Constrained Diffusion Models

Yuanhao JIANG

July 7, 2025

Abstract

Abstract

Contents

Contents **i**

1 paperworks **1**

1.1 Data 1

1.2 Model Architecture 1

1.3 Score-Based Diffusion Models 1

1.3.1 Forward Process 1

1.3.2 Denoising Score Matching 2

1.3.3 Reverse Process 2

1.4 Reimannian Score-Based Diffusion Models 3

1.4.1 Geodesic Random Walk 3

1.4.2 Constrained Symplectic Euler-Like Method 4

Bibliography **6**

Chapter 1

paperworks

1.1 Data

CATH S40 domain structures, it contains non-redundant data with no pair of domains with $\geq 40\%$ sequence similarity (according to BLAST).

Extract the $C\alpha$ (alpha carbon) atoms for each protein domain in the dataset, so each data is a 3D point cloud of $C\alpha$ atoms

$$x = [x_1, x_2, ..., x_l], \quad x_i \in \mathbb{R}^3 \quad \forall i = 1, ..., l,$$

where l the number of $C\alpha$ atoms (residues) is varying for different domain.

1.2 Model Architecture

For variable-length, 3D point cloud data, I use Euclidean neural networks [Thomas et al., 2018, Weiler et al., 2018, Kondor et al., 2018]. It is rotation and translation invariant for 3D point cloud, and all its operations are per-node, or per-edge based, so it is compatible with variable-sized structures.

1.3 Score-Based Diffusion Models

1.3.1 Forward Process

Use Variance Preserving (VP) SDE [Song et al., 2020]

$$dx = -\frac{1}{2}\beta(t)x + \sqrt{\beta(t)}dw \tag{1.1}$$

with closed form posterior

$$p\left(x_t|x_0\right) = \mathcal{N}\left(x_t :\ x_0 e^{-\frac{1}{2}\int_0^t \beta(s)ds}, \left(1 - e^{-\frac{1}{2}\int_0^t \beta(s)ds}\right) I\right)$$

for data perturbation.

1.3.2 Denoising Score Matching

Ideally, for each time t, the score function of x_t can be trained via

$$\mathbb{E}_{p(x_t)}\left[\left\|\nabla_{x_t}\log p\left(x_t\right)-s_\theta\left(x_t,t\right)\right\|_2^2\right].$$

The unknown term $\nabla_{x_t}\log p\left(x_t\right)$ (true score) can be eliminated with the score matching objective [Hyvärinen, 2005]

$$J_t^{\mathrm{SM}}(\theta)=\mathbb{E}_{p(x_t)}\left[\left\|\operatorname{tr}\left(\nabla_{x_t}s_\theta\left(x_t,t\right)\right)-\frac{1}{2}\|s_\theta(x_t,t)\|_2^2\right\|_2^2\right].$$

Note that the term $\operatorname{tr}\left(\nabla_{x_t}s_\theta\left(x_t,t\right)\right)$ can be computationally intensive, since we have the analytic form of the posterior, this can be eased by using instead the denoising score matching objective [Vincent, 2011]

$$J_t^{\mathrm{DSM}}=\mathbb{E}_{p(x_0)}\mathbb{E}_{p(x_t|x_0)}\left[\left\|s_\theta\left(x_t,t\right)-\nabla_{x_t}\log p\left(x_t|x_0\right)\right\|_2^2\right].$$

The overall objective is given by a weighted sum (or average)

$$J^{\mathrm{DSM}}=\mathbb{E}_{t\sim\mathcal{U}(0,1)}\left[\lambda\left(t\right)J_t^{\mathrm{DSM}}\right].$$

1.3.3 Reverse Process

For an SDE

$$dx=f(x,t)dt+g(t)dw,$$

the reverse time SDE is given by [Anderson, 1982]

$$dx=\left[f(x,t)-g^2(t)\nabla_x\log p\left(x_t\right)\right]dt+g\left(t\right)d\tilde{w},$$

where t goes from 1 to 0. So for VPSDE (1.1) its reverse process is

$$dx=\left[-\frac{1}{2}\beta(t)-\beta^2(t)\nabla_x\log p\left(x_t\right)\right]dt+\sqrt{\beta(t)}d\tilde{w},$$

and the true score is approximated by the trained neural network s_θ. For discretisation the Predictor-Corrector (PC) sampler [Song et al., 2020] is used, i.e., the predictor step steps into next time step in the reverse SDE

$$x_{k+1}\leftarrow x_k+\left[f(x,t)-g^2(t)s_\theta\left(x_k,t\right)\right](-\delta t)+g\left(t\right)\sqrt{\delta t}z_{k+1},$$

and the corrector step corrects the sampling using MCMC method in a fixed time t

$$x_{k+1}\leftarrow x_{k+1}+\epsilon_{k+1}s_\theta\left(x_{k+1},t\right)+\sqrt{2}z'_{k+1},$$

for white noises z_{k+1} and z'_{k+1}.

1.4 Reimannian Score-Based Diffusion Models

The constraints I use are

1. fixed bond lengths ($C\alpha$-$C\alpha$ distances),

$$\|x_{i+1} - x_i\| = \text{const};$$

2. fixed bond angles,

$$\angle\left(x_{i-1}, x_i, x_{i+1}\right) = \text{const};$$

3. ....

Suppose, for an $x \in \mathbb{R}^{l\times 3}$ on the manifold, its constraints are expressed as

$$h_i(x) = 0, \quad i = 1, ..., m,$$

the Jacobian matrix of the constraints is given by

$$J(x) = \begin{bmatrix} \nabla_x^T h_1(x) \\ \nabla_x^T h_2(x) \\ \vdots \\ \nabla_x^T h_m(x) \end{bmatrix} \in \mathbb{R}^{m\times 3l}.$$

Since now we have no closed form solution to the SDE, a discretisation rule is need to perturb the data.

1.4.1 Geodesic Random Walk

The projector of the tangent space of x is given by

$$\Pi(x) = I - J^T(x)(J(x)J^T(x))^{-1}J(x),$$

so the SDE on manifold is given by

$$dx = \Pi(x)f(x,t)dt + \Pi(x)g(t)dw. \tag{1.2}$$

To sample from it, we follows the geodesic random walk in [De Bortoli et al., 2022]. At time t, first sample on tangent space of x_t

$$v_{k+1} = \Pi\left(x_t\right)\left(f(x_t,t)\delta t + g(t)\sqrt{\delta t}z_{k+1}\right),$$

then move along the geodesic on manifold $\mathcal{M}$ defined by the constraints, i.e.,

$$x_{k+1} \leftarrow \exp_{x_k}\left(v_{k+1}\right).$$

The exponential map $\exp_x(v)$ is known for sphere and torus in [De Bortoli et al., 2022], however, for manifold defined by our constraints, it is unkonwn. It can be approximated by the solution of the optimisation problem

$$\widehat{\exp_x}(v) = \arg\min_{x'} \frac{1}{2}\|x' - (x+v)\|^2 \quad \text{subject to} \quad h(x') = 0.$$

Reformulate to obtain

$$\arg\min_{\delta x} \frac{1}{2}\|x + \delta x - (x+v)\|^2 \quad \text{subject to} \quad h(x+\delta x) = 0.$$

Expand the constraint and only keep the first order terms $h(x+\delta x) = h(x) + J(x)\delta x$, and Lagrangian is

$$\mathcal{L}(\delta x, \lambda) = \frac{1}{2}\|\delta x - v\|^2 + \lambda^T (h(x) + J(x)\delta x).$$

Now we can solve with KKT conditions as the problem is convex, i.e.,

$$\begin{aligned} \nabla_{\delta x}\mathcal{L}(\delta x, \lambda) &= 0 \\ h(x) + J(x)\delta x &= 0 \end{aligned}$$

This yields the optimal update

$$\delta x = v - J^T(x)\lambda$$

where λ solves

$$J(x)J^T(x)\lambda = h(x) + J(x)v = J(x)v. \qquad \text{(since we start on manifold.)}$$

In the end the exponential map is

$$\widehat{\exp_x}(v) = x + \delta x.$$

1.4.2 Constrained Symplectic Euler-Like Method

This is an alternative to geodesic random walk that avoid approximating the exponential map. Instead of trying to sample from (1.2), we use the equivalent

$$\begin{aligned} dx &= f(x,t)dt + g(t)dw - J^T(x)\lambda(t)dt, \\ h(x) &= 0. \end{aligned}$$

We use the constrained Symplectic Euler-like method [Leimkuhler and Matthews, 2015]

$$x_{k+1} = x_k + f(x_k, t_k)\delta t + g(t_k)\sqrt{\delta t} z_{k+1} - J^T(x_k)\lambda_k \delta t$$

where λ_k is solved by the constraints

$$h(x_{k+1}) = h\left(x_k + f(x_k, t_k)\delta t + g(t_k)\sqrt{\delta t} z_{k+1} - J^T(x_k)\lambda_k \delta t\right) = 0.$$

Let $Q_k = x_k + f(x_k, t_k)\delta t + g(t_k)\sqrt{\delta t} z_{k+1}$ and $\Lambda_k = \lambda_t \delta t$ we have

$$h\left(Q_k + J^T(x_k)\Lambda_k\right) = 0,$$

which can be solved by Gauss-Newton iteration

$$\begin{aligned} Q^\ell &\coloneqq Q_k - J^T\left(x_k\right)\Lambda^\ell \\ \Lambda_k^{\ell+1} &= \Lambda_k^\ell + \left[J\left(Q^\ell\right)J^T(x_k)\right]^{-1} h\left(Q^\ell\right). \end{aligned}$$

Bibliography

[Anderson, 1982] Anderson, B. D. O. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326.

[De Bortoli et al., 2022] De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. (2022). Riemannian score-based generative modelling. *Advances in Neural Information Processing Systems*, 35:2406–2422.

[Hyvärinen, 2005] Hyvärinen, A. (2005). Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, 6(24):695–709.

[Kondor et al., 2018] Kondor, R., Lin, Z., and Trivedi, S. (2018). Clebsch-Gordan Nets: A Fully Fourier Space Spherical Convolutional Neural Network.

[Leimkuhler and Matthews, 2015] Leimkuhler, B. and Matthews, C. (2015). *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*, volume 39 of *Interdisciplinary Applied Mathematics*. Springer International Publishing, Cham.

[Song et al., 2020] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

[Thomas et al., 2018] Thomas, N., Smidt, T., Kearnes, S., Yang, L., Li, L., Kohlhoff, K., and Riley, P. (2018). Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds.

[Vincent, 2011] Vincent, P. (2011). A Connection Between Score Matching and Denoising Autoencoders. *Neural Computation*, 23(7):1661–1674.

[Weiler et al., 2018] Weiler, M., Geiger, M., Welling, M., Boomsma, W., and Cohen, T. (2018). 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data.