Computational and Structural Biotechnology Journal 23 (2024) 1641–1653

Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

Method Article

Annealed fractional Lévy–Itō diffusion models for protein generation

Eric Paquet [a,b,*], Farzan Soleymani [c], Herna Lydia Viktor [b], Wojtek Michalowski [c]

[a] National Research Council, 1200 Montreal Road, Ottawa, ON, K1A 0R6, Canada
[b] School of Electrical Engineering and Computer Science, University of Ottawa, ON, K1N 6N5, Canada
[c] Telfer School of Management, University of Ottawa, ON, K1N 6N5, Canada

ARTICLE INFO

Keywords:
Diffusion
Fractional
Generative model
Lévy–Itō
Noising
Protein
Stable distribution
Stochastic differential equation
Score

ABSTRACT

Protein generation has numerous applications in designing therapeutic antibodies and creating new drugs. Still, it is a demanding task due to the inherent complexities of protein structures and the limitations of current generative models. Proteins possess intricate geometry, and sampling their conformational space is challenging due to its high dimensionality. This paper introduces novel Markovian and non-Markovian generative diffusion models based on fractional stochastic differential equations and the Lévy distribution, allowing for a more effective exploration of the conformational space. The approach is applied to a dataset of 40, 000 proteins and evaluated in terms of Fréchet distance, fidelity, and diversity, outperforming the state-of-the-art by 25.4%, 35.8%, and 11.8%, respectively.

1. Introduction

Proteins perform numerous cellular functions such as enzymatic activity, structural support, transport and storage, signalling, regulation of gene expression, immune response, and catalysis of biochemical reactions [1]. They consist of linear chains of amino acids. These sequences form three-dimensional structures known as conformations, which result from the interactions between the amino acids and their environment [2]. These conformations determine, in turn, the functionality of proteins [3]. Protein generative models hold immense potential in various fields, from medicine to materials science. Generative models can be employed to design new therapeutic proteins [4] and understand disease mechanisms and protein dysfunctions [5]. They may accelerate research through cost-effective rapid prototyping [6] and may be customised for specific functions [7] in addition to being able to generate proteins that do not exist in nature. In materials science, generated proteins may be employed to create new biomaterials and to form nanoscale structures [8].

Nonetheless, generating proteins is a challenging problem due to the complexity of their geometry [9], the limited amount of structural data for specific proteins, the conformational variability, and importantly, the high dimensionality of their conformational space, which makes sampling a daunting task [10]. This paper proposes novel Markovian and non-Markovian diffusion probabilistic models based on fractional stochastic differential equations (SDEs) and the Lévy distribution [11], which allow for a more effective exploration of the conformational space and more accurate generated results. The proposed approach replaces the Wiener process in the forward noising equation with a Lévy process. The Lévy distribution is heavy-tailed and, unlike the Gaussian distribution, entails large fluctuations. The process may be reversed with a backward denoising fractional SDE [12], which involves fractional derivatives. Instead of employing only one distribution for noising, a family of Lévy distributions is used. The Lévy distribution is annealed from large to small fluctuations (i.e. from a heavy-tailed distribution to a Gaussian distribution), in a process reminiscent of simulated annealing, to improve the convergence of the calculations. Fifteen Lévy–Itō models are evaluated against five non-fractional state-of-the-art models on a dataset of 40,000 proteins with three metrics: the Fréchet distance [13], the density, and the coverage [14].

The paper is organised as follows. Diffusion processes, SDEs, and score-matching techniques are introduced in Section 2.1. The solution of the backward equation with the exponential integrator method is presented in Section 2.2. Lévy–Itō diffusion models are introduced in Section 3, while the fractional Riesz derivative approximation and the stability index annealing are addressed in Section 4. The representation of proteins is

[*] Corresponding author at: National Research Council, 1200 Montreal Road, Ottawa, ON, K1A 0R6, Canada.
E-mail addresses: Eric.Paquet@nrc-cnrc.gc.ca (E. Paquet), fsoleyma@uottawa.ca (F. Soleymani), hviktor@uottawa.ca (H.L. Viktor), wojtek@telfer.uottawa.ca (W. Michalowski).

https://doi.org/10.1016/j.csbj.2024.04.009
Received 24 January 2024; Received in revised form 3 April 2024; Accepted 4 April 2024
Available online 17 April 2024

discussed in Section 5. The implementation and the methodology appear in Section 6, and the experimental results and their discussion are reported in Section 7. Section 8 concludes the paper. All the mathematical symbols appearing in this article are defined in Table A.4.

2. Background

Score-based probabilistic diffusion models are a subclass of diffusion models that employ a score function to guide the generation process. This score function estimates the gradient of the log probability of the data, providing a way to progressively denoise a sample from a random distribution to a data-like distribution [15,16]. The relevance of these models to protein generation and biological systems is manifold. Score-based models can generate high-quality samples of protein structures by learning the distribution of existing protein data. They can be used to predict the structure of proteins or to design new proteins with desired functions [17]. The iterative refinement process employed in these models mimics the natural folding of proteins, potentially providing insights into how amino acid sequences determine their 3-D structure [18]. The mathematical framework of score-based diffusion models, often involving stochastic differential equations (SDEs), captures the complex, high-dimensional relationships inherent in biological data. This may be crucial for understanding multifaceted biological systems in which interactions at the molecular level influence macroscopic behaviour. Protein space is vast and largely unexplored. Score-based diffusion models can efficiently explore this space by generating novel protein sequences and structures, facilitating the discovery of proteins with unique or enhanced functionalities for medical and industrial applications [19,20].

The underlying SDEs in score-based diffusion models are similar to the dynamic processes observed in biological systems, in which deterministic and stochastic factors influence changes over time. This similarity may make these models particularly useful for simulating and understanding the dynamics of biological processes, such as enzymatic reactions, cellular signalling pathways or evolutionary changes [21]. Biological systems are often noisy and subject to multiple sources of uncertainty. Score-based diffusion models naturally incorporate noise and uncertainty, which may be advantageous for accurate modelling of biological phenomena that are inherently stochastic [16,22,23]. These models can integrate different types of biological data (e.g., genomic, proteomic, metabolomic) to generate comprehensive models of biological systems. This integrative approach may lead to a deeper understanding of the underlying mechanisms of complex biological processes and diseases. In summary, score-based probabilistic diffusion models, with their robust mathematical foundation and ability to model complex, noisy and high-dimensional data, have significant potential to advance our understanding and capabilities in protein generation and broader biological systems analysis. Their ability to generate new, plausible patterns from learned data distributions makes them a powerful tool for innovation in synthetic biology, drug design and systems biology.

2.1. Diffusion processes, SDEs, and score-matching techniques

A diffusion model consists of a fixed forward noising process that adds noise to the data and a learned backward denoising process that iteratively removes noise from them. The denoising process is trained to match the corresponding noising process at each iteration. Samples from the data probability distribution may be generated from random noise by simulating the backward diffusion process [15,24]. The forward noising process may be represented by an SDE such as

$$d\mathbf{x}_t = \mathbf{F}_t\mathbf{x}dt + \mathbf{G}_t d\mathbf{w}, \quad \mathbf{x}\in\mathbb{R}^D, \mathbf{F}\in\mathbb{R}^{D\times D}, \mathbf{G}\in\mathbb{R}^{D\times D}, \mathbf{w}\in\mathbb{R}^D \tag{1}$$

where $\mathbf{F}_t$ is the drift matrix, $\mathbf{G}_t$ is the diffusion matrix, and $\mathbf{w}$ is a standard Wiener process [25]. This equation is linear in both the drift and the diffusion and corresponds to the noising process in which noise is gradually added to the data until a white noise is obtained. As demonstrated by [26,27], the reverse-time diffusion process has a closed-form solution:

$$d\mathbf{x}_t = \left[\mathbf{F}_t\mathbf{x} - \mathbf{G}_t\mathbf{G}_t^T\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\right]dt + \mathbf{G}_t d\bar{\mathbf{w}} \tag{2}$$

where $p_t(\mathbf{x})$ is the data probability distribution, $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ is its gradient, known as the score function, and $\bar{\mathbf{w}}$ denotes a standard Wiener process in the reverse-time direction. This equation corresponds to the denoising process in which noise is progressively removed until a new datum, distributed according to $p_t(\mathbf{x})$, is generated. The score function is unknown [15,24] and must be approximated with a score-matching technique:

$$\mathbb{E}_{p(\mathbf{x}_t)}\left[\frac{1}{2}\left\|\mathbf{s}_\theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)\right\|^2_{\mathbf{\Lambda}_t}\right] \tag{3}$$

where $\mathbf{s}_\theta(\mathbf{x}_t;t)$ is a time-dependent deep neural network called the score network, represents the network parameters, $\mathbb{E}$ is the mathematical expectation, and $\mathbf{\Lambda}_t$ is a metric (a symmetric matrix). The mathematical expectation is defined as

$$\mathbb{E}_{p(\mathbf{x})}f(\mathbf{x}) \overset{\wedge}{=} \int p(\mathbf{x})f(\mathbf{x})d\mathbf{x} \tag{4}$$

while the metric is associated with a quadratic form:

$$\left\|\nabla_{\mathbf{x}_t}\log p_{0t}(\mathbf{x}_t) - \mathbf{s}_\theta(\mathbf{x}_t,t)\right\|^2_{\mathbf{\Lambda}_t} = \left[\nabla_{\mathbf{x}_t}\log p_{0t}(\mathbf{x}_t)\right]\mathbf{\Lambda}_t\mathbf{s}_\theta(\mathbf{x}_t,t) \tag{5}$$

The metric is determined by the data and the nature of the problem and is often approximated with the identity matrix.

Unfortunately, $\nabla_{\mathbf{x}}\log p_t(\mathbf{x})$ does not have a closed-form solution and is intractable. Nonetheless, as demonstrated by [27,28], the score function may be evaluated by employing the gradient of the noising distribution $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_0)$:

$$\mathbb{E}_{p(\mathbf{x}_t)}\left[\frac{1}{2}\left\|\mathbf{s}_\theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)\right\|^2_{\mathbf{\Lambda}_t}\right] = \mathbb{E}_{p(\mathbf{x}_0)p_{0t}(\mathbf{x}_t|\mathbf{x}_0)}\left[\frac{1}{2}\left\|\mathbf{s}_\theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x}_0)\right\|^2_{\mathbf{\Lambda}_t}\right] + \Omega \tag{6}$$

where Ω is a constant, and $p(\mathbf{x}_t|\mathbf{x}_0)$ is the probability, at time t, of having a noisy datum $\mathbf{x}_t$ given an uncorrupted datum $\mathbf{x}_0$. As opposed to the score function, the gradient of the noising distribution has a fixed closed form, which is often chosen to be Gaussian:

$$p_{0t}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mu_t\mathbf{x}_0, \mathbf{\Sigma}_t), \quad \mu_t, \mathbf{\Sigma}_t \in \mathbb{R}^{D\times D}, \mathbf{L}_t\mathbf{L}_t^T = \mathbf{\Sigma}_t \tag{7}$$

Table 1
VPSDE parametrisation.

$\mathbf{F}_t$	$\mathbf{G}_t$	$\boldsymbol{\mu}_t$	$\boldsymbol{\Sigma}_t$
$\frac{1}{2}\frac{d\log\beta_t}{dt}\mathbf{I}$	$\sqrt{-\frac{d\log\beta_t}{dt}}\mathbf{I}$	$\sqrt{\beta_t}\mathbf{I}$	$\left(1-\beta_t\right)\mathbf{I}$

with mean $\boldsymbol{\mu}_t$ and covariance matrix $\boldsymbol{\Sigma}_t$. It is convenient to express the covariance matrix in terms of its Cholesky decomposition, where $\mathbf{L}_t$ is the Cholesky matrix [24]. This factorisation is the product of a lower triangular matrix and its transpose. The parameters of the score network may be learned by minimising the denoising score matching loss [29,30] with stochastic gradient descent optimisation techniques employing mini-batches [31], such as the adaptive moment estimation method (Adam) [31]:

$$\theta^* = \arg\min_{\theta} \mathcal{L}\left(\theta\right)$$
$$\mathcal{L}\left(\theta\right) = \mathbb{E}_{t\sim\mathcal{U}[0,T]}\boldsymbol{\Lambda}\left(t\right)\mathbb{E}_{p(\mathbf{x}_0)p_{0t}(\mathbf{x}_t|\mathbf{x}_0)}\left[\left\|\nabla_{\mathbf{x}_t}\log p_{0t}\left(\mathbf{x}_t\left|\mathbf{x}_0\right.\right) - \mathbf{s}_\theta\left(\mathbf{x}_t,t\right)\right\|^2_{\boldsymbol{\Lambda}_t}\right], \quad \boldsymbol{\Lambda}\left(t\right)\in\mathbb{R}^+ \tag{8}$$

where $\mathcal{L}\left(\theta\right)$ is the loss function, $\mathcal{U}$ is the uniform distribution, and $\boldsymbol{\Lambda}\left(t\right)$ is a positive weighting function. It is important to note that the time must be sampled randomly (from a uniform distribution) to avoid bias when training the score network. Therefore, $\left\{\mathbf{x}_t\right\}\left|t\in[0,T]\right.$ must be time-encoded for the optimisation algorithm to identify the time correctly [32]. The time-dependent weight $\boldsymbol{\Lambda}\left(t\right)$ [33,27] determines the importance assigned to each time step and depends on the nature of the problem and the data. It may also be used as a forgetting mechanism.

The score tends to vary rapidly. As pointed out by [30], parametrising the score network in terms of the learned noise $\boldsymbol{\varepsilon}_\theta\left(\mathbf{x},t\right)$, that is,

$$\mathbf{s}_\theta\left(\mathbf{x}_t,t\right) \approx -\mathbf{L}_t^{-T}\boldsymbol{\varepsilon}_\theta\left(\mathbf{x}_t,t\right) \tag{9}$$

may significantly improve the accuracy of the learning process. Therefore, when expressed in terms of the learned noise, the training loss function becomes:

$$\bar{\mathcal{L}}\left(\theta\right) = \mathbb{E}_{t\sim\mathcal{U}[0,T]}\mathbb{E}_{p(\mathbf{x}_0),\,\boldsymbol{\varepsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\left\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta\left(\boldsymbol{\mu}_t\mathbf{x}_0 + \mathbf{L}_t\boldsymbol{\varepsilon},t\right)\right\|^2_{\bar{\boldsymbol{\Lambda}}_t}\right], \quad \bar{\boldsymbol{\Lambda}}_t = \mathbf{L}_t^{-1}\boldsymbol{\Lambda}_t\mathbf{L}_t^{-T} \tag{10}$$

This equation is employed to determine the parameters of the noise network $\boldsymbol{\varepsilon}_\theta\left(\mathbf{x}_t,t\right)$.

2.2. Solution of the backward equation with the exponential integrator method

Once the score network has been learned, it may be substituted in the backward diffusion equation in place of the score function:

$$d\hat{\mathbf{x}}_t = \left[\mathbf{F}_t\hat{\mathbf{x}} + \frac{1+\eta^2}{2}\mathbf{G}_t\mathbf{G}_t^T\mathbf{L}_t^{-T}\boldsymbol{\varepsilon}_\theta\left(\hat{\mathbf{x}},t\right)\right]dt + \eta\,\mathbf{G}_t d\bar{\mathbf{w}}, \quad \eta\geq 0 \tag{11}$$

A parameter η is introduced here to further generalise the backward equation to SDEs and ordinary differential equations (ODEs): it is equal to 1 for SDEs and 0 for ODEs. This equation may be solved with an explicit numerical method, such as the Euler method [34]. Unfortunately, this approach results in low accuracy and is unstable when the step size is insufficiently small [27,35]. Therefore, [24] proposed solving Eq. (11) with an exponential integrator (EI) to take advantage of the semi-linear structure of the reverse process, a technique known as diffusion exponential integrator sampler (DEIS):

$$\hat{\mathbf{x}}_{t-\Delta t} = \boldsymbol{\Psi}\left(t-\Delta t,t\right)\hat{\mathbf{x}}_t + \left[\int_t^{t-\Delta t}\frac{1+\eta^2}{2}\boldsymbol{\Psi}\left(t-\Delta t,\tau\right)\mathbf{G}_\tau\mathbf{G}_\tau^T\mathbf{L}_\tau^{-T}\boldsymbol{\varepsilon}_\theta\left(\hat{\mathbf{x}}_\tau,\tau\right)d\tau\right] + \int_t^{t-\Delta t}\eta\boldsymbol{\Psi}\left(t-\Delta t,t\right)\mathbf{G}_\tau d\bar{\mathbf{w}} \tag{12}$$

where Δt is a small time interval, $\hat{\mathbf{x}}_t$ is an estimate of $\mathbf{x}_t$, and the transition matrix $\boldsymbol{\Psi}$ is given by

$$\frac{\partial\boldsymbol{\Psi}\left(t-\Delta t,t\right)}{\partial t} = \mathbf{F}_t\boldsymbol{\Psi}\left(t-\Delta t,t\right), \quad \boldsymbol{\Psi}\left(t,t\right) = \mathbf{I} \tag{13}$$

This solves the backward equation exactly if the noise $\boldsymbol{\varepsilon}_\theta\left(\hat{\mathbf{x}}_t,t\right)$ is constant over the time interval $[t-\Delta t,t]$ [24]. The backward equation does not need to be trained for each value of η. Indeed, a Fokker–Planck–Kolmogorov (FPK) equation may be associated with the backward equation [27]:

$$d\hat{\mathbf{x}}_t = \left[\mathbf{F}_t\hat{\mathbf{x}} - \frac{1+\eta^2}{2}\mathbf{G}_t\mathbf{G}_t^T\nabla_{\mathbf{x}_t}p_t\left(\mathbf{x}\right)\right]dt + \eta\,\mathbf{G}_t d\bar{\mathbf{w}} \Rightarrow \frac{\partial p_t\left(\mathbf{x}\right)}{\partial t} = -\nabla\cdot\left\{\left[\mathbf{F}_t\mathbf{x} - \frac{1}{2}\mathbf{G}_t\mathbf{G}_t^T\nabla_{\mathbf{x}}p_t\left(\mathbf{x}\right)\right]p_t\left(\mathbf{x}\right)\right\} \Rightarrow \frac{\partial p_t\left(\mathbf{x}\right)}{\partial\eta} = 0 \tag{14}$$

This equation does not depend on η and, as demonstrated by [27], the score function is the same regardless of its value. As a result, the score network is trained only once with $\eta = 1$. The parameter η is only employed in the generative process to parametrise the various models. When $\eta = 1$, the model corresponds to the well-known Markovian denoising diffusion probabilistic model (DDPM), and when $\eta = 0$, it corresponds to the non-Markovian denoising diffusion implicit model (DDIM) [36]. These models do not result from an ad hoc procedure but follow naturally from the SDEs and the numerical method (DEIS) employed for solving them. In this work, the variance-preserving SDE (VPSDE) parametrisation is chosen for the drift matrix, the diffusion matrix, and the mean and the covariance of the noising distribution [30], as reported in Table 1.

In this work, the noising parameter is scheduled (parametrised) according to

$$\beta_t = \beta_{\min} + \frac{t}{T}\left(\beta_{\max} - \beta_{\min}\right) \tag{15}$$

with $\beta_{\min} = \beta_0 = 0$ and $\beta_{\max} = \beta_T = 1$ [30]. With this parametrisation, the transition matrix becomes

$$\mathbf{\Psi}(t-\Delta t, t) = \sqrt{\frac{\beta_{t-\Delta t}}{\beta_t}} \tag{16}$$

while the solution of the backward equation with DEIS is given by

$$\hat{\mathbf{x}}_{t-\Delta t} = \underbrace{\sqrt{\beta_{t-\Delta t}}\left(\frac{\hat{\mathbf{x}}_t - \sqrt{1-\beta_t}\,\varepsilon_\theta\left(\hat{\mathbf{x}}_t, t\right)}{\sqrt{\beta_t}}\right)}_{\hat{\mathbf{x}}_0} + \sqrt{1-\beta_{t-\Delta t} - \eta^2 \frac{1-\beta_{t-\Delta t}}{1-\beta_t}\left(1-\frac{\beta_t}{\beta_{t-\Delta t}}\right)}\,\varepsilon_\theta\left(\hat{\mathbf{x}}_t, t\right) + \tag{17}$$
$$\eta\sqrt{\frac{1-\beta_{t-\Delta t}}{1-\beta_t}\left(1-\frac{\beta_t}{\beta_{t-\Delta t}}\right)}\,\varepsilon, \quad \eta \in [0,1], \ \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

This equation implies that the solution is normally distributed with mean μ_η and variance σ_η^2:

$$\begin{gathered} \hat{\mathbf{x}}_{t-\Delta t} \sim \mathcal{N}\left(\mu_\eta, \sigma_\eta^2 \mathbf{I}\right) \\ \mu_\eta = \sqrt{\beta_{t-\Delta t}}\left(\frac{\hat{\mathbf{x}}_t - \sqrt{1-\beta_t}\,\varepsilon_\theta(\hat{\mathbf{x}}_t, t)}{\sqrt{\beta_t}}\right) + \sqrt{1-\beta_{t-\Delta t} - \eta^2 \tfrac{1-\beta_{t-\Delta t}}{1-\beta_t}\left(1-\tfrac{\beta_t}{\beta_{t-\Delta t}}\right)}\,\varepsilon_\theta\left(\hat{\mathbf{x}}_t, t\right) \\ \sigma_\eta^2 = \eta^2 \tfrac{1-\beta_{t-\Delta t}}{1-\beta_t}\left(1-\tfrac{\beta_t}{\beta_{t-\Delta t}}\right) \end{gathered} \tag{18}$$

the latter result being obtained with the reparametrisation trick [37]. The following section extends the previous results to a non-Gaussian distribution, namely the Lévy or stable distribution.

3. Lévy–Itō diffusion models

The Lévy or stable distribution [11] is an extreme distribution [38] with a heavy tail. Unlike the Gaussian distribution, the suppression of large fluctuations is polynomial and not exponential [11], meaning that very large values are most likely to occur:

$$p_L(x; \alpha, \gamma, \mu, \sigma) \sim \left.\frac{C_\pm}{|x|^{1+\alpha}}\right|_{x\to\pm\infty} \tag{19}$$

where $C_\pm$ are constants and $\alpha \in]0,2]$ is the stability index. The stable distribution does not have a closed form, but it may be expressed in terms of the Fourier transform of its characteristic function $\varphi(k; \alpha, \gamma, \mu, \sigma)$:

$$\begin{gathered} p_L(x;\alpha,\gamma,\mu,\sigma) = \int_{-\infty}^{\infty} \varphi(k;\alpha,\gamma,\mu,\sigma)\exp[-2\pi i k x]\,dk, \quad i = \sqrt{-1}\ : \\ \varphi(k;\alpha,\gamma,\mu,\sigma) = \begin{cases} \exp\left[ik\mu - |k\sigma|\left(1 + 2i\frac{\gamma}{\pi}\operatorname{sgn}(k)\log|k\sigma|\right)\right] & \Leftrightarrow \alpha = 1 \\ \exp\left[ik\mu - |k\sigma|^\alpha\left(1 + i\gamma\tan\left(\frac{\pi\alpha}{2}\right)\operatorname{sgn}(k)\left(|k\sigma|^{1-\alpha} - 1\right)\right)\right] & \Leftrightarrow \alpha \neq 1 \end{cases} \end{gathered} \tag{20}$$

where $\gamma \in [-1,1]$ is the skewness parameter, $\mu \in \mathbb{R}$ is the location parameter, and $\sigma \in \mathbb{R}^+$ is the scale parameter. Stable distributions with large stability indices have heavier tails and random walks characterised by larger steps. The skewness determines the distribution's degree of asymmetry (only symmetrical distributions are considered in this work, i.e. $\gamma = 0$). The location parameter specifies the location of the distribution's maximum, while the scale distribution determines its spread. The following alternative notation is also employed for the distribution:

$$\mathcal{S}\alpha\mathcal{S}(\sigma) \equiv p_L(x; \alpha, 0, 0, \sigma) \tag{21}$$

with $\mathcal{S}\alpha\mathcal{S}(\sigma)|_{\alpha=2} = \mathcal{N}\left(0, \sqrt{2}\sigma\right)$ corresponding to the Gaussian distribution. Therefore, the Lévy distribution is a generalisation of the normal distribution.

The Lévy distribution is illustrated for various values of the stability index in Fig. 1, and the corresponding heavy tails are reported in Fig. 2. From a random walk perspective, the Lévy distribution allows transitions much larger than its Gaussian counterpart's [38]. These large steps are required to explore the solution space in its entirety, which may not be readily accessible to small Gaussian steps [38]. This phenomenon is illustrated in Fig. 3, which shows random walks of a hundred steps for various values of the stability index, demonstrating that smaller values of the index favour further exploration of the solution space.

The forward SDE, as described by Eq. (1), may be generalised to include Lévy processes [39,11]:

$$d\mathbf{x}_t = \mathbf{F}_t\mathbf{x}dt + \mathbf{G}_t d\mathbf{w} + \mathbf{H}_t d\mathbf{L}_\alpha \tag{22}$$

where $\mathbf{L}_\alpha = \left[dL_{\alpha,1}, dL_{\alpha,2}, \ldots, dL_{\alpha,D}\right]$ is a Lévy process with stability index α, and $\mathbf{H}_t \in \mathbb{R}^{D\times D}$ is known as the fractional diffusion matrix. It should be noted that a one-dimensional Lévy distribution is associated with each dimension. Such a model is known as a fractional diffusion model (FDM) or a Lévy–Itō model (LIM).

As demonstrated by [40], the reverse-time fractional diffusion denoising process associated with this equation has a closed-form solution, given by

$$d\mathbf{x}_t = \left[\mathbf{F}_t\mathbf{x} - \mathbf{G}_t\mathbf{G}_t^T\nabla_{\mathbf{x}}\log p_t(\mathbf{x})\right]dt + \mathbf{G}_t d\bar{\mathbf{w}} - \alpha\mathbf{H}_t^\alpha \frac{\partial_{|\mathbf{x}_t|}^{\alpha-2}\nabla_{\mathbf{x}_t} p_t\left(\mathbf{x}_t\right)}{p_t\left(\mathbf{x}_t\right)} + \mathbf{H}_t d\bar{\mathbf{L}}_\alpha \tag{23}$$

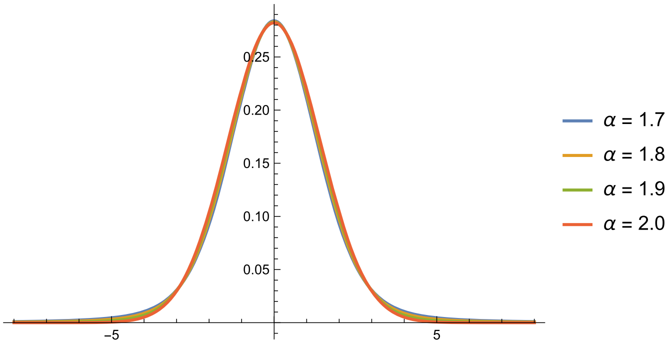

Fig. 1. Lévy distribution for various values of the stability index.

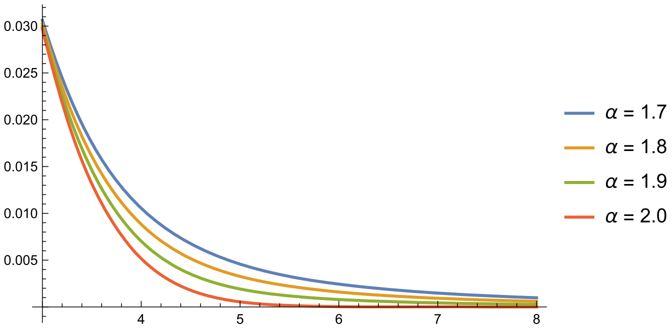

Fig. 2. Lévy distribution heavy tails corresponding to Fig. 1

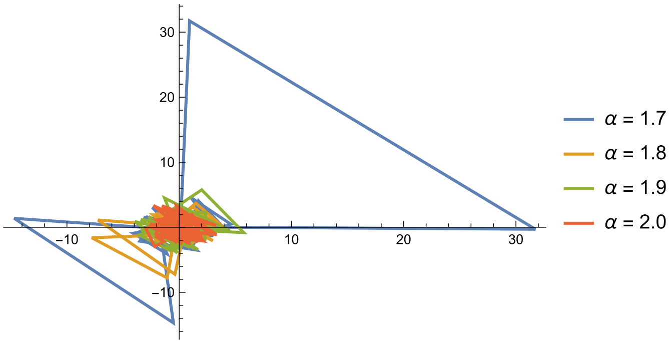

Fig. 3. Lévy random walks corresponding to the distributions shown in Fig. 1

where $\partial^{\alpha-2}_{|\mathbf{x}_t|}$ is the fractional Riesz derivative of order $\alpha - 2$, and $\bar{\mathbf{L}}_\alpha$ is a Lévy process in the reverse-time direction. The fractional exponentiation of $\mathbf{H}_t$ is evaluated using its spectral decomposition [41]:

$$\mathbf{H}_t^\alpha = \mathbf{U}\begin{bmatrix} \lambda_1^\alpha & & & \\ & \lambda_2^\alpha & & \\ & & \ddots & \\ & & & \lambda_D^\alpha \end{bmatrix}\mathbf{U}^T \tag{24}$$

where $\mathbf{U}$ is the eigenvector matrix associated with $\mathbf{H}$ and $\left\{\boldsymbol{\Lambda}_i\right\}_{i=1}^D$ are the corresponding eigenvalues. The fractional Riesz derivative, which is a generalisation of the standard derivative, may be evaluated based on the properties of the Fourier transform [42,43]. Indeed, it is well known from Fourier analysis [44] that the fractional derivative may be obtained by raising the Fourier frequencies by the fractional exponent and then taking the inverse Fourier transform:

$$\begin{gathered} \partial^\alpha_{|\mathbf{x}|}\left[f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_D(\mathbf{x})\right] \equiv \left[\partial^\alpha_{|x_1|} f_1(\mathbf{x}), \partial^\alpha_{|x_2|} f_2(\mathbf{x}), \dots, \partial^\alpha_{|x_D|} f_D(\mathbf{x})\right] \\ \partial^\alpha_{|x_i|} f = \mathcal{F}^{-1}\left(\left|k_i\right|^\alpha \mathcal{F}[f](k)\right) \end{gathered} \tag{25}$$

Here, $\mathcal{F}$ is the Fourier transform and $\mathcal{F}^{-1}$ is the corresponding inverse transform:

$$\begin{aligned} \mathcal{F}[f](\mathbf{k}) &= \int d^D x \ \exp(-2\pi i\, \mathbf{k}\cdot\mathbf{x})\, f(\mathbf{x}) \\ \mathcal{F}^{-1}[f](\mathbf{x}) &= \int d^D k \ \exp(2\pi i\, \mathbf{k}\cdot\mathbf{x})\, f(\mathbf{k}) \end{aligned} \tag{26}$$

where $i = \sqrt{-1}$ is the imaginary unit. Unlike the Wiener process, the backward equation involves not only the score function but also the data distribution and the fractional Riesz derivative of the score function. In this work, only Lévy-driven SDEs are considered:

$$d\mathbf{x}_t = \mathbf{F}_t \mathbf{x} dt + \mathbf{H}_t d\mathbf{L}_\alpha \tag{27}$$

As for the Wiener process, a fractional Fokker–Planck equation (FFPE) [40,38] may be associated with the forward (noising) process:

$$d\mathbf{x}_t = \mathbf{F}_t \mathbf{x} dt + \mathbf{H}_t d\mathbf{L}_\alpha \Rightarrow \frac{\partial p_t(\mathbf{x})}{\partial t} = -\nabla \cdot \left[\mathbf{F}_t \mathbf{x} - \mathbf{H}_t^\alpha \mathbf{q}(t,x)\, p_t(\mathbf{x})\right], \quad q_i(t,x) = \frac{\partial_{|x_i|}^{\alpha-2} \partial_{\mathbf{x}_i} p_t(\mathbf{x})}{p_t(\mathbf{x})} \tag{28}$$

where $\mathbf{q}(t,\mathbf{x}) = \left[q_i(t,x)\right]$.

4. Approximation of the fractional Riesz derivative and annealing of the stability index

The backward fractional equation has a high computational complexity, which may impede the training of the score network. More importantly, unlike its gradient, the noising distribution $p_{0t}\left(\mathbf{x}_t \left| \mathbf{x}_0\right.\right)$ cannot replace the data distribution $p_t(\mathbf{x})$ in the score-matching technique. For this reason, and because the solution of the backward equation is an incremental process, the fractional Riesz derivative is expressed as a truncated series expansion. As proposed by [42], the fractional Riesz derivative may be expanded as

$$\frac{\partial_{|x_i|}^{\alpha-2} \partial_{x_i} p_t(\mathbf{x})}{p_t(\mathbf{x})} \approx \frac{1}{h^{\alpha-2}} \sum_{k\in\mathbb{Z}} \frac{(-1)^k \Gamma(\alpha-1)}{\Gamma\left(\frac{\alpha}{2}-k\right)\Gamma\left(\frac{\alpha}{2}+k\right)} \partial_{x_i} p_t\left(x_1, x_2, \dots, x_i - kh, \dots x_d\right)[1-kh]\, \partial_{x_i} p_t(\mathbf{x}) \tag{29}$$

where $h \in \mathbb{R}^+$ is a parameter. If the expansion is truncated to first order, one obtains

$$\frac{\partial_{|x_i|}^{\alpha-2} \partial_{x_i} p_t(\mathbf{x})}{p_t(\mathbf{x})} \approx \frac{1}{h^{\alpha-2}} \frac{\Gamma(\alpha-1)}{\Gamma\left(\frac{\alpha}{2}\right)^2} \partial_{x_i} \log p_t(\mathbf{x}) \tag{30}$$

where $\Gamma(z) = \int_0^\infty t^{z-1} \exp(-t)\, dt$ is the gamma function. From this approximation of the fractional derivative, by employing the VPSDE parametrisation with $\mathbf{H}_t = \mathbf{G}_t$ and solving the backward equation with the DEIS method, one obtains

$$\begin{gathered} \hat{\mathbf{x}}_{t-\Delta t} = \left(1 + \frac{\beta_t}{\alpha_t}\Delta t\right)\hat{\mathbf{x}}_t + \alpha_t \left(\beta_t \Delta t \frac{1}{h^{\alpha_t-2}} \frac{\Gamma(\alpha_t-1)}{\Gamma\left(\frac{\alpha_t}{2}\right)^2}\right) \nabla_{\mathbf{x}_t} p\left(\hat{\mathbf{x}}_t\right) + \left(\beta_t \Delta t\right)^{1/\alpha_t} \varepsilon \\ \varepsilon \sim \mathcal{S}\alpha_t\mathcal{S}(\mathbf{I}) \end{gathered} \tag{31}$$

where

$$\mathcal{S}\alpha_t\mathcal{S}(\mathbf{I}) \stackrel{\wedge}{=} \left[\mathcal{S}\alpha_t\mathcal{S}(1), \mathcal{S}\alpha_t\mathcal{S}(1), \dots, \mathcal{S}\alpha_t\mathcal{S}(1)\right] \in \mathbb{R}^D \tag{32}$$

This equation is isomorphic to its non-fractional counterpart, with the Wiener process being replaced by the Lévy process, and with the introduction of a fractional exponent $\left(1/\alpha_t\right)$ for the noising parameter β_t. It should be noted that the stability index is now a function of time. As for Eq. (11), the parameter η may be introduced, thus encompassing both Markovian and non-Markovian models. The loss function is similar to its non-fractional (Wiener process) counterpart, with the important distinction that the noise is distributed according to a stable distribution instead of a Gaussian one:

$$\mathcal{L}_{p_L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,T]} \mathbb{E}_{p(\mathbf{x}_0),\, \varepsilon\sim\mathcal{S}\alpha_t\mathcal{S}(\mathbf{I})} \left[\left\|\varepsilon - \varepsilon_\theta\left(\mu_t \mathbf{x}_0 + \sigma \mathbf{I}\, \varepsilon, t\right)\right\|^2\right] \tag{33}$$

where σ is the scale parameter of the Lévy distribution (not to be confused with the standard deviation of the Gaussian distribution). Unfortunately, it is challenging to denoise the large divergent noise generated by the heavy tail [40]. This is in contrast with the Gaussian noise, which does not diverge. However, these large steps are required to explore the solution space fully, which may not be readily achievable with small Gaussian steps [38]. Therefore, with the aim of maximising benefits and minimising drawbacks, the stability index is annealed from a low value (large fluctuations) to $\alpha = 2$ (small fluctuations, Gaussian distribution):

$$\alpha_t = \alpha_{\min} + \frac{t}{T}\left(\alpha_{\max} - \alpha_{\min}\right) \tag{34}$$

with $1 \le \alpha_{\min} < 2$ and $\alpha_{\max} = 2$. Values less than one are not employed because the mean and the variance become infinite [45]. Invariant representations of proteins are addressed in the next section (also see Fig. 4 for the graphical summary of the proposed approach).

5. Protein backbone representation

Proteins' tertiary structures, which include their complex three-dimensional arrangements, are influenced by the interactions of their amino acid side chains along with the backbone. For instance, Fig. 5 shows the backbone of entry 10GS of the Protein Data Bank (PDB), a database of three-dimensional structural data for large biological molecules, such as proteins and nucleic acids [46]. The backbone corresponds to the folding of the protein's amino acid sequence [47,48]. Each amino acid sequence is oriented, begins with the N-terminus, and ends with the C-terminus [47,48]. The folding results from interactions between amino acids and their interaction with the surrounding environment, which is essentially aqueous [47]. The resulting structure is called the native state [47,48]. Proteins' backbones may have any arbitrary orientation. The neural network must learn proteins' backbones irrespective of orientation to generate new proteins. There are essentially two ways to achieve this objective: by employing equivariant neural networks (ENNs) [49] or a rotation- and translation-invariant representation [50,51]. ENNs have an architecture that implements an equivariant map in which the same symmetry group acts on the domain and codomain. For the noise network $\varepsilon_\theta(\mathbf{x}, t)$, this means that if the coordinates are rotated and translated, the generated noise follows the same transformation:

$$\mathbf{R}\varepsilon + \mathbf{t} = \varepsilon_\theta(\mathbf{R}\mathbf{x} + \mathbf{t}, t) \tag{35}$$

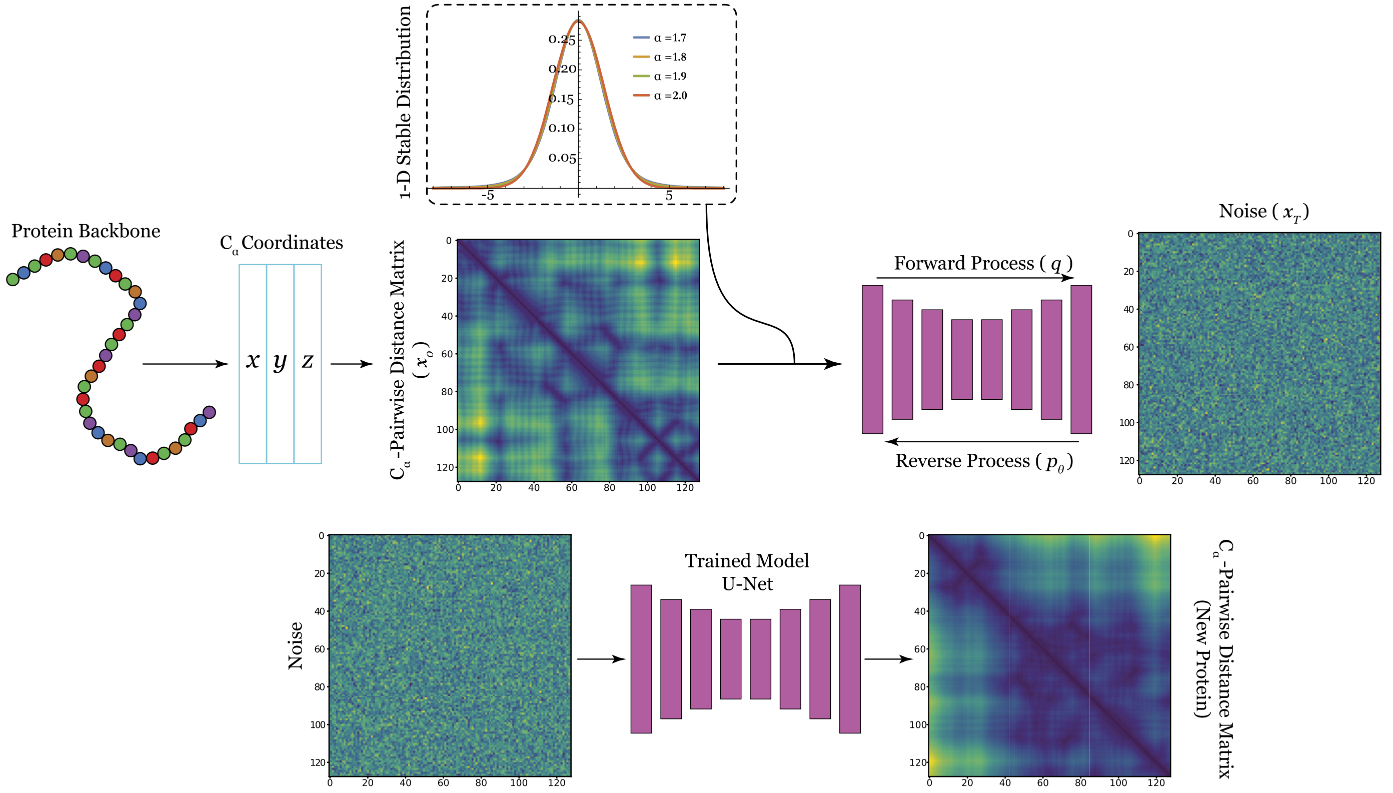

Fig. 4. Graphical summary of the proposed approach.

where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation. This is in contrast with an invariant map in which

$$\varepsilon = \varepsilon_\theta\left(\mathbf{R}\mathbf{x} + \mathbf{t}, t\right) \tag{36}$$

This is the approach that has been adopted in this work. The three-dimensional structure of a backbone may be characterised by the position or coordinates of its α-carbons [47,48], that is, the first carbon atom of each amino acid forming the sequence [47,48]. The α-carbons may be ordered because, as stated earlier, their corresponding amino acid sequence is oriented. An invariant representation is obtained by evaluating the Euclidean distance between each pair of α-carbons [50] (Fig. 6 shows the distance matrix of protein 10GS). The distance matrix is defined as

$$\begin{aligned} \mathbf{D} &= \left[d_{ij}\right] \\ d_{ij} &= \left\|\mathbf{a}_i - \mathbf{a}_j\right\|_2 \end{aligned} \tag{37}$$

where $\left\{\mathbf{a}_i\right\}_{i=1}^{D}$ are the Cartesian coordinates of the α-carbons. This representation is unique because of the amino acid sequence ordering. The invariance comes from the Euclidean distance, which is invariant to translation and rotation. The coordinates of the α-carbons may be retrieved from the distance matrix with various techniques [51], among which the alternative direction method of multipliers (ADMM) is one of the most accurate [52]. This method is a combination of dual ascent with decomposition and the method of multipliers:

$$\begin{aligned} \mathbf{G}_{k+1}, \eta_{k+1} &= \underset{\mathbf{G},\boldsymbol{\eta}}{\arg\min}\; \xi\|\boldsymbol{\eta}\|_1 + \frac{1}{2}\left(\sum_{i=1,j=1}^{m}\left(g_{ii} + g_{jj} - 2g_{ij} + \eta_{ij} - d_{ij}^2\right)^2\right) + \frac{\chi}{2}\left\|\mathbf{G} - \mathbf{Z}_k + \mathbf{U}_k\right\|_2^2 \\ \mathbf{Z}_{k+1} &= \Pi_{S_+^n}\left(\mathbf{G}_{k+1} + \mathbf{U}_k\right) \\ \mathbf{U}_{k+1} &= \mathbf{U}_k + \mathbf{G}_{k+1} - \mathbf{Z}_{k+1} \end{aligned} \tag{38}$$

where $\mathbf{G} = \left[g_{ij}\right] = \left[\left\langle \mathbf{a}_i, \mathbf{a}_j \right\rangle\right]$ is the Gram matrix (containing the inner product of each pair of α-carbon coordinates), $\boldsymbol{\eta}$ is a slack matrix, $\chi > 0$ is an augmented Lagrangian penalty, $\xi \in \mathbb{R}$, $\Pi_{S_+^n}$ is a projection in the symmetric positive semi-definite matrix space, and $\mathbf{U}$ is an auxiliary matrix. After convergence, the coordinates of the α-carbons may be obtained with the singular value decomposition factorisation technique [52].

6. Implementation and methodology

The noise network $\varepsilon_\theta\left(\mathbf{x}, t\right)$ is implemented with a standard U-Net [53] in which, to improve performance, the original ReLU activation function is substituted for a Swish function with hyperparameter $\boldsymbol{\Lambda}$ [54]:

$$\tau\left(x\right) = \frac{x}{1 + \exp\left(-\lambda x\right)} \tag{39}$$

The network comprises a contracting and an expansive path [53]. The contracting path involves multiple convolution kernels, Swish functions, and a maximum pooling for downsampling. At each downsampling step, the number of feature channels is doubled. The expansive path consists of an upsampling, multiple convolutions that halve the number of feature channels, a concatenation with the correspondingly cropped feature map from

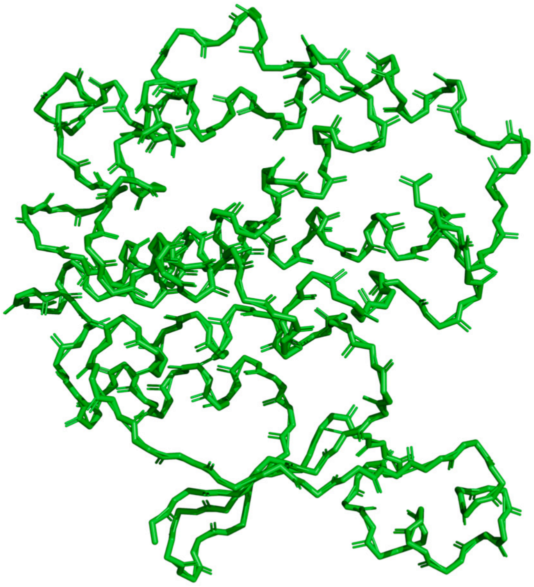

Fig. 5. Backbone of protein 10GS from the PDB

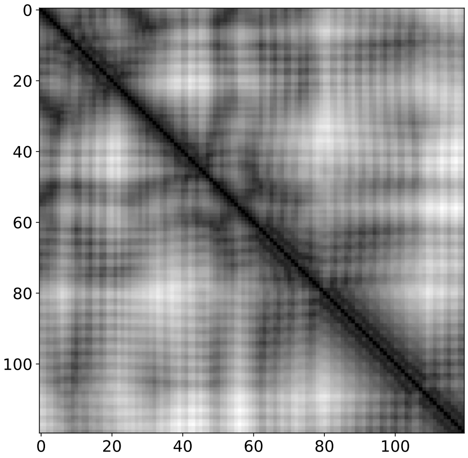

Fig. 6. Distance matrix of protein 10GS from the PDB (first 120 amino acids).

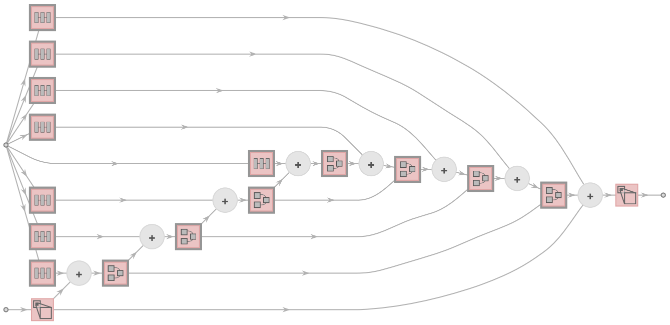

Fig. 7. Outline of the architecture of the U-NetDetailed architecture of the U-Net.

the contracting path, and Swish functions. The network also implements position embedding [32] for time encoding, as described by Eq. (8) below. The neural network architecture is outlined in Fig. 7, with the full details shown in Fig. 8.

The loss function of the noise network, which corresponds to Eq. (33), is optimised with the Adam stochastic optimisation method [31]. This algorithm evaluates and accumulates, for each iteration, the expectations of both the gradient and its second moment. The accumulated expectations are rescaled as time passes. Finally, a correction is applied to the learnable parameters. The stability index of the multidimensional Lévy distribution is annealed according to Eq. (34). The U-Net neural network and the Lévy–Itō diffusion model were implemented in the Mathematica language, version 13.3. The calculations were performed on a computing node with two 16-core Intel Xeon Gold 6130 CPUs clocked at 2.1 GHz, 192 GB of

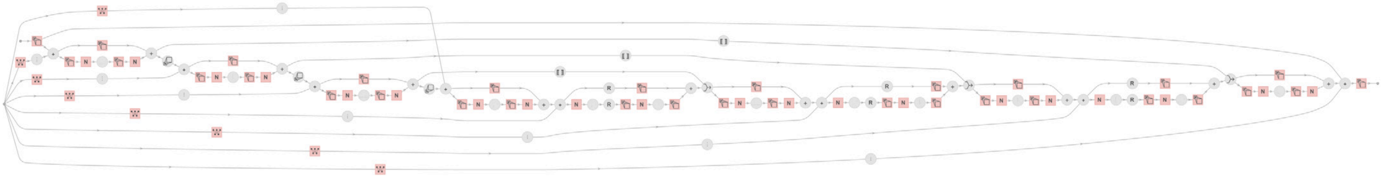

Fig. 8. Detailed architecture of the U-Net.

Table 2
Hyperparameters of the U-Net.

Number of steps used to corrupt the distance matrices	200
Base channel size of the U-Net	32
Encoded time step for the corruption of the distance matrices	16
Depth of the U-Net	3
Mini-batch size for training	1024
Initial noising parameter	0.0001
Final noising parameters	0.02

RAM, and two NVidia V100 GPUs with 32 GB of RAM each. Both GPUs were used concurrently for the calculations. The hyperparameters of the network were determined by grid search and are reported in Table 2.

The results were evaluated with three metrics: the Fréchet distance, the density, and the coverage. The Fréchet distance [13] measures the similarity between the real and generated data distributions:

$$d_F^2(\mathbf{X},\mathbf{Y}) = \left|\boldsymbol{\mu}_{\mathbf{X}} - \boldsymbol{\mu}_{\mathbf{Y}}\right|^2 + \operatorname{tr}\left[\boldsymbol{\Sigma}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{Y}} - 2\left(\boldsymbol{\Sigma}_{\mathbf{X}}\boldsymbol{\Sigma}_{\mathbf{Y}}\right)^{1/2}\right] \in \mathbb{R}^+ \tag{40}$$

where $\boldsymbol{\mu}_{\mathbf{X}}$ and $\boldsymbol{\Sigma}_{\mathbf{X}}$ are the mean and covariance of the real data, while $\boldsymbol{\mu}_{\mathbf{Y}}$ and $\boldsymbol{\Sigma}_{\mathbf{Y}}$ are the mean and covariance of the generated data. The square root is evaluated according to Eq. (24) [41]. The density metric measures the fidelity, the degree to which the generated samples resemble the real ones [14]. It is defined as

$$\rho = \frac{1}{kM}\sum_{j=1}^{M}\sum_{i=1}^{N}\mathbb{1}_{\mathbf{y}_j \in \bigcirc\left(\mathbf{x}_i, \mathrm{NND}_k\left(\mathbf{x}_i\right)\right)}, \quad \rho \geq 0 \tag{41}$$

where $\mathbb{1}$ is a binary (0 or 1) indicator, $\mathrm{NND}_k\left(\mathbf{x}_i\right)$ is the Euclidean distance between $\mathbf{x}_i$ and its k^{th} nearest neighbour, $\bigcirc\left(\mathbf{x}_i r\right)$ is the hypersphere of radius r centred on $\mathbf{x}_i$, $\left\{\mathbf{x}_i\right\}_{i=1}^{N}$ are the real data, and $\left\{\mathbf{y}_j\right\}_{j=1}^{M}$ are the generated data. The density is not bounded: the larger the value, the better the fidelity. The coverage metric, on the other hand, measures the diversity, the degree to which the generated samples cover the full variability of the real samples [14]. It is defined as

$$\kappa = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{\exists\ j\left|\mathbf{y}_j \in \bigcirc\left(\mathbf{x}_i, \mathrm{NND}_k\left(\mathbf{x}_i\right)\right)\right.}, \quad \kappa \in [0,1] \tag{42}$$

Unlike the density, the coverage is bounded and normalised. The density and coverage have been proposed as more robust alternatives to the precision and recall as defined in [55], which tend to overestimate the true manifolds around outliers [14].

7. Experimental results and discussion

The dataset was generated automatically by retrieving 42,082 Homo sapiens proteins from the PDB. For each protein downloaded, the coordinates of the α-carbons were extracted, and the corresponding distance matrices were evaluated for the first 32 amino acids with the help of Eq. (37). During the training phase, the dataset was divided into a training (90%) and a validation set (10%). A total of twenty Lévy–Itō models were trained. The models were parametrised by their stability index annealing schedule α_t, defined by Eq. (34), and by their model hyperparameter η (Markovian versus non-Markovian). For each model, the quality of the generated distance matrices (one thousand generated distance matrices for each model) was assessed with three metrics, namely the Fréchet distance, the density, and the coverage, which are defined by Eq. (40), (41) and (42), respectively. The results are reported in Table 3.

To facilitate graphical comparison, 3-D surface plots for the Fréchet distance, the density and the coverage are provided in Figs. 9, 10 and 11, respectively.

The best results appear in bold. The best Fréchet distance and coverage were obtained with a stability index annealing schedule of 1.9–2 and a model parameter of 0.8. The best density was obtained with a stability index annealing schedule of 1.7–2 and a model parameter of 1. As shown in the table, Markovian models tend to outperform non-Markovian models when Lévy distributions are employed. Pure non-Markovian (DDIM) models ($\eta = 0$) have poor performances irrespective of the stability index annealing schedule. The best performances were obtained for Markovian (DDPM) and quasi-Markovian models ($\eta = 0.8$). The annealed Lévy distribution improves all metrics, particularly the coverage, which attained values as high as 0.98. Compared with the best Gaussian model, which appears in italics, the best Lévy–Itō model improved the Fréchet distance by 25.4%, the density by 35.8%, and the coverage by 11.8%. Therefore, the Lévy–Itō model outperforms the Gaussian model both in the Markovian and non-Markovian regimes.

Table 3
Evaluation of the Lévy–Itō models for various stability index annealing schedules and model hyperparameters.

α_t : $[\alpha_{\min}, \alpha_{\max} = 2]$	η	Fréchet distance	Density	Coverage
1.7	0 (DDIP)	3356.07	151.116	0.480562
1.8		1461.18	722.888	0.731619
1.9		132.6	2220.11	0.87636
1.7	0.2	3002.92	377.008	0.641818
1.8		1303.88	1384.61	0.852051
1.9		115.51	2964.98	0.929732
1.7	0.5	4300.96	3167.42	0.96647
1.8		741.59	4134.47	0.969488
1.9		62.74	4528.85	0.980538
1.7	**0.8**	586.77	5139.79	0.978993
1.8		276.59	4614.55	0.962739
1.9		**29.48**	5393.87	**0.982296**
1.7	**1 (DDPM)**	227.61	**5471.74**	0.974455
1.8		46.41	5129.79	0.951333
1.9		31.74	5145.97	0.923697
2	0	38.82	407.418	0.633382
2	0.2	29.86	2133.49	0.886127
2	0.5	39.53	4029.47	0.878998
2	0.8	39.9	4541.96	0.767193
2	1	40.46	4716.76	0.663181

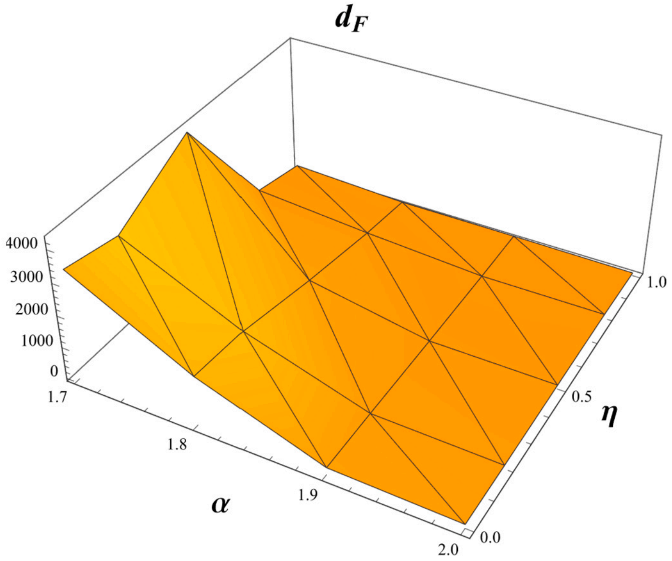

Fig. 9. A 3-D surface plot of the dependence of the Fréchet distance on the stability index and the parameter η. The surface has been interpolated to order three.

8. Conclusions and future work

A new diffusion model for data generation has been proposed based on fractional SDEs, the exponential integrator method, and the annealed stable distribution. The model outperforms Markovian and non-Markovian Gaussian diffusion models when evaluated according to the Fréchet distance, density, and coverage when applied to distance matrices. This improved performance originates from the Lévy distribution. As shown in Fig. 1, which represents Lévy random walks for various values of the stability index, Lévy distributions tend to explore the solution space further than their Gaussian counterparts. This may be explained by their heavy tails, which generate larger steps than normal distributions. These large steps make it possible to reach areas that would otherwise remain inaccessible with a Gaussian random walk [38]. Initially, the stability index is low to promote solution space exploration. Then, it is progressively increased (annealed) to converge to a particular solution. This approach is reminiscent of simulated annealing [56], an optimisation method in which the temperature is initially high to allow the algorithm to escape local minima and is then progressively annealed to lower values to converge to the optimal solution. In future work, the proposed method will be applied to more sophisticated diffusion models, such as the generalised denoising diffusion implicit models [24]. Asymmetrical Lévy distributions and smaller stability indices (heavier tails) could also be employed. Currently, the multidimensional Lévy distribution is diagonal, which means that it is implicitly assumed that there is little or no covariation [57] between the dimensions, covariation being a generalisation of the notion of covariance for stable distributions. Therefore, it is proposed to employ non-diagonal stable distributions [45,58], which are defined as

$$L_{\alpha,\mu}(\mathbf{k}) = \exp\left(-I_{\mathbf{x}}(\mathbf{k})\right) = \mathbb{E}\left[\exp\left(i\langle\mathbf{k},\mathbf{x}\rangle\right)\right] = \exp\left[-i\langle\mu,\mathbf{k}\rangle - \int\limits_{S^d} \psi_\alpha\left(\langle\mathbf{k},\mathbf{s}\rangle\right)\Delta(ds)\right] \tag{43}$$

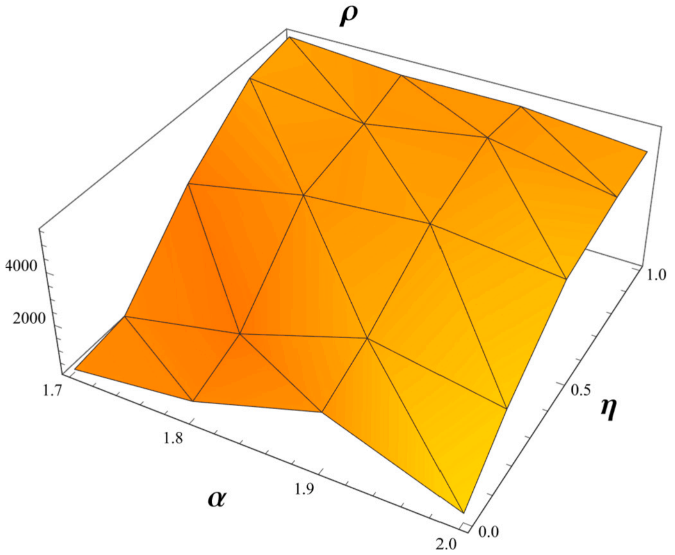

Fig. 10. A 3-D surface plot of the dependence of the density on the stability index and the parameter η. The surface has been interpolated to order three.

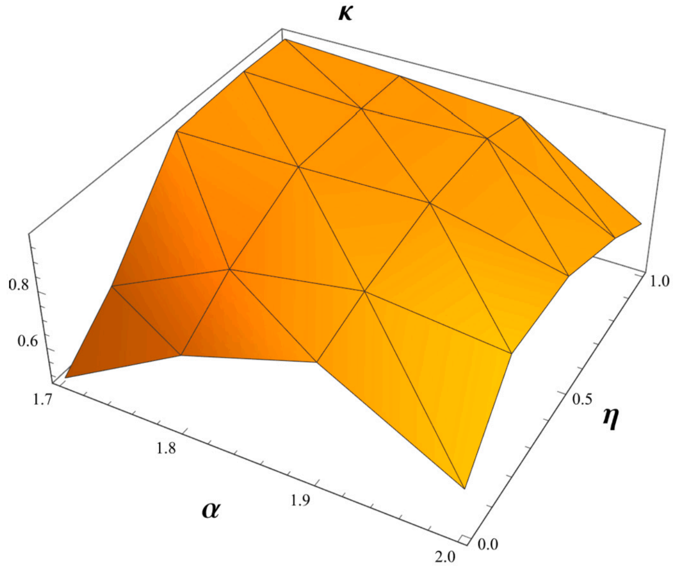

Fig. 11. A 3-D surface plot of the dependence of the coverage on the stability index and the parameter η. The surface has been interpolated to order three.

where $\mathbb{E}$ is the mathematical expectation, $\langle \mathbf{k}, \mathbf{x} \rangle$ is the inner product, and

$$\psi_\alpha(u) \overset{\wedge}{=} \begin{cases} |u|^\alpha \left(1 - i\,\mathrm{sgn}(u) \tan \frac{\pi\alpha}{2}\right) & \text{if } \alpha \neq 1 \\ |u| \left(1 + i \frac{\pi}{2} \mathrm{sgn}(u) \ln |u|\right) & \text{if } \alpha = 1 \end{cases} \tag{44}$$

Unlike the univariate case, only two parameters are required: a stability index α and a localisation vector μ. The information about the scale and the asymmetry is encapsulated in $\Delta(ds)$, which is a measure, or partition, defined on the hypersphere S^D. The application of the multidimensional Lévy distribution for protein generation shall, therefore, be the foundation of our future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the Artificial Intelligence for Design Challenge program from the National Research Council Canada for funding this project.

Appendix A

Table A.4
Definitions of the mathematical symbols used in this work.

Symbol	Definition
$\mathbf{x}_t$	Distance matrix
$\mathbf{F}_t$	Drift matrix
$\mathbf{G}_t$	Diffusion matrix
$\mathbf{w}$	Wiener process
∇_x	Gradient
T	Transpose
$\mathbf{s}_\theta(\mathbf{x}_t, t)$	Score function
$\mathbf{\Lambda}_t$	Metrix (symmetric matrix)
$p(\mathbf{x}_t)$	Data distribution
$p(\mathbf{x}_t \vert \mathbf{x}_0)$	Probability, at time t , of having a noisy datum $\mathbf{x}_t$ given an uncorrupted datum $\mathbf{x}_0$
$\mathcal{N}$	Normal distribution
$\mathbf{L}_t$	Cholesky matrix
θ	Parameters of the neural network
λ_t	Positive weighting function
$\mathcal{U}$	Uniform distribution
$\boldsymbol{\varepsilon}_t(\mathbf{x}_t, t)$	Learned noise
$\boldsymbol{\varepsilon}$	Noise
$\bar{\mathbf{\Lambda}}_t$	$\mathbf{L}_t^{-1}\mathbf{\Lambda}_t\mathbf{L}_t^{-T}$
η	Markovian parametrisation
Δt	Time interval
$\mathbf{\Psi}$	Transition matrix
$\hat{\mathbf{x}}$	Estimate
$\mathbb{R}$	Real numbers
$\widehat{=}$	Equal by definition
$\boldsymbol{\mu}_t$	Mean
$\mathbf{\Sigma}_t$	Covariance matrix
$\sim$	Distributed as
β_t	Noising parameters
$p_L, \mathcal{S}\alpha\mathcal{S}$	Lévy (stable) distribution
α	Stability index
γ	Skewness parameter
μ	Localisation parameter
$\mathbf{H}_t$	Fractional diffusion matrix
$\mathbf{L}_\alpha$	Lévy process with stability index α
$\partial_{\vert x_i \vert}^{\alpha-2}$	Fractional Riesz derivative of order $\alpha - 2$
$\mathbf{U}$	Eigenvectors
λ_i	Eigenvalue
$\mathcal{F}$	Fourier transform
$\mathbf{k}$	Fourier frequencies
$\nabla\cdot$	Divergence
Γ	Gamma function
σ	Standard deviation
$\mathbf{R}, \mathbf{t}$	Rotation matrix and translation vector
$\mathbf{D}$	Distance matrix
$\mathbf{a}_i$	Cartesian coordinates of the α-carbon
$\mathbf{G}$	Gram matrix
$\boldsymbol{\eta}$	Slack matrix
$\mathcal{X}$	Augmented Lagrangian penalty
$\prod_{\mathcal{S}_+^N}$	Projection in the symmetric positive semi-definite matrix space
$\tau(x)$	Swish activation function
d_F	Fréchet distance
ρ	Density
$\mathbb{I}$	Binary indicator
$\bigcirc$	Hypersphere
$\in$	Belongs to
κ	Coverage
$\exists$	There exists
$\mathbb{E}$	Mathematical expectation
$\langle \mathbf{k}, \mathbf{x} \rangle$	Inner product
sgn	Sign function

References

[1] Whitford D. Proteins: structure and function. John Wiley & Sons; 2013.

[2] Dill KA, MacCallum JL. The protein-folding problem, 50 years on. Science 2012;338(6110):1042–6.

[3] Luo Y. Sensing the shape of functional proteins with topology. Nat Comput Sci 2023;3(2):124–5.

[4] Wang L, Wang N, Zhang W, Cheng X, Yan Z, Shao G, et al. Therapeutic peptides: current applications and future directions. Signal Transduct Targeted Ther 2022;7(1):48.

[5] Valastyan JS, Lindquist S. Mechanisms of protein-folding diseases at a glance. Dis. Models Mech. 2014;7(1):9–14.

[6] Dopp JL, Rothstein SM, Mansell TJ, Reuel NF. Rapid prototyping of proteins: mail order gene fragments to assayable proteins within 24 hours. Biotechnol Bioeng 2019;116(3):667–76.

[7] Van Landuyt L, Lonigro C, Meuris L, Callewaert N. Customized protein glycosylation to improve biopharmaceutical function and targeting. Curr Opin Biotechnol 2019;60:17–28.

[8] Gagner JE, Kim W, Chaikof EL. Designing protein-based biomaterials for medical applications. Acta Biomater 2014;10(4):1542–57.

[9] Banavar JR, Giacometti A, Hoang TX, Maritan A, Škrbić T. A geometrical framework for thinking about proteins. Proteins: Struct Funct Bioinform.

[10] Hatfield M, Lovas S. Conformational sampling techniques. Curr Pharm Des 20, https://doi.org/10.2174/13816128113199990603.

[11] Lévy P. Sur les intégrales dont les éléments sont des variables aléatoires indépendantes. Ann Sc Norm Super Pisa, Cl Sci 1934;3(3–4):337–66.

[12] West BJ, Bologna M, Grigolini P, West BJ, Bologna M, Grigolini P. Failure of traditional models. Phys Fractal Oper 2003:37–75.

[13] Dowson D, Landau B. The Fréchet distance between multivariate normal distributions. J Multivar Anal 1982;12(3):450–5.

[14] Naeem MF, Oh SJ, Uh Y, Choi Y, Yoo J. Reliable fidelity and diversity metrics for generative models. In: International conference on machine learning, PMLR; 2020. p. 7176–85.

[15] Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B. Score-based generative modeling through stochastic differential equations. arXiv preprint. arXiv:2011.13456.

[16] Song Y, Durkan C, Murray I, Ermon S. Maximum likelihood training of score-based diffusion models. Adv Neural Inf Process Syst 2021;34:1415–28.

[17] Lee JS, Kim J, Kim PM. Proteinsgm: score-based generative modeling for de novo protein design. BioRxiv. 2022.

[18] Wu KE, Yang KK, Berg Rvd, Zou JY, Lu AX, Amini AP. Protein structure generation via folding diffusion. arXiv preprint. arXiv:2209.15611.

[19] Watson JL, Juergens D, Bennett NR, Trippe BL, Yim J, Eisenach HE, et al. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. BioRxiv. 2022.

[20] Tang X, Dai H, Knight E, Wu F, Li Y, Li T, et al. A survey of generative ai for de novo drug design: new frontiers in molecule and protein generation. arXiv preprint. arXiv:2402.08703.

[21] Guo Z, Liu J, Wang Y, Chen M, Wang D, Xu D, et al. Diffusion models in bioinformatics: a new wave of deep learning revolution in action. arXiv preprint. arXiv:2302.10907.

[22] Ilan Y. Making use of noise in biological systems. Prog Biophys Mol Biol 2023;178:83–90.

[23] Sagarin RD, Taylor T. Natural security: how biological systems use information to adapt in an unpredictable world. Secur Inform 2012;1:1–9.

[24] Zhang Q, Chen Y. Fast sampling of diffusion models with exponential integrator. arXiv preprint. arXiv:2204.13902.

[25] Särkkä S, Solin A. Applied stochastic differential equations, vol. 10. Cambridge University Press; 2019.

[26] Anderson BD. Reverse-time diffusion equation models. Stoch Process Appl 1982;12(3):313–26.

[27] Vincent P. A connection between score matching and denoising autoencoders. Neural Comput 2011;23(7):1661–74.

[28] Jo J, Lee S, Hwang SJ. Score-based generative modeling of graphs via the system of stochastic differential equations. In: International conference on machine learning, PMLR; 2022. p. 10362–83.

[29] Gehring J, Auli M, Grangier D, Yarats D, Dauphin YN. Convolutional sequence to sequence learning. In: International conference on machine learning, PMLR; 2017. p. 1243–52.

[30] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. Adv Neural Inf Process Syst 2020;33:6840–51.

[31] Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint. arXiv:1412.6980.

[32] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Adv Neural Inf Process Syst 30.

[33] Hyvärinen A, Dayan P. Estimation of non-normalized statistical models by score matching. J Mach Learn Res 6(4).

[34] Mannella R. Numerical integration of stochastic differential equations. arXiv preprint. arXiv:cond-mat/9709326.

[35] Dockhorn T, Vahdat A, Kreis K. Score-based generative modeling with critically-damped Langevin diffusion. arXiv preprint. arXiv:2112.07068.

[36] Song J, Meng C, Ermon S. Denoising diffusion implicit models. arXiv preprint. arXiv:2010.02502.

[37] Maddison CJ, Mnih A, Teh YW. The concrete distribution: a continuous relaxation of discrete random variables. arXiv preprint. arXiv:1611.00712.

[38] Paquet E, Viktor HL, Madi K, Wu J. Deformable protein shape classification based on deep learning, and the fractional Fokker–Planck and Kähler–Dirac equations. IEEE Trans Pattern Anal Mach Intell 2022;45(1):391–407.

[39] Applebaum D. Lévy processes and stochastic calculus. Cambridge University Press; 2009.

[40] Yoon E, Park K, Kim J, Lim S. Score-based generative models with Lévy processes. In: NeurIPS 2022 workshop on score-based methods; 2022.

[41] Strang G. Introduction to linear algebra. SIAM; 2022.

[42] Ortigueira MD, et al. Riesz potential operators and inverses via fractional centred derivatives. Int J Math Math Sci 2006.

[43] Şimşekli U. Fractional Langevin Monte Carlo: exploring Lévy driven stochastic differential equations for Markov chain Monte Carlo. In: International conference on machine learning, PMLR; 2017. p. 3200–9.

[44] Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. Geometric deep learning: going beyond Euclidean data. IEEE Signal Process Mag 2017;34(4):18–42.

[45] Samorodnitsky G, Taqqu MS, Linde R. Stable non-Gaussian random processes: stochastic models with infinite variance. Bull Lond Math Soc 1996;28(134):554–5.

[46] Burley SK, Berman HM, Bhikadiya C, Bi C, Chen L, Di Costanzo L, et al. Rcsb protein data bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. Nucleic Acids Res 2019;47(D1):D464–74.

[47] Dill KA, Ozkan SB, Shell MS, Weikl TR. The protein folding problem. Annu Rev Biophys 2008;37:289–316.

[48] Frauenfelder H. The physics of proteins: an introduction to biological physics and molecular biophysics. Springer Science & Business Media; 2010.

[49] Bao F, Zhao M, Hao Z, Li P, Li C, Zhu J. Equivariant energy-guided sde for inverse molecular design. arXiv preprint. arXiv:2209.15408.

[50] Hoffmann M, Noé F. Generating valid Euclidean distance matrices. arXiv preprint. arXiv:1910.03131.

[51] Kloczkowski A, Jernigan RL, Wu Z, Song G, Yang L, Kolinski A, et al. Distance matrix-based approach to protein structure prediction. J Struct Funct Genomics 2009;10(1):67–81.

[52] Anand N, Huang P. Generative modeling for protein structures. Adv Neural Inf Process Syst 31.

[53] Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference. Proceedings, part III, vol. 18. Springer; 2015. p. 234–41.

[54] Ramachandran P, Zoph B, Le QV. Searching for activation functions. arXiv preprint. arXiv:1710.05941.

[55] Kynkäänniemi T, Karras T, Laine S, Lehtinen J, Aila T. Improved precision and recall metric for assessing generative models. Adv Neural Inf Process Syst 32.

[56] Guilmeau T, Chouzenoux E, Elvira V. Simulated annealing: a review and a new scheme. In: 2021 IEEE statistical signal processing workshop (SSP). IEEE; 2021. p. 101–5.

[57] Cheng B, Rachev ST. Multivariate stable futures prices. Math Finance 1995;5(2):133–53.

[58] Paquet E, Viktor HL, Guo H. Learning in the presence of large fluctuations: a study of aggregation and correlation. In: New frontiers in mining complex patterns: first international workshop, NFMCP 2012, held in conjunction with ECML/PKDD 2012. Revised selected papers, vol. 1. Springer; 2013. p. 49–63.