Diffusion Model Study Group #2 Score-based generative modeling

Tanishq Abraham

EleutherAI

9/3/2022

Any questions from last time?

What is score matching?

If the data distribution is $p(\mathbf{x})$, then the score function is defined as

$$\nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Note that if $p(\mathbf{x}) = \frac{e^{-f(\mathbf{x})}}{Z}$ (Z is our normalizing constant that makes density estimation intractable), then:

$$\nabla_{\mathbf{x}} \log p\,(\mathbf{x}) = -\nabla_{\mathbf{x}} f\,(\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log Z}_{=0} = -\nabla_{\mathbf{x}} f\,(\mathbf{x})$$

Don't need Z!

Modeling the score function → score-based model

$$\mathbf{s}_\theta(\mathbf{x}) \approx \log p(\mathbf{x})$$

Trained with the following objective:

$$\mathbb{E}_{p(\mathbf{x})}[\|\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})\|_2^2]$$

Used for training energy-based models

Different types of score matching

Hyvärinen score matching:

$$\mathcal{L}_{matching} = \mathbb{E}_{p(\mathbf{x})}\left[\operatorname{tr}\left(\nabla_x \mathbf{s}_\theta(\mathbf{x})\right) + \frac{1}{2}\|\mathbf{s}_\theta(\mathbf{x})\|_2^2\right]$$

Sliced score matching:

$$\mathcal{L}_{sliced} = \mathbb{E}_{p_{data}}\left[\mathbf{v}^\intercal \nabla_{\mathbf{x}}^2 \log p_\theta\,(\mathbf{x})\mathbf{v} + \frac{1}{2}\left(\mathbf{v}^\intercal \nabla_{\mathbf{x}} \log p_\theta\,(\mathbf{x})\right)^2\right]$$

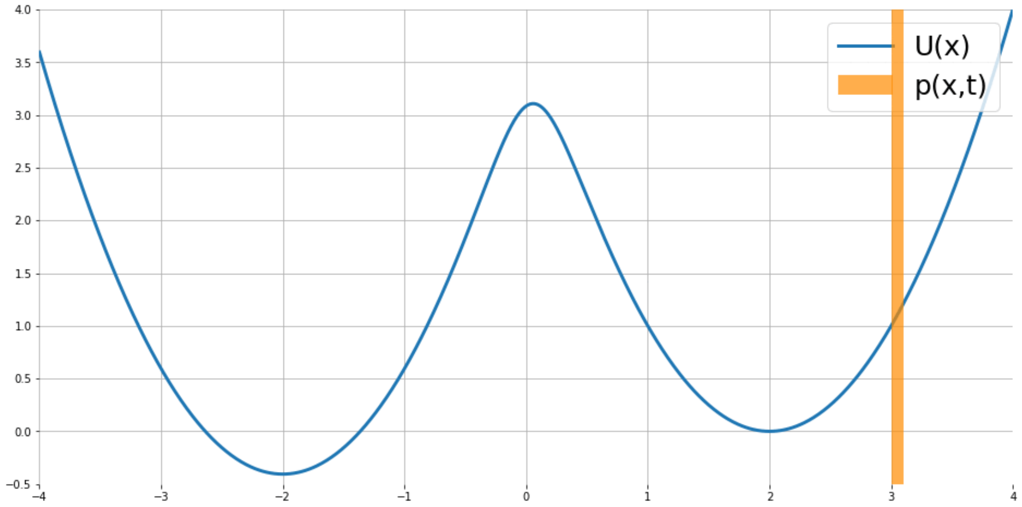
How to sample? – Langevin dynamics!

After sampling from a prior distribution $\mathbf{x_0} \sim \pi(\mathbf{x})$, we iterate as follows:

$$\mathbf{x_{i+1}} \leftarrow \mathbf{x_i} + \epsilon \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \sqrt{2\epsilon}\mathbf{z_i}, \quad i = 0,1,\cdots,K,$$

where $\mathbf{z_i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and ϵ is some step size

As $\epsilon \to 0$ and $K \to \infty$, $\mathbf{x_i}$ is guaranteed to converge to $p(\mathbf{x})$

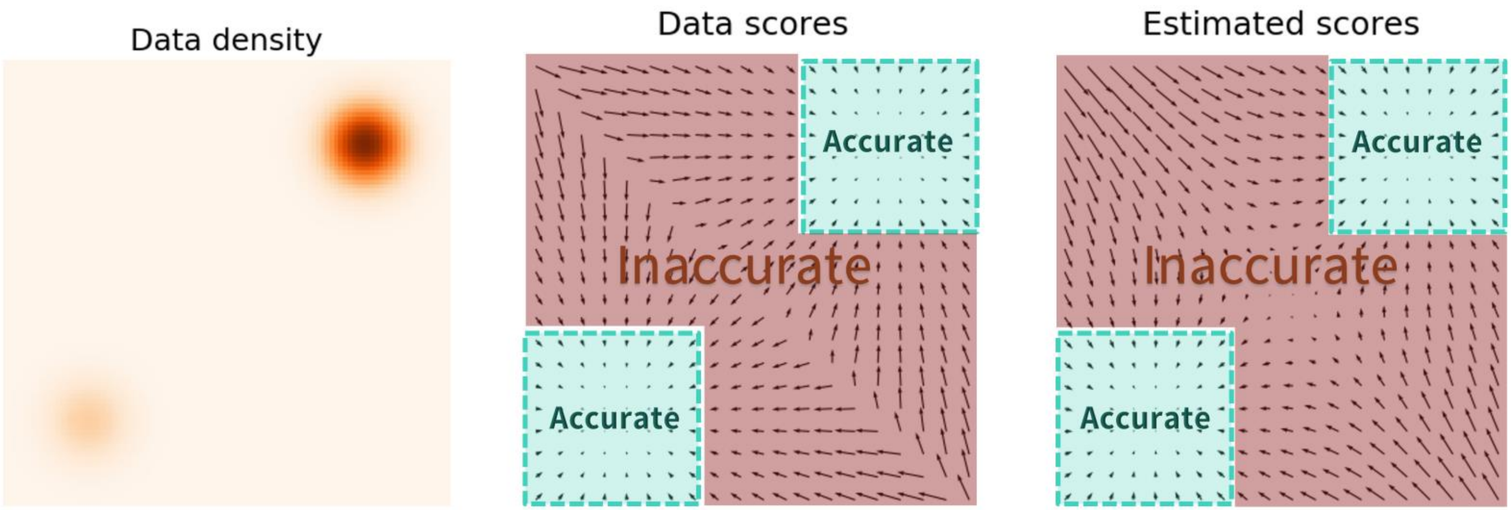


Pitfalls of naïve score matching

The manifold hypothesis – data is embedded in a lower-dimensional manifold while score is over the entire *ambient space*

Inaccurate scores in regions of low data density:

$$\mathbb{E}_{p(\mathbf{x})}[||\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})||_2^2] = \int p(\mathbf{x})\, ||\nabla_{\mathbf{x}} \log p(\mathbf{x}) - \mathbf{s}_\theta(\mathbf{x})||_2^2 \mathrm{d}\mathbf{x}$$



How to solve? – perturb the data!

Noisy data distribution:

$$q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2 \mathbf{I})$$
$$q_\sigma(\tilde{\mathbf{x}}) = \int q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

If σ is small enough:

$$\mathbf{s}_\theta(\mathbf{x}) = \nabla_x \log q_\sigma(\mathbf{x}) \approx \nabla_x \log p(\mathbf{x})$$

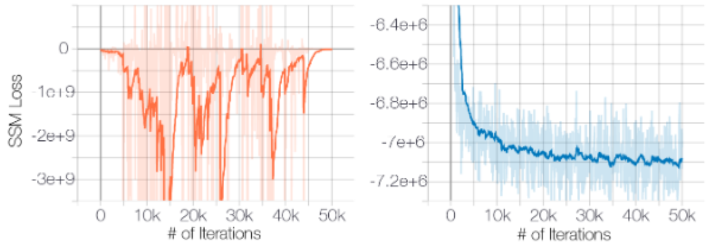
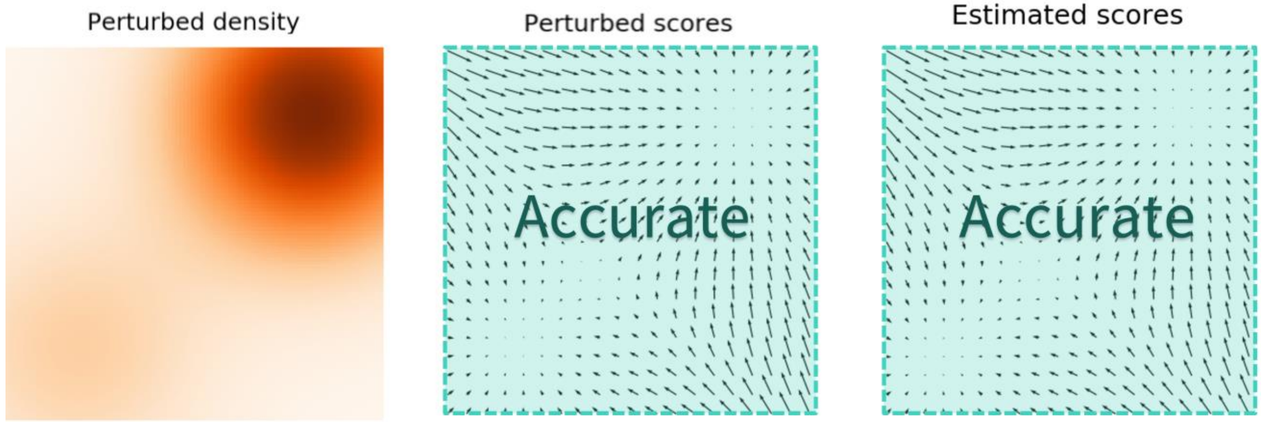


Figure 1: **Left**: Sliced score matching (SSM) loss w.r.t. iterations. No noise is added to data. **Right**: Same but data are perturbed with $\mathcal{N}(0, 0.0001)$.



Denoising score matching

Score matching of the perturbed distribution:

$$\mathcal{L}_{DSM} = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}[\|\nabla_{\mathbf{x}} \log q_\sigma(\tilde{\mathbf{x}}) - \mathbf{s}_\theta(\tilde{\mathbf{x}})\|_2^2]$$

The following objective is equivalent!

$$\mathcal{L}_{DSM} = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}},\mathbf{x})}[\|\nabla_{\mathbf{x}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) - \mathbf{s}_\theta(\tilde{\mathbf{x}})\|_2^2]$$

Since $\log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = -\frac{1}{2\sigma^2}(\tilde{\mathbf{x}} - \mathbf{x})^2$, then $\nabla_{\mathbf{x}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = -\frac{1}{\sigma^2}(\tilde{\mathbf{x}} - \mathbf{x})$

Final objective is:

$$\mathcal{L}_{DSM} = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}},\mathbf{x})}\left[\left\|\frac{1}{\sigma^2}(\tilde{\mathbf{x}} - \mathbf{x}) + \mathbf{s}_\theta(\tilde{\mathbf{x}})\right\|_2^2\right]$$

Tweedie's formula - optimal denoising function $f^*(\tilde{\mathbf{x}}) = \mathbf{x} \approx \tilde{\mathbf{x}} + \sigma^2 \nabla_{\tilde{\mathbf{x}}} \log p(\tilde{\mathbf{x}})$

What’s the right σ?

- Larger noise scale:
  - Pro: better covers lower density regions
  - Con: Significantly different from original distribution
- Smaller noise scale:
  - Pro: Close enough to original distribution
  - Con: Does not cover lower density region

- Can we achieve the best of both worlds?
  - Yes: use multiple σ!

See if this sounds familiar…

Let there be T increasing standard deviations $\sigma_1 < \sigma_2 < \cdots \sigma_t < \cdots < \sigma_T$

Then we have a noisy distribution at each scale:

$$q_{\sigma_t}(\tilde{\mathbf{x}}) = \int p(\mathbf{x})\mathcal{N}(\mathbf{x}, \sigma_t^2 \mathbf{I})\mathrm{d}\mathbf{x}$$

We train a *single* score network conditioned on the noise scale such that $\mathbf{s}_\theta(\tilde{\mathbf{x}}, t) \approx \nabla_\mathbf{x} \log q_{\sigma_t}(\tilde{\mathbf{x}})$

This gives us a new objective:

$$\mathcal{L}_{ncsn} = \sum_{t=1}^{T} \lambda(t) \mathbb{E}_{q_{\sigma_t}(\tilde{\mathbf{x}})}\left[\left\| \nabla_\mathbf{x} \log q_{\sigma_t}(\tilde{\mathbf{x}}) - \mathbf{s}_\theta(\tilde{\mathbf{x}}, t) \right\|_2^2\right]$$

See if this sounds familiar…

Noise-conditional score network (NCSN) objective:

$$\mathcal{L}_{ncsn} = \sum_{t=1}^{T} \lambda(t)\ell(\theta; t)$$

$$\ell(\theta; t) = E_{q_{\sigma_t}(\tilde{\mathbf{x}}, \mathbf{x})}\left[\left\|\frac{1}{\sigma^2}(\tilde{\mathbf{x}} - \mathbf{x}) + \mathbf{s}_\theta(\tilde{\mathbf{x}}, t)\right\|_2^2\right]$$

$\lambda(t) = \sigma_t^2$ for similar magnitude at any loss scale

$\mathbf{s}_\theta(\tilde{\mathbf{x}}, t)$ is a neural network that is conditioned on the timescale

See if this sounds familiar…

Inference process:

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**

 $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$ $\triangleright$ 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 $\tilde{\mathbf{x}}_T$

Experimental setup

Timesteps $T = 10$ for training, $T = 100$ for inference

$\sigma_1 = 0.01$ linearly increases to $\sigma_T = 1$

Model is a modified U-net known as RefineNet, parameters shared across time, timesteps specified via instance normalization

Random flips

Training with Adam

Results

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
CIFAR-10 Conditional		
EBM [12]	8.30	37.9
SNGAN [36]	$8.60 \pm .08$	25.5
BigGAN [6]	**9.22**	**14.73**

Table 1: Inception and FID scores for CIFAR-10

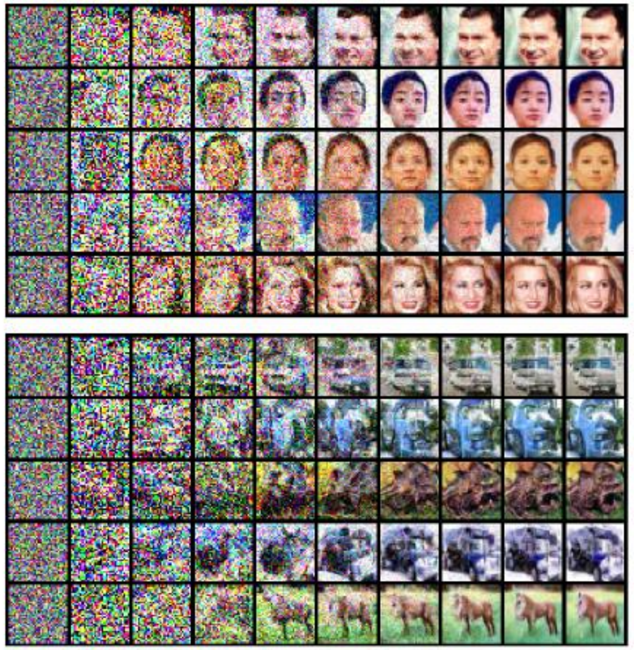


Figure 4: Intermediate samples of annealed Langevin dynamics.

Results

(a) MNIST (b) CelebA (c) CIFAR-10

Direct comparison to DDPMs

Our model architecture, forward process definition, and prior differ from NCSN [55, 56] in subtle but important ways that improve sample quality, and, notably, we directly train our sampler as a latent variable model rather than adding it after training post-hoc. In greater detail:

1. We use a U-Net with self-attention; NCSN uses a RefineNet with dilated convolutions. We condition all layers on t by adding in the Transformer sinusoidal position embedding, rather than only in normalization layers (NCSNv1) or only at the output (v2).
2. Diffusion models scale down the data with each forward process step (by a $\sqrt{1-\beta_t}$ factor) so that variance does not grow when adding noise, thus providing consistently scaled inputs to the neural net reverse process. NCSN omits this scaling factor.
3. Unlike NCSN, our forward process destroys signal ($D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I})) \approx 0$), ensuring a close match between the prior and aggregate posterior of $\mathbf{x}_T$. Also unlike NCSN, our β_t are very small, which ensures that the forward process is reversible by a Markov chain with conditional Gaussians. Both of these factors prevent distribution shift when sampling.
4. Our Langevin-like sampler has coefficients (learning rate, noise scale, etc.) derived rigorously from β_t in the forward process. Thus, our training procedure directly trains our sampler to match the data distribution after T steps: it trains the sampler as a latent variable model using variational inference. In contrast, NCSN's sampler coefficients are set by hand post-hoc, and their training procedure is not guaranteed to directly optimize a quality metric of their sampler.

Appendix

Proof for Hyvärinen score matching

However, Fisher divergence is not directly computable, because the score of the data distribution $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ is unknown. Score matching eliminates the data score using integration by parts. To simplify our discussion, we consider the Fisher divergence between distributions of 1-D random variables. We have

$$
\begin{aligned}
&\frac{1}{2}\mathbb{E}_{p_{\text{data}}}[(\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x))^2] \\
=&\frac{1}{2}\int p_{\text{data}}(x)(\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x))^2 \mathrm{d}x \\
=&\underbrace{\frac{1}{2}\int p_{\text{data}}(x)(\nabla_x \log p_{\text{data}}(x))^2 \mathrm{d}x}_{\text{const}} + \frac{1}{2}\int p_{\text{data}}(x)(\nabla_x \log p_\theta(x))^2 \mathrm{d}x \\
&- \int p_{\text{data}}(x)\nabla_x \log p_\theta(x)\nabla_x \log p_{\text{data}}(x)\mathrm{d}x.
\end{aligned}
$$

By integration by parts, we have

$$
\begin{aligned}
&-\int p_{\text{data}}(x)\nabla_x \log p_\theta(x)\nabla_x \log p_{\text{data}}(x)\mathrm{d}x \\
=&-\int \nabla_x \log p_\theta(x)\nabla_x p_{\text{data}}(x)\mathrm{d}x \\
=&-p_{\text{data}}(x)\nabla_x \log p_\theta(x)\Big|_{-\infty}^{\infty} + \int p_{\text{data}}(x)\nabla_x^2 \log p_\theta(x)\mathrm{d}x \\
\overset{(i)}{=}&\mathbb{E}_{p_{\text{data}}}[\nabla_x^2 \log p_\theta(x)],
\end{aligned}
$$

where (i) holds if we assume $p_{\text{data}}(x) \to 0$ when $|x| \to 0$. Now, substituting the results of integration by parts into the 1-D Fisher divergence, we obtain

$$
\begin{aligned}
&\frac{1}{2}\mathbb{E}_{p_{\text{data}}}[(\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x))^2] \\
=&\mathbb{E}_{p_{\text{data}}}[\nabla_x^2 \log p_\theta(x)] + \frac{1}{2}\mathbb{E}_{p_{\text{data}}}[(\nabla_x \log p_\theta(x))^2] + \text{const}.
\end{aligned}
$$

Proof for denoising score matching

Proof that $J_{ESMq_\sigma} \smile J_{DSMq_\sigma}$ (11)

The explicit score matching criterion using the Parzen density estimator is defined in Eq. 7 as

$$J_{ESMq_\sigma}(\theta) = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta) - \frac{\partial \log q_\sigma(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\right\|^2\right]$$

which we can develop as

$$J_{ESMq_\sigma}(\theta) = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\psi(\tilde{\mathbf{x}};\theta)\|^2\right] - S(\theta) + C_2 \tag{16}$$

where $C_2 = \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\frac{\partial \log q_\sigma(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\right\|^2\right]$ is a constant that does not depend on θ, and

$$\begin{aligned}
S(\theta) &= \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\right\rangle\right] \\
&= \int_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}}) \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}}) \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\frac{\partial}{\partial \tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})}{q_\sigma(\tilde{\mathbf{x}})}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial}{\partial \tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial}{\partial \tilde{\mathbf{x}}} \int_{\mathbf{x}} q_0(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) d\mathbf{x}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \left\langle \psi(\tilde{\mathbf{x}};\theta), \int_{\mathbf{x}} q_0(\mathbf{x}) \frac{\partial q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}} d\mathbf{x}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \left\langle \psi(\tilde{\mathbf{x}};\theta), \int_{\mathbf{x}} q_0(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}} d\mathbf{x}\right\rangle d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \int_{\mathbf{x}} q_0(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle d\mathbf{x} d\tilde{\mathbf{x}} \\
&= \int_{\tilde{\mathbf{x}}} \int_{\mathbf{x}} q_\sigma(\tilde{\mathbf{x}}, \mathbf{x}) \left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle d\mathbf{x} d\tilde{\mathbf{x}} \\
&= \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}},\mathbf{x})}\left[\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle\right].
\end{aligned}$$

Substituting this expression for $S(\theta)$ in Eq. 16 yields

$$\begin{aligned}
J_{ESMq_\sigma}(\theta) &= \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\psi(\tilde{\mathbf{x}};\theta)\|^2\right] \\
&\quad - \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle\right] + C_2.
\end{aligned} \tag{17}$$

12

We also have defined in Eq. 9,

$$J_{DSMq_\sigma}(\theta) = \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\psi(\tilde{\mathbf{x}};\theta) - \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\|^2\right],$$

which we can develop as

$$\begin{aligned}
J_{DSMq_\sigma}(\theta) &= \mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\frac{1}{2}\|\psi(\tilde{\mathbf{x}};\theta)\|^2\right] \\
&\quad - \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\left\langle \psi(\tilde{\mathbf{x}};\theta), \frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\rangle\right] + C_3
\end{aligned} \tag{18}$$

where $C_3 = \mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\frac{1}{2}\left\|\frac{\partial \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}{\partial \tilde{\mathbf{x}}}\right\|^2\right]$ is a constant that does not depend on θ.

Looking at equations 17 and 18 we see that $J_{ESMq_\sigma}(\theta) = J_{DSMq_\sigma}(\theta) + C_2 - C_3$. We have thus shown that the two optimization objectives are equivalent.