University of Oxford

Here is a very long title which should spread over two or more lines.

by

Yuanhao Jiang

Wolfson College

A dissertation submitted in partial fulfilment of the degree of Master of Science in Applied Statistics.

Department of Statistics, 24–29 St Giles,
Oxford, OX1 3LB

September 2025

This is my own work (except where otherwise indicated)

Candidate: Full Name

Signed:.................................

Date:...................................

Abstract

The abstract should go here.

Acknowledgements

I would like to thank the following:

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background on Diffusion Models

Diffusion models have recently emerged as one of the most powerful classes of deep generative models, achieving state-of-the-art performance in a wide range of domains, most prominently in image generation Ho et al. (2020); Dhariwal and Nichol (2021). These models construct a forward process that gradually corrupts data with Gaussian noise and train a neural network to approximate the reverse-time dynamics that iteratively remove noise. The resulting generative process is both probabilistically principled and empirically effective, offering advantages in sample quality and training stability compared to earlier paradigms such as variational autoencoders (VAEs) Kingma and Welling (2022) and generative adversarial networks (GANs) Goodfellow et al. (2020).

Score-based diffusion models Song et al. (2020, 2021) improve discrete-time denoising diffusion probabilistic models with continuous-time stochastic differential equations (SDEs). Instead of directly parameterising the reverse process, they learn the score function—the gradient of the log-density of the noisy distribution—which enables the use of a variety of forward SDEs and corresponding reverse-time samplers. This framework has proven highly flexible and has been successfully applied to domains including image generation, audio synthesis, point clouds, and molecular data.

1.2 Motivation: Protein Structure Generation

Proteins are essential macromolecules whose biological functions are determined by their three-dimensional structures. The ability to generate novel protein structures has immense implications for biomedical science, including de novo protein design, drug discovery, and enzyme engineering. However, protein structures are highly constrained: bond lengths and angles must remain within narrow physical ranges, torsional angles follow characteristic distributions, and valid folds must satisfy global topological requirements. These constraints make the generative task significantly more challenging than in image or text domains.

Recent advances in deep learning, most notably AlphaFold Jumper et al. (2021), have demonstrated the power of data-driven models in predicting native structures from sequence. Yet, the problem of generating realistic, diverse backbones without explicit sequence conditioning remains an open challenge.

1.3 Diffusion Models for Protein Structures

The probabilistic formulation of score-based diffusion models makes them a natural candidate for modelling protein structures. Unlike GANs, diffusion models provide a likelihood-based framework, and unlike VAEs, they avoid restrictive parametric assumptions about the latent space. Moreover, their iterative denoising dynamics align well with the notion of refining approximate structures toward physically plausible conformations.

Recent work has demonstrated the potential of diffusion-based approaches in protein modelling and design. For example, RFdiffusion Watson et al. (2023) introduced a powerful framework for de novo protein design, achieving unprecedented results in generating novel folds and functional structures. FoldingDiff Wu et al. (2024) proposed a diffusion-based model that explicitly learns the folding process, while DiffDock Yim et al. (2024) applied diffusion models to protein-ligand docking, highlighting the versatility of the paradigm in structural biology. These developments underscore the promise of diffusion generative models in capturing the rich geometric and biophysical constraints inherent to protein structures.

Applying diffusion models to proteins requires careful representation choices. In this work, we consider the protein backbone as a graph where nodes correspond to Cα atoms and edges encode both spatial proximity and sequential adjacency. This enables the use of graph neural networks (GNNs) Scarselli et al. (2009) as score models, incorporating geometric information through radial basis encodings, directional vectors, and positional embeddings. By training such models on large structural datasets, it becomes possible to learn score functions over protein conformational space and to generate new backbone structures through reverse diffusion.

1.4 Structure of the Dissertation

Section 2 presents the theoretical foundations of score-based diffusion, the representation of protein backbones, and the model architectures explored. Section 3 describes the experimental setup, training procedure, and evaluation metrics, and reports quantitative and qualitative results. And discusses the implications of our findings, limitations, and avenues for future research. Section 4 concludes the dissertation by summarising contributions and highlighting potential future directions.

2 Score-Based Diffusion for Generative Modelling

2.1 Background and Motivation

Generative modelling aims to learn a distribution from which new samples can be drawn that resemble those observed in a dataset. Classical approaches include variational autoencoders (VAEs) Kingma and Welling (2022), which maximise a variational lower bound on the likelihood, and generative adversarial networks (GANs) Goodfellow et al. (2020), which frame generation as an adversarial game. While each of these paradigms has been successful, they also suffer from characteristic drawbacks such as blurry reconstructions (VAEs) and mode collapse (GANs).

Diffusion models represent a more recent class of generative models that address many of these limitations. The central idea is deceptively simple: one defines a forward process that gradually corrupts data with noise until the signal is destroyed, and then learns a reverse process that reconstructs data from noise. By formulating the forward process as a diffusion and training a neural network to approximate the reverse dynamics, diffusion models combine the flexibility of deep networks with the probabilistic rigour of latent-variable models.

In their original formulation, denoising diffusion probabilistic models (DDPMs) Ho et al.

(2020) defined the forward process as a discrete-time Gaussian Markov chain. Subsequent work by Song et al. Song et al. (2020) showed that diffusion models can be generalised to continuous-time stochastic differential equations (SDEs). In this framework, the key object is the score function, i.e. the gradient of the log-density of the noisy distribution. Learning this score function allows one to simulate the reverse SDE and thereby generate new samples.

This chapter develops the theoretical foundations of score-based diffusion models in detail. We begin with the definition of the forward diffusion process, proceed to the derivation of the reverse-time dynamics, introduce score matching as the training objective, and describe practical sampling algorithms and noise schedules. Together, these elements form the mathematical basis for applying diffusion models to protein backbone generation in later chapters.

2.2 Forward Diffusion Processes

The first component of a diffusion generative model is the forward, or noising, process. This process gradually perturbs the data distribution into a tractable prior distribution, typically a standard Gaussian. The key requirement is that this transformation be both easy to sample from and analytically tractable, so that training objectives can be computed efficiently.

2.2.1 Discrete-Time Formulation

Denoising diffusion probabilistic models (DDPMs) Ho et al. (2020) define the forward process as a Markov chain of T steps:

$$q\left(x_{1:T} \mid x_0\right) = \prod_{t=1}^{T} q\left(x_t | x_{t-1}\right),$$

where each transition adds a small amount of Gaussian noise,

$$q\left(x_t \mid x_{t-1}\right) = \mathcal{N}\Big(x_t; \sqrt{1-\beta_t}\, x_{t-1},\ \beta_t I\Big).$$

Here, $\{\beta_t\}_{t=1}^{T}$ is a variance schedule with $\beta_t \in (0,1)$ controlling the noise injected at each step. This design ensures that after sufficiently many steps, the distribution of x_T approaches an isotropic Gaussian, i.e. $q(x_T) \approx \mathcal{N}(0, I)$.

Remark 2.1. *A key property of the Gaussian forward process is that one can sample x_t at any arbitrary time step directly from the data x_0:*

$$q\left(x_t \mid x_0\right) = \mathcal{N}\Big(x_t; \sqrt{\bar{\alpha}_t}\, x_0,\ (1-\bar{\alpha}_t) I\Big),$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. This closed form is crucial for training, as it allows drawing noisy samples x_t without explicitly simulating all intermediate steps.

2.2.2 Continuous-Time Formulation

Song et al. Song et al. (2020) generalised the discrete diffusion process to continuous time by taking the limit as $T \to \infty$ and $\beta_t \to 0$. In this setting, the forward process can be expressed as a stochastic differential equation (SDE) of the form

$$dx = f(x,t)\,dt + g(t)\,dW_t, \tag{1}$$

where W_t is the Brownian motion, $f(x,t)$ is a drift term, and $g(t)$ controls the diffusion magnitude. Different choices of f and g yield different diffusion processes,

- **Variance-preserving (VP) SDE**: maintains the marginal variance of x_t at one throughout the process.
- **Variance-exploding (VE) SDE**: variance grows unbounded as $t \to T$, pushing the distribution to white noise.
- **Sub-variance-preserving (sub-VP) SDE**: an interpolation between VP and VE.

2.2.3 VP SDE

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

$$dx = -\frac{1}{2}\beta(t)\,x\,dt + \sqrt{\beta(t)}\,dW_t, \tag{2}$$

where $\beta(t)$ is a noise schedule (e.g. linear, cosine), with closed form posterior

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

for data perturbation, i.e.,

$$x_t = \sqrt{e^{-\int_0^t \beta(s)ds}}x_0 + \sqrt{1 - e^{-\int_0^t \beta(s)ds}}z_t, \quad z_t \sim \mathcal{N}\left(0, I\right). \tag{3}$$

Notice that eq. (3) is exactly the discrete perturbation scheme in DDPM Ho et al. (2020).

This continuous formulation provides a flexible mathematical foundation that unifies discrete DDPMs with stochastic differential equations, and it allows the use of stochastic calculus tools to analyse and manipulate diffusion models. In particular, it enables the derivation of the reverse-time SDE, which defines the generative process and will be introduced in the next section.

2.3 Reverse-Time SDE

The forward diffusion process, whether in discrete or continuous form, progressively perturbs data until its distribution approaches a simple prior such as an isotropic Gaussian. To perform generative modelling, we require a process that inverts this corruption: starting from noise and evolving back toward the data distribution. This is formalised through the theory of time-reversal for diffusion processes.

2.4 Discrete-Time Reverse Process

In the discrete DDPM formulation Ho et al. (2020), the generative model is defined as a reverse Markov chain

$$p_\theta\left(x_{0:T}\right) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t),$$

with $p(x_T) = \mathcal{N}(x_T; 0, I)$ as the prior. The reverse conditionals are modelled as Gaussians

$$p_\theta\left(x_{t-1} \mid x_t\right) = \mathcal{N}\big(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\big),$$

where μ_θ and Σ_θ are learned using a neural network.

2.4.1 Continuous-Time Reverse SDE

In the continuous-time setting, Anderson Anderson (1982) established that the time reversal of an Itô SDE (1) is itself an SDE with modified drift:

$$dx = \left[f(x,t) - g(t)^2 \nabla_x \log p_t(x)\right] \, dt + g(t)\, d\bar{W}_t, \tag{4}$$

where $\bar{W}_t$ is the Brownian motion running backward in time. This equation defines the reverse-time SDE. It depends explicitly on the score function $\nabla_x \log p_t(x)$ of the forward process marginal distribution $p_t(x)$. Thus, if one can estimate the score function accurately for all times t, it becomes possible to simulate the reverse SDE and generate samples from the data distribution.

2.4.2 Implications for Generative Modelling

This result provides the key insight underlying score-based diffusion: generative modelling reduces to score estimation. Rather than directly modelling likelihoods or sampling distributions, we train a neural network $s_\theta(x,t)$ to approximate the score $\nabla_x \log p_t(x)$. The network is then used to guide the drift of the reverse SDE, producing realistic samples when integrated from Gaussian noise at $t = T$ back to $t = 0$.

2.5 Score Matching and Training Objective

The reverse-time SDE depends on the score function $\nabla_x \log p_t(x)$, which is generally intractable. Thus, the central learning problem in score-based diffusion is to approximate this score with a neural network $s_\theta(x,t)$. Training requires a loss function that encourages s_θ to match the true score across noise levels.

2.5.1 Denoising Score Matching

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)

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

Hyvärinen Hyvärinen (2005) introduced score matching as a method to estimate unnormalised density models. Directly minimising the Fisher divergence between the model score and the data score avoids the need to compute the partition function. However, applying this to diffusion models requires extending the idea to noisy samples.

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)

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

Vincent Vincent (2011) showed that training a denoising autoencoder to reconstruct clean data from corrupted observations is equivalent to score matching under certain conditions. This connection underpins the training of diffusion models: the neural network is trained to denoise x_t into x_0, thereby learning the score function implicitly.

2.5.2 Objective in Continuous-Time Diffusion

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

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

Song et al. Song et al. (2020) formulated the training objective for continuous SDEs as a weighted denoising score matching loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\,\mathbb{E}_{x_0\sim p_{\text{data}}}\,\mathbb{E}_{x_t\sim q(x_t|x_0)}\Big[\lambda(t)\,\|s_\theta(x_t, t) - \nabla_x \log q(x_t \mid x_0)\|^2\Big],$$

where $q(x_t \mid x_0)$ denotes the forward diffusion distribution, and $\lambda(t)$ is a time-dependent weighting function. The target score $\nabla_x \log q(x_t \mid x_0)$ has a closed form under the Gaussian forward process, enabling efficient training.

2.5.3 Summary

The learning problem for score-based diffusion therefore reduces to denoising score matching: the neural network $s_\theta(x, t)$ is trained to approximate $\nabla_x \log p_t(x)$ across different noise levels. Once trained, this network can be used to drive the reverse SDE, enabling generation of new samples from Gaussian noise.

2.6 Sampling Procedures

Once the score network $s_\theta(x, t)$ has been trained, new samples can be generated by simulating the reverse diffusion process. This requires integrating the reverse-time SDE (4) starting from Gaussian noise $x(T) \sim \mathcal{N}(0, I)$ and evolving toward $t = 0$.

2.6.1 Reverse SDE Sampling

The simplest approach is to discretise the reverse SDE using the Euler-Maruyama method. For a decreasing sequence of time steps $\{t_i\}_{i=0}^N$ with $t_N = T$ and $t_0 = 0$, the update rule is

$$x_{t_{i-1}} = x_{t_i} + \left[f(x_{t_i}, t_i) - g(t_i)^2 s_\theta(x_{t_i}, t_i)\right] \Delta t + g(t_i)\sqrt{\Delta t}\, z_i,$$

where $z_i \sim \mathcal{N}(0, I)$ and $\Delta t = t_{i-1} - t_i$. This procedure generates approximate samples from the data distribution. While straightforward, its quality depends heavily on the number of discretisation steps: smaller steps yield higher fidelity at the cost of computational time.

2.6.2 Predictor-Corrector Sampling

The Predictor-Corrector sampler was proposed by Song et al. (2020) as an improved sampling method. It is based on combining two components:

1. **Predictor step:** one update using the reverse SDE (as above), advancing the trajectory toward lower noise levels.
2. **Corrector step:** one or more steps of stochastic refinement, implemented as Langevin dynamics:

   $$x \leftarrow x + \alpha\, s_\theta(x, t) + \sqrt{2\alpha}\, z, \quad z \sim \mathcal{N}(0, I),$$

   where α is a step size. This step locally improves sample quality by pushing the state toward regions of higher density according to the score network.

By alternating predictor and corrector steps, the algorithm achieves a better trade-off between quality and efficiency than pure reverse SDE sampling. In practice, only a few corrector iterations per time step are needed to significantly improve sample fidelity.

2.6.3 Summary

Sampling in score-based diffusion models is performed by simulating the reverse SDE, either directly or with predictor-corrector refinements. Both methods start from Gaussian noise and progressively refine the signal into a clean sample, guided by the score network. In this dissertation, we adopt the variance-preserving (VP) SDE with a cosine noise schedule and use predictor-corrector sampling as our default procedure.

2.7 Noise Schedules

The design of the noise schedule plays a central role in the performance of score-based diffusion models. The schedule determines how the variance of the forward diffusion process evolves over time, which directly influences both training stability and the quality of generated samples.

Linear Schedules

In the original DDPM formulation Ho et al. (2020), the variance schedule $\{\beta_t\}_{t=1}^T$ was chosen to increase linearly from a small value to a larger one over T steps. Equivalently, in the continuous SDE formulation, this corresponds to a linear growth of the noise rate $\beta(t)$. While simple, this schedule can be suboptimal: early time steps may inject too little noise (leading to poor coverage of high-noise regions during training), while later steps may over-noise the data (resulting in wasted computation).

Cosine Variance-Preserving Schedule

Nichol and Dhariwal Nichol and Dhariwal (2021) proposed a cosine schedule for the variance-preserving (VP) SDE, which has since become widely adopted. Instead of linear growth, the cosine schedule ensures a smoother increase in noise variance, matching the cumulative signal-to-noise ratio (SNR) to a cosine function:

$$\bar{\alpha}(t) = \frac{\cos^2\left(\frac{\pi}{2}\left(\frac{t+s}{1+s}\right)\right)}{\cos^2\left(\frac{\pi}{2}\frac{s}{1+s}\right)},$$

where $\bar{\alpha}(t)$ is the cumulative product of noise attenuation coefficients, T is the diffusion horizon, and s is a small offset (typically 10^{-4}) to prevent singularities. This schedule maintains higher signal levels for longer, enabling the score network to learn more effectively across a wider range of noise intensities.

Remark 2.2.

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t} z_t, \quad \alpha_t = \cos^2\left(\frac{t+s}{1+s} \cdot \frac{\pi}{2}\right), \quad z_t \sim \mathcal{N}(0, I).$$

This is a DDPM perturbation scheme, the corresponding SDE is given by solving

$$\alpha_t = \cos^2\left(\frac{t+s}{1+s} \cdot \frac{\pi}{2}\right) = e^{-\int_0^t \beta(s)ds},$$

it follows that

$$\begin{aligned}
\beta(t) &= -\frac{d}{dt}\log\cos^2\left(\frac{t+s}{1+s} \cdot \frac{\pi}{2}\right) \\
&= -\frac{1}{\cos^2\left(\frac{t+s}{1+s} \cdot \frac{\pi}{2}\right)}\left[2\cos\left(\frac{t+s}{1+s} \cdot \frac{\pi}{2}\right)\right]\left[-\sin\left(\frac{t+s}{1+s} \cdot \frac{\pi}{2}\right)\right]\frac{d}{dt}\left(\frac{t+s}{1+s} \cdot \frac{\pi}{2}\right) \\
&= \frac{\pi}{1+s}\tan\left(\frac{t+s}{1+s} \cdot \frac{\pi}{2}\right).
\end{aligned}$$

Empirically, the cosine VP schedule improves sample quality and accelerates convergence compared to linear schedules. It balances the trade-off between injecting sufficient noise for diversity and preserving enough signal for stable training. Moreover, because it avoids excessively noisy late stages, fewer discretisation steps are needed during sampling, reducing computational cost.

Choice in This Work

In this dissertation, we adopt the cosine VP schedule as the forward diffusion process. This choice is motivated by its demonstrated effectiveness in image and structural generative tasks, and by its compatibility with the variance-preserving SDE formulation used in our score model. All experiments in later chapters therefore use the cosine VP schedule as the default setting.

2.8 Summary

This chapter has established the theoretical foundations of score-based diffusion models, which form the core generative framework employed in this dissertation. We began by introducing the forward diffusion process, first in its discrete DDPM formulation and then in its continuous-time generalisation as a stochastic differential equation. We then derived the reverse-time SDE, showing that generative modelling reduces to the problem of score estimation. This led naturally to the denoising score matching objective, which enables training a neural network to approximate the score function across all noise levels. Finally, we discussed practical sampling algorithms, including reverse SDE integration and predictor-corrector methods, and examined the role of noise schedules, with particular emphasis on the cosine variance-preserving schedule adopted in this work.

Together, these elements provide a principled probabilistic framework for deep generative modelling. Unlike GANs or VAEs, score-based diffusion models avoid common issues such as mode collapse or restrictive latent assumptions, while offering strong theoretical guarantees and flexible sampling procedures. The remainder of this dissertation builds upon these foundations to adapt score-based diffusion to the setting of protein backbone generation, where structural constraints and geometric representations introduce unique modelling challenges.

3 Experiments

3.1 Data

CATH S40 domain structures, it contains non-redundant data with no pair of domains with ≥ 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.

3.2 Model Architecture

3.2.1 Message Passing Layers

Graph Neural Networks Graph Neural Networks (GNNs) were first introduced in the seminal work by Scarselli et al. (2009), which proposed a recursive framework for learning over graphs. The field progressed significantly with the development of Graph Convolutional Networks (GCNs) by Kipf and Welling (2017), which introduced a scalable and differentiable message-passing scheme. Further, Graph Attention Networks Veličković et al. (2018), brought in attention mechanisms to improve expressiveness and performance. To better model the complicated structure, I use the graph transformer network Shi et al. (2021) which incorporates multi-head self-attention over local neighborhoods.

3.2.2 Time Conditioning via Gaussian Fourier Features

Following Song et al. (2020), we embed the diffusion time step $t \in (0, 1)$ using random Fourier features Tancik et al. (2020)

$$\gamma(t) = [\cos(2\pi W t), \sin(2\pi W t)], \quad W \sim \mathcal{N}\left(0, s^2\right),$$

to provide continuous and expressive encoding of time.

3.2.3 Positional Encoding for Node Order Awareness

Graph-based models are inherently invariant to node permutations, which can be a limitation when modeling protein backbones, where the sequential order of residues encodes biologically meaningful directionality along the chain. To inject this ordering information, each node is assigned a scalar index $p \in [0, 1]$ representing its normalized position in the sequence. This index is first mapped to a high-dimensional representation using a sinusoidal encoding Vaswani et al. (2017)

$$\rho(p) = [\sin(\omega_1 p), \cos(\omega_1 p), \sin(\omega_2 p), \cos(\omega_2 p), \ldots]$$

where w_k's are fixed frequencies. The resulting positional embedding is then passed through a learnable transformation, such as a multilayer perceptron (MLP), before being incorporated into the model

3.2.4 Residual Network Architecture

Overall residual connections are used at each GNN layer.

4 Conclusion

Appendix

```
Put your R code here.
```

References

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

Dhariwal, P. and Nichol, A. (2021). Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Commun. ACM*, 63(11):139–144.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

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

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.

Kingma, D. P. and Welling, M. (2022). Auto-Encoding Variational Bayes.

Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks.

Nichol, A. and Dhariwal, P. (2021). Improved Denoising Diffusion Probabilistic Models.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., and Sun, Y. (2021). Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification.

Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021). Maximum Likelihood Training of Score-Based Diffusion Models. In *Advances in Neural Information Processing Systems*, volume 34, pages 1415–1428. Curran Associates, Inc.

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*.

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., and Ng, R. (2020). Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ukasz Kaiser, Ł., and Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph Attention Networks.

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

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Ovchinnikov, S., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., and Baker, D. (2023). De novo design of protein structure and function with RFdiffusion. *Nature*, 620(7976):1089–1100.

Wu, K. E., Yang, K. K., van den Berg, R., Alamdari, S., Zou, J. Y., Lu, A. X., and Amini, A. P. (2024). Protein structure generation via folding diffusion. *Nature Communications*, 15(1):1059.

Yim, J., Stärk, H., Corso, G., Jing, B., Barzilay, R., and Jaakkola, T. S. (2024). Diffusion models in protein structure and docking. *WIREs Computational Molecular Science*, 14(2):e1711.