Diffusion models for particle physics simulations

Arjun Sharma

May 15, 2025

1. **Particle physics simulations**

2. **Diffusion Models**

3. **Equivariant learning**

4. **Appendices**

Particle physics simulations

Collisions

$$\text{High energy protons} \xrightarrow{\text{collisions}} \text{Non-isolatable elementary particles} \xrightarrow{\text{radiate}} \text{Parton shower} \xrightarrow[\text{hadronization}]{\text{cooling}} \text{Jet}$$

$$\text{Initial particles} \to \text{Jet constituents}$$

Simulation-based hypothesis testing

$$p(\beta|o) = \frac{p(o|\beta)p(\beta)}{\int p(\beta')p(o|\beta')\,d\beta'}$$

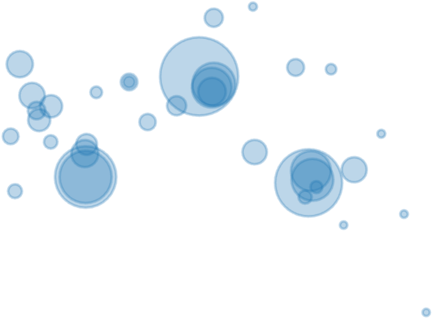
- Input: z_0, β
- Latent states: $z_i \sim p_i(z_i|\beta, z_{<i})$
- Output: $o \sim p(o|\beta, z)$

$$p(o|\beta) = \int p(o, \beta|\beta)dz$$

Limitations

- Fit simplified phenomenological models to parton shower, hadronization
- Slow: 10 s / jet
- One-shot generation: 3-5 OoM speedup!

Representing jets



Gouskos and Qu 2018

Particle clouds + feature vectors

Diffusion Models

Generative modeling

Given: $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N$ from $p_{\text{data}}(\mathbf{x})$

Learn f_θ s.t.

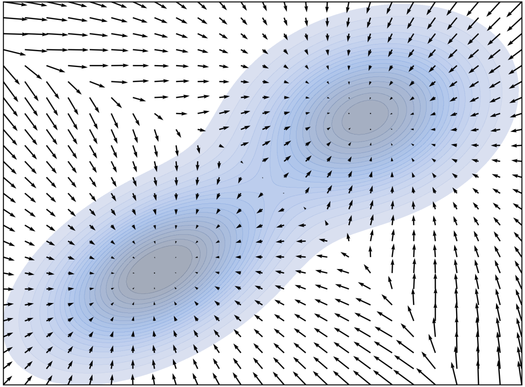
$$p_\theta(\mathbf{x}) \quad = \frac{e^{f_\theta(\mathbf{x})}}{Z_\theta}$$

$$Z_\theta = \int e^{f_\theta(\mathbf{x})}\, d\mathbf{x}$$

Score functions

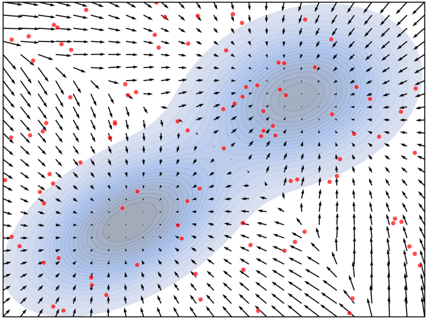
$$p_\theta(\mathbf{x}) = \frac{e^{f_\theta(\mathbf{x})}}{Z_\theta}$$

$$\begin{aligned}\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) &= \nabla_{\mathbf{x}} f_\theta(\mathbf{x}) - \nabla_{\mathbf{x}} \log Z_\theta \\ &= \nabla_{\mathbf{x}} f_\theta(\mathbf{x}) = s_\theta(\mathbf{x})\end{aligned}$$



Sampling

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha \cdot s_\theta(\mathbf{x}_t)$$
$$x_0 \sim \pi_{\text{init}}$$
$$t = 0, 1, \ldots T \to \infty$$



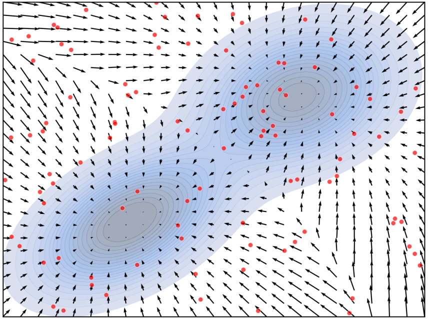
Sampling with Langevin dynamics

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha \cdot s_\theta(\mathbf{x}_t) + \alpha\epsilon \cdot \mathbf{z}_t$$

$$x_0 \sim \pi_{\text{init}}$$

$$\mathbf{z}_t \sim \mathcal{N}(0, 1)$$

$$t = 0, 1, \ldots T \to \infty$$



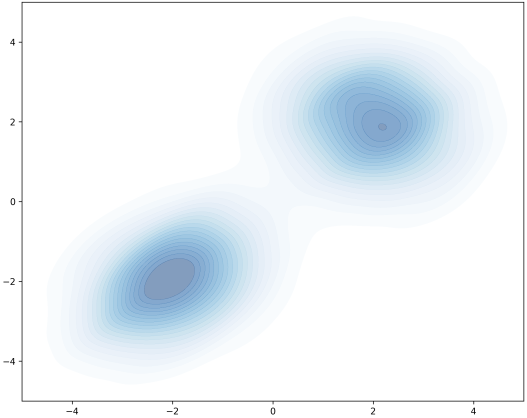
Training objective for score

$$\mathcal{L}_{\mathsf{ESM}}(\theta) = \frac{1}{2}\mathbb{E}_{p_{\mathsf{data}}(\mathbf{x})}\left[||\nabla_{\mathbf{x}} \log p_{\mathsf{data}}(\mathbf{x}) - s_\theta(\mathbf{x})||_2^2\right]$$

$$\iff \mathcal{L}_{\mathsf{ISM}}(\theta) = \frac{1}{2}\mathbb{E}_{p_{\mathsf{data}}(\mathbf{x})}\left[s_\theta(\mathbf{x})^2\right] + \mathbb{E}_{p_{\mathsf{data}}(\mathbf{x})}\left[\nabla_{\mathbf{x}} s_\theta(\mathbf{x})\right]$$

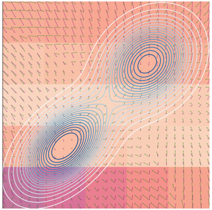
$$\approx \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{2}s_\theta(\mathbf{x}_i)^2 + \nabla_{\mathbf{x}_i} s_\theta(\mathbf{x}_i)\right]$$

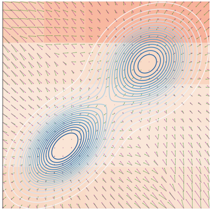
Training



Drawbacks

Field in low-density regions





Scaling

$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[s_\theta(\mathbf{x})^2\right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\nabla_{\mathbf{x}} s_\theta(\mathbf{x})\right]$$

$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[s_\theta(\mathbf{x})^2\right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\text{trace}(\nabla_{\mathbf{x}} s_\theta(\mathbf{x}))\right]$$

Forward propagation

Backprops $\propto$ dim$(\mathbf{x})$

Noise perturbation

$$
\begin{aligned}
\tilde{\mathbf{x}} &= \mathbf{x} + z \\
z &\sim \mathcal{N}(0, \sigma^2 I) \\
q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) &= \mathcal{N}(\mathbf{x}, \sigma^2 I)
\end{aligned}
$$

Denoising objective

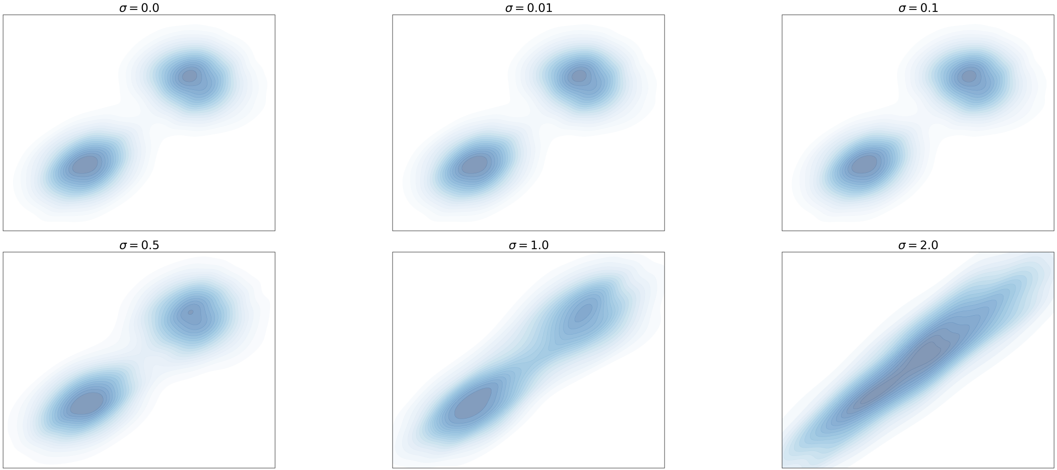
$$\mathcal{L}_{\mathsf{ESM}\sigma}(\theta, \sigma) = \frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[||s_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})||_2^2\right]$$

$$\iff \mathcal{L}_{\mathsf{ISM}\sigma}(\theta, \sigma) = \frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}, \mathbf{x})}\left[||s_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})||_2^2\right]$$

Gradient of the corrupted distribution

$$
\begin{aligned}
\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) &= \nabla_{\tilde{\mathbf{x}}} \log \frac{1}{(2\pi)^{\frac{d}{2}}\sigma^d} e^{-\frac{1}{2\sigma^2}||\tilde{\mathbf{x}}-\mathbf{x}||^2} \\
&= \nabla_{\tilde{\mathbf{x}}} \log e^{-\frac{1}{2\sigma^2}||\tilde{\mathbf{x}}-\mathbf{x}||^2} - \nabla_{\tilde{\mathbf{x}}} \log (2\pi)^{\frac{d}{2}}\sigma^d \\
&= -\nabla_{\tilde{\mathbf{x}}} \frac{1}{2\sigma^2}||\tilde{\mathbf{x}}-\mathbf{x}||^2 \\
&= \frac{||\tilde{\mathbf{x}}-\mathbf{x}||}{\sigma^2} \\
\therefore \mathcal{L}_{\mathsf{ISM}\sigma}(\theta,\sigma) &= \frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}},\mathbf{x})}\left[||s_\theta(\tilde{\mathbf{x}}) - \frac{||\tilde{\mathbf{x}}-\mathbf{x}||}{\sigma^2}||_2^2\right]
\end{aligned}
$$

Multiple noise levels



Multiple noise levels

$$\mathcal{L}(\theta, \{\sigma_i\}_{i=1}^{L}) = \frac{1}{L}\sum_{i=1}^{L} \lambda(\sigma_i)\mathbb{E}_{q_{\sigma_i}(\tilde{\mathbf{x}},\mathbf{x})}\left[||s_\theta(\tilde{\mathbf{x}}, \sigma_i) - \frac{||\tilde{\mathbf{x}} - \mathbf{x}||}{\sigma^2}||_2^2 \right]$$

Sampling with annealed Langevin dynamics

$$\tilde{\mathbf{x}}_0 \sim \pi_{\mathsf{init}}$$

From $i = 1, 2, \ldots L$ with $\frac{\sigma_1}{\sigma_2} = \cdots = \frac{\sigma_{L-1}}{\sigma_L} > 1$

$$\begin{aligned}
\alpha_i &= \gamma \frac{\sigma_i^2}{\sigma_L^2} \\
\tilde{\mathbf{x}}_t &= \tilde{\mathbf{x}}_{t-1} + \alpha_i s_\theta(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \epsilon \alpha_i \mathbf{z}_t \\
\mathbf{z}_t &\sim \mathcal{N}(0, 1)
\end{aligned}$$

Conditional generation

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}$$

$$\begin{aligned}
\nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) &= \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{x}) - \nabla_{\mathbf{x}} \log p(\mathbf{y}) \\
&= \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) \\
&= s_\theta(\mathbf{x}) + \gamma \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x})
\end{aligned}$$

Single model for guidance

$$
\begin{aligned}
p(\mathbf{y}|\mathbf{x}) &= \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) - \nabla_{\mathbf{x}} \log p(\mathbf{x}) \\
\implies \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) &= \nabla_{\mathbf{x}} \log p(\mathbf{x}) + \gamma \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) - \gamma \nabla_{\mathbf{x}} \log p(\mathbf{x}) \\
&= \gamma \nabla_{\mathbf{x}} \log p(\mathbf{x}|\mathbf{y}) + (\gamma - 1) \nabla_{\mathbf{x}} \log p(\mathbf{x}) \\
&\approx \gamma s_\theta(\mathbf{x}, \mathbf{y}) - (\gamma - 1) s_\theta(\mathbf{x}, \phi)
\end{aligned}
$$

Final score model

$$s_theta(\mathbf{x}, \mathbf{y}, \beta)$$

Equivariant learning

- Equivariant: $f(g(I)) = g'(f(I))$, $g \in G, g' \in G'$
- Same-equivariant: $f(g(I)) = g(f(I))$
- Invariant: $f(g(I)) = f(I)$

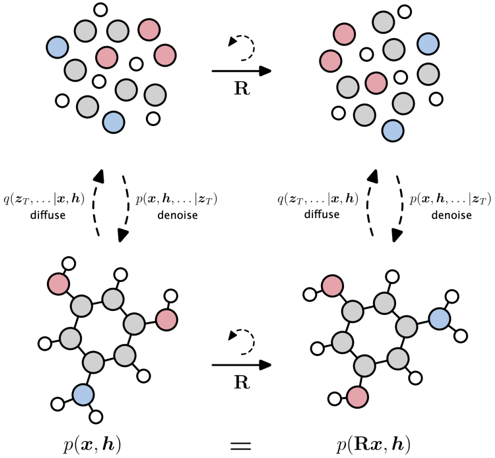
- Images: translation, rotation, reflection
- Graphs, sets: permutation

Inductive biases

In ML

1. Data augmentation
2. Model architecture

Equivariant diffusion models



Hoogeboom et al. 2022

Lorentz equivariance in diffusion?

$$\text{eg.} \begin{pmatrix} \gamma & -\gamma\beta & 0 & 0 \\ -\gamma\beta & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(relative velocity $\beta = \frac{v}{c}$, boost factor $\gamma = \frac{1}{(1-\beta^2)^2}$)

Appendices

Derivation of divergence objective for score

$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[||\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - s_\theta(\mathbf{x})||_2^2\right]$$

$$= \frac{1}{2}\int p_{\text{data}}(\mathbf{x})\left(\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} f_\theta(\mathbf{x})\right)^2\, d\mathbf{x}$$

$$= \frac{1}{2}\int p_{\text{data}}(\mathbf{x})\left(\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})\right)^2\, d\mathbf{x} + \frac{1}{2}\int p_{\text{data}}(\mathbf{x})\left(\nabla_{\mathbf{x}} f_\theta(\mathbf{x})\right)^2\, d\mathbf{x} - \int p_{\text{data}}(\mathbf{x})\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) \cdot \nabla_{\mathbf{x}}$$

Constant in training

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

$$= C + \frac{1}{2}\int p_{\text{data}}(\mathbf{x})\left(\nabla_{\mathbf{x}} f_\theta(\mathbf{x})\right)^2\, d\mathbf{x} - \int \nabla_{\mathbf{x}} p_{\text{data}}(\mathbf{x}) \cdot \nabla_{\mathbf{x}} f_\theta(\mathbf{x})\, d\mathbf{x}$$

Integrate by parts

$$= C + \frac{1}{2}\int p_{\text{data}}(\mathbf{x})\left(\nabla_{\mathbf{x}} f_\theta(\mathbf{x})\right)^2\, d\mathbf{x} - p_{\text{data}}(\mathbf{x})\nabla_x f_\theta(\mathbf{x})|_{-\infty}^{\infty} + \int p_{\text{data}}(\mathbf{x})\nabla_{\mathbf{x}}\nabla_{\mathbf{x}} f_\theta(\mathbf{x}) d\mathbf{x}$$

Boundary values $\rightarrow$ 0

$$= C + \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\nabla_{\mathbf{x}} f_\theta(\mathbf{x})^2\right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\nabla_{\mathbf{x}}^2 f_\theta(\mathbf{x})\right] = C + \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[s_\theta(\mathbf{x})^2\right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\nabla_{\mathbf{x}} s_\theta(\mathbf{x})\right]$$

Derivation of noise-perturbed divergence objective

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[||s_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})||_2^2\right]$$

$$= \frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[s_\theta(\tilde{\mathbf{x}})^2\right] - \underbrace{\int q_\sigma(\tilde{\mathbf{x}}) s_\theta(\tilde{\mathbf{x}}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})\, d\tilde{\mathbf{x}}}_{\text{Use } \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) = \frac{\nabla_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})}{q_\sigma(\tilde{\mathbf{x}})}} + \underbrace{\frac{1}{2}\int q_\sigma(\tilde{\mathbf{x}})(\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}))^2\, d\tilde{\mathbf{x}}}_{\text{Constant in training}}$$

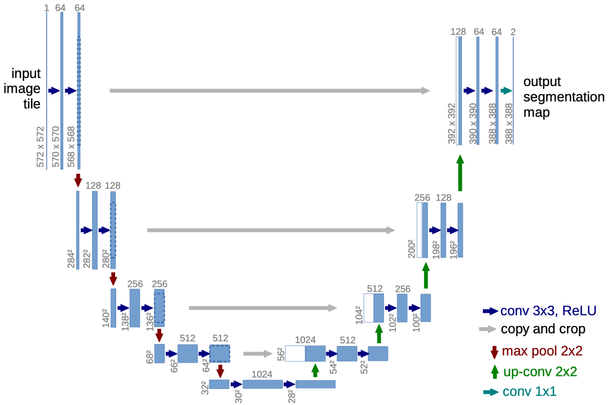
$$\underbrace{\int s_\theta(\tilde{\mathbf{x}}) \nabla_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}})\, d\tilde{\mathbf{x}}}_{\text{Use } q_\sigma(\tilde{\mathbf{x}}) = \int p_{\text{data}}(x) p_\sigma(\tilde{\mathbf{x}}|bx)}$$

$$= \int s_\theta(\tilde{\mathbf{x}}) \underbrace{\nabla_{\tilde{\mathbf{x}}}\left(\int p_{\text{data}}(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\, d\mathbf{x}\right)}_{\text{Differentiation under integral}} d\tilde{\mathbf{x}}$$

$$= \int s_\theta(\tilde{\mathbf{x}}) \left(\int p_{\text{data}}(\mathbf{x}) \underbrace{\nabla_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}_{\text{Use } \nabla_{\tilde{\mathbf{x}}} q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\, d\mathbf{x}\right) d\tilde{\mathbf{x}} = \int\int s_\theta(\tilde{\mathbf{x}}) p_{\text{data}}(\mathbf{x}) q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\, d\mathbf{x}\, d\tilde{\mathbf{x}}$$

$$= \mathbf{E}_{q_\sigma(\tilde{\mathbf{x}},\mathbf{x})}\left[s_\theta(\tilde{\mathbf{x}}) \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\right]$$

U-Net



Source: Ronneberger et al. (2015)