-GP [18]	$7.86 \pm .07$	36.4
MoLM [45]	$7.90 \pm .10$	**18.9**
SNGAN [36]	$8.22 \pm .05$	21.7
ProgressiveGAN [25]	$8.80 \pm .05$	-
NCSN (Ours)	$\mathbf{8.87} \pm .12$	25.32

High Resolution Image Generation

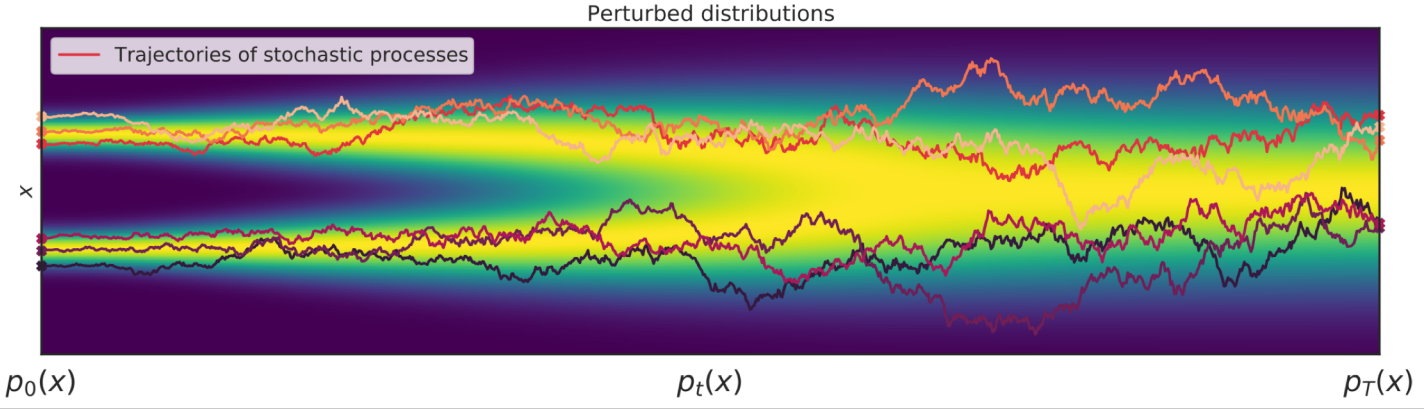


Using an infinite number of noise scales



$t \in [0, T]$: continuous index of perturbed distributions

Compact representation of infinite distributions



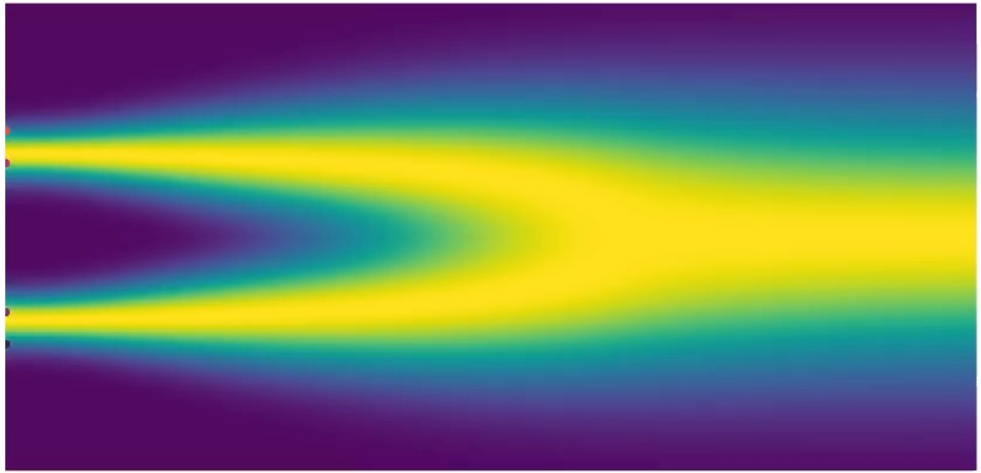
- Stochastic process $\{\mathbf{x}(t)\}_{t=0}^{T}$ → Marginal probability densities $\{p_t(\mathbf{x})\}_{t=0}^{T}$
- Stochastic differential equation: $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + \sigma(t)\mathrm{d}\mathbf{w}$

Deterministic drift

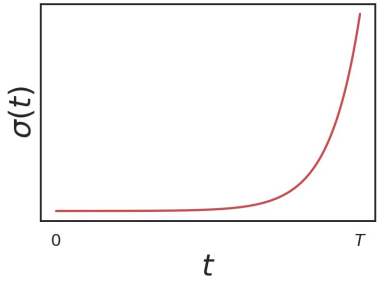
Infinitesimal white noise

Score-based generative modeling via SDEs

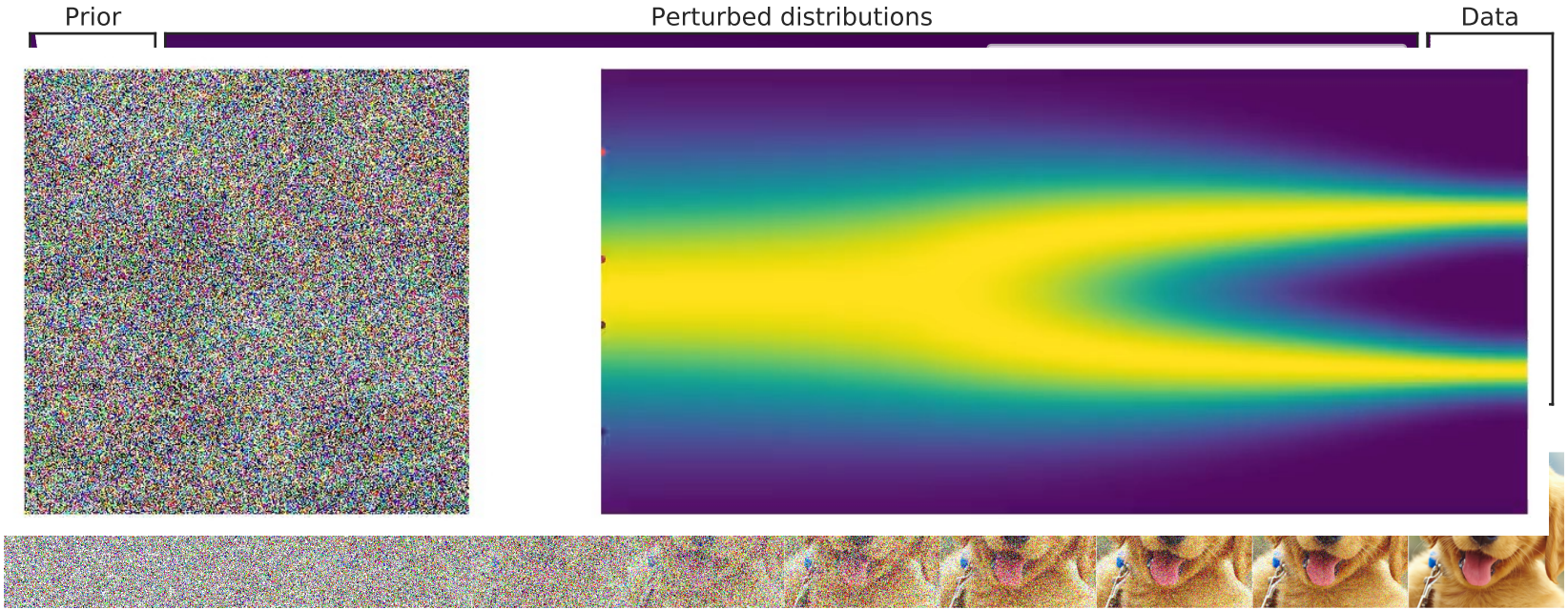
Data



$$\mathrm{d}\mathbf{x} = \sigma(t)\mathrm{d}\mathbf{w}$$



Score-based generative modeling via SDEs



$$\mathrm{d}\mathbf{x} = \sigma(t)\mathrm{d}\mathbf{w}$$

Time reversal

$$\{p_t(\mathbf{x})\}_{t=0}^{T}$$

$$\mathrm{d}\mathbf{x} = -\sigma^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\mathrm{d}t + \sigma(t)\mathrm{d}\bar{\mathbf{w}}$$

Score function!

Score-based generative modeling via SDEs

- Time-dependent score-based model

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

- Training:

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

- Reverse-time SDE

$$\mathrm{d}\mathbf{x} = -\sigma^2(t)\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)\mathrm{d}t + \sigma(t)\mathrm{d}\bar{\mathbf{w}}$$

- Euler-Maruyama (analgous to Euler for ODEs)

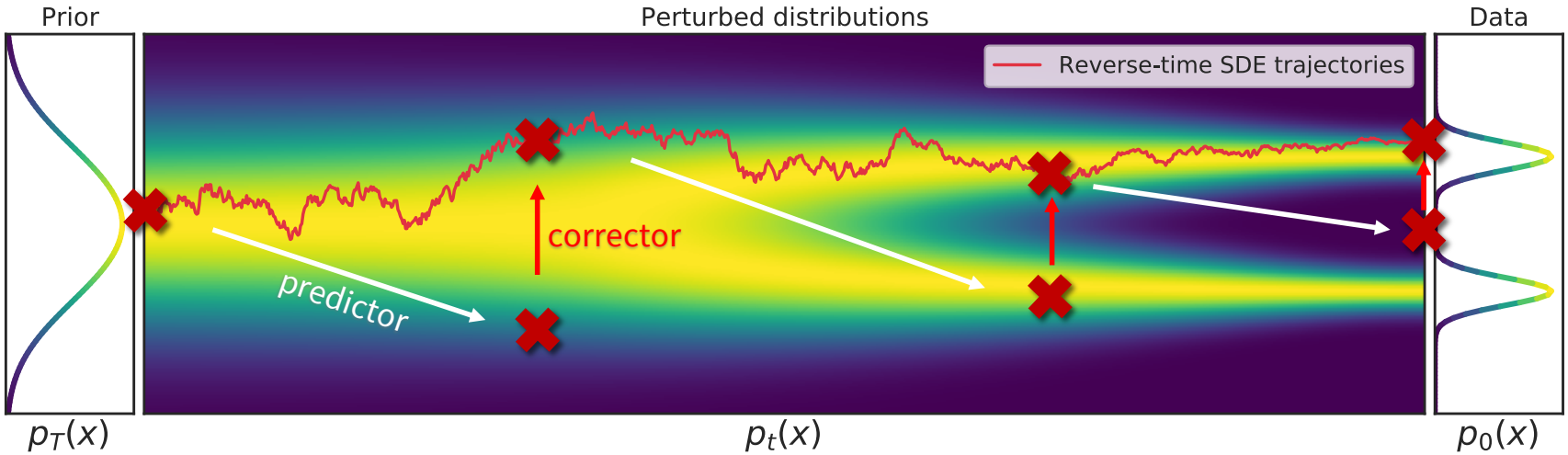
$$\mathbf{x} \leftarrow \mathbf{x} - \sigma(t)^2 \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)\Delta t + \sigma(t)\ \mathbf{z} \quad (\mathbf{z} \sim \mathcal{N}(\mathbf{0}, |\Delta t|\ \mathbf{I}))$$

$$t \leftarrow t + \Delta t$$

Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021.

Predictor-Corrector sampling methods

- Predictor-Corrector sampling.
  - **Predictor:** Numerical SDE solver
  - **Corrector:** Score-based MCMC



Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021.

Results on predictor-corrector sampling

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**

Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. "Score-Based Generative Modeling through Stochastic Differential Equations." ICLR 2021.

High-Fidelity Generation for 1024x1024 Images

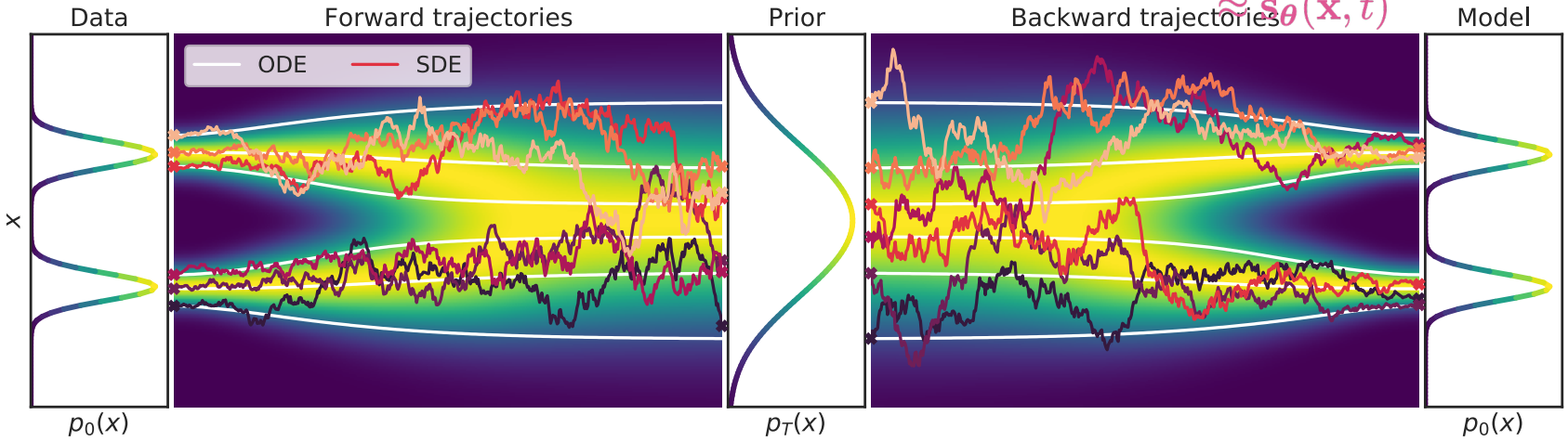
Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021.

Probability flow ODE: turning the SDE to ODE

- Probability flow ODE (ordinary differential equation)

$$\mathrm{d}\mathbf{x} = \sigma(t)\mathrm{d}\mathbf{w} \quad \{p_t(\mathbf{x})\}_{t=0}^{T} \quad \mathrm{d}\mathbf{x} = -\frac{1}{2}\sigma(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \mathrm{d}t$$

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



Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. "Score-Based Generative Modeling through Stochastic Differential Equations." ICLR 2021.

Solving Reverse-ODE for Sampling

- **More efficient samplers** via black-box ODE solvers

NFE = Number of score Function Evaluations

- Predictor-corrector:
  - $>$ 1000 NFE
- ODE
  - $\approx$100 NFE

- **Exact likelihood** though models are trained with score matching.

$$\log p_0(\mathbf{x}) = \log p_T(\mathbf{x}) - \frac{1}{2}\int_0^T \sigma(t)^2 \mathrm{trace}(\nabla_{\mathbf{x}} \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, t))\mathrm{d}t$$

Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021.

Solving Reverse-ODE for Sampling

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**

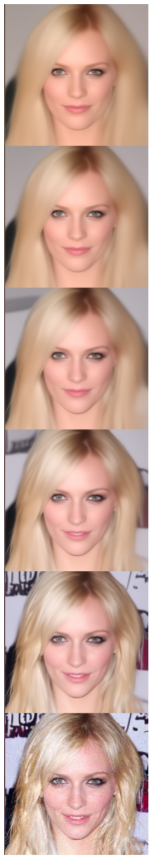
models trained with score matching

black-box ODE Solvers for sampling

Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021.

Probability flow ODE: latent space manipulation

Interpolation

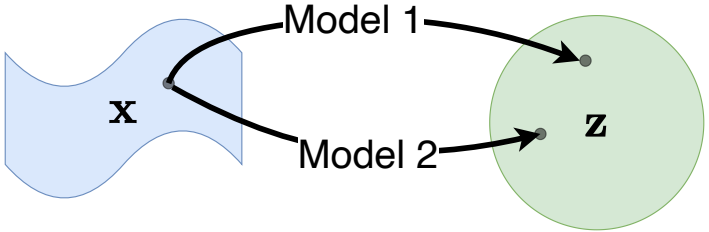


Temperature scaling

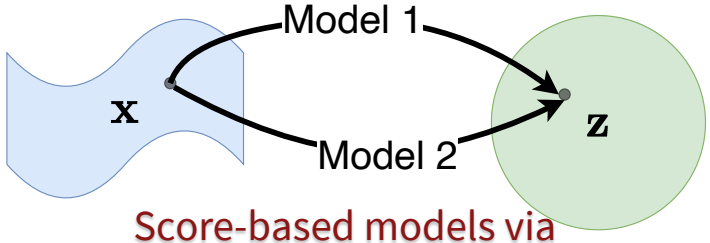
Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021.

Probability flow ODE: uniquely identifiable encoding

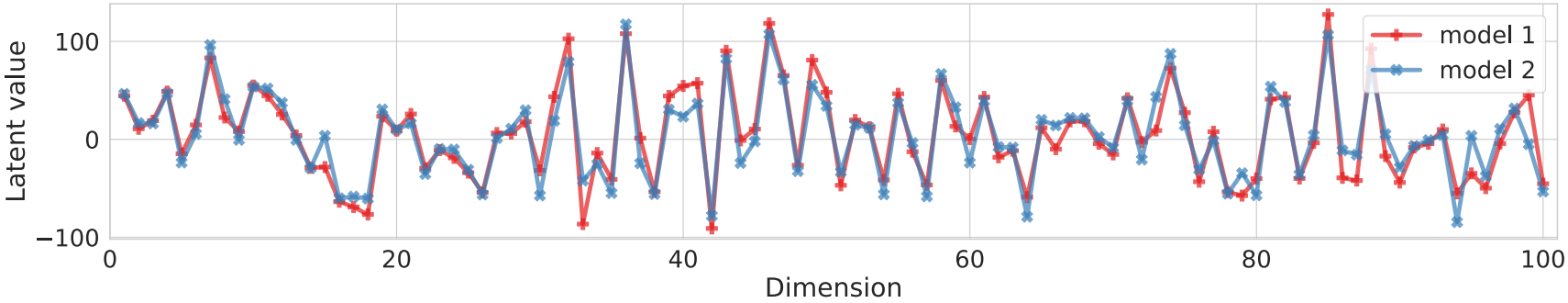
- Uniquely **identifiable** encoding



Flow models, VAE, etc

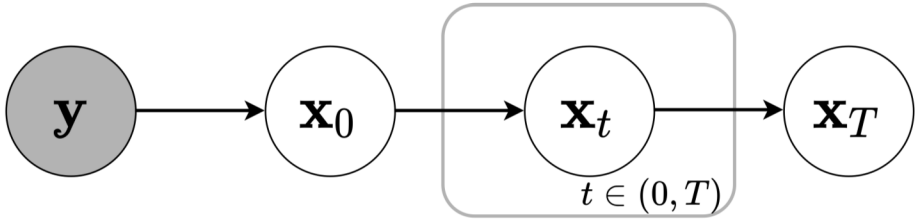


Score-based models via probability flow ODE



Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. "Score-Based Generative Modeling through Stochastic Differential Equations." ICLR 2021.

Controllable Generation



- Conditional reverse-time SDE via unconditional scores

$$\mathrm{d}\mathbf{x} = -\sigma^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x} \mid \mathbf{y})\mathrm{d}t + \sigma(t)\mathrm{d}\bar{\mathbf{w}}$$

$$\mathrm{d}\mathbf{x} = -\sigma^2(t)[\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(\mathbf{y} \mid \mathbf{x})]\mathrm{d}t + \sigma(t)\mathrm{d}\bar{\mathbf{w}}$$

unconditional score, Trained w/o y

Trained separately or specified with domain knowledge

Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. "Score-Based Generative Modeling through Stochastic Differential Equations." ICLR 2021.

Controllable Generation: class-conditional generation

- $\mathbf{y}$ is the **class label**
- $p_t(\mathbf{y} \mid \mathbf{x})$ is a time-dependent classifier

class: bird

class: deer

Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021.

Controllable Generation: inpainting

- $\mathbf{y}$ is the **masked image**
- $p_t(\mathbf{y} \mid \mathbf{x})$ can be approximated without training

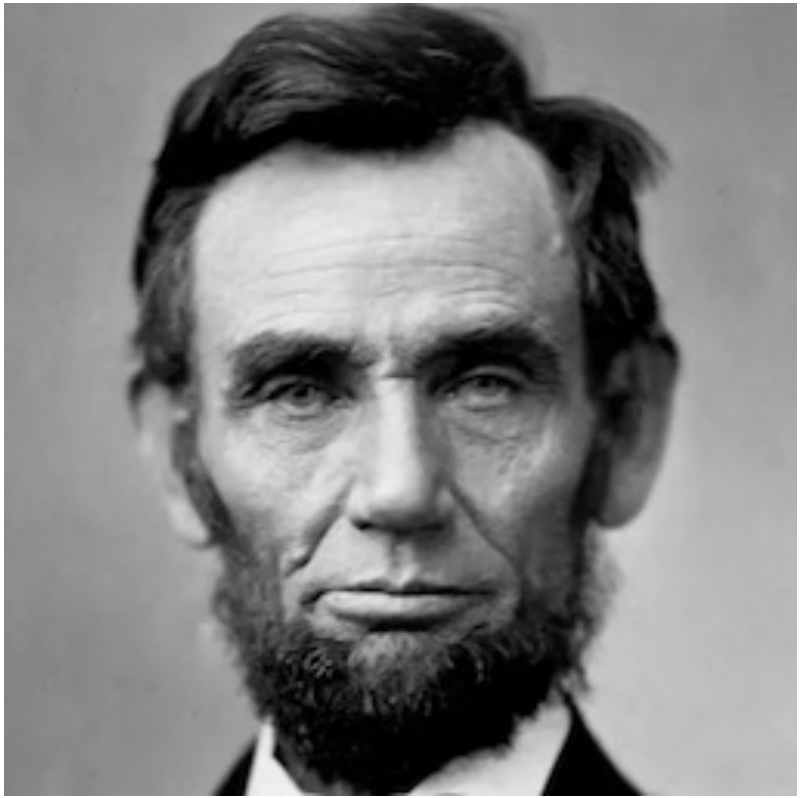
Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021.

Controllable Generation: colorization

- $\mathbf{y}$ is the **gray-scale image**
- $p_t(\mathbf{y} \mid \mathbf{x})$ can be approximated without training

Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. “Score-Based Generative Modeling through Stochastic Differential Equations.” ICLR 2021.

Controllable Generation: colorization



Resolution: 1024x1024

Controllable generation: Text-guided generation

An abstract painting of computer science:

A painting of the starry night by van Gogh

https://colab.research.google.com/drive/1FuOobQOmDJuG7rGsMWfQa883A9r4HxEO?usp=sharing

Conclusion

- **Gradients of distributions (scores) can be estimated easily**
  - **Flexible architecture choices** — no need to be normalized/invertible
  - **Stable training** — no minimax optimization

- **Better or comparable sample quality to GANs**
  - State-of-the-art performance on CIFAR-10 and others
  - Scalable to resolution of 1024x1024 for image generation

- **Exact likelihood computation**
  - State-of-the-art likelihood on CIFAR-10
  - Sample editing via manuplating latent codes
  - Uniquely identifiable encoding