nature reviews bioengineering

https://doi.org/10.1038/s44222-023-00114-9

Review article

Check for updates

Diffusion models in bioinformatics and computational biology

Zhiye Guo [1,2,3], Jian Liu[1,2,3], Yanli Wang[1,2,3], Mengrui Chen[1,2,3], Duolin Wang[1,2], Dong Xu[1,2] & Jianlin Cheng [1,2]

Abstract

Denoising diffusion models embody a type of generative artificial intelligence that can be applied in computer vision, natural language processing and bioinformatics. In this Review, we introduce the key concepts and theoretical foundations of three diffusion modelling frameworks (denoising diffusion probabilistic models, noise-conditioned scoring networks and score stochastic differential equations). We then explore their applications in bioinformatics and computational biology, including protein design and generation, drug and small-molecule design, protein–ligand interaction modelling, cryo-electron microscopy image data analysis and single-cell data analysis. Finally, we highlight open-source diffusion model tools and consider the future applications of diffusion models in bioinformatics.

Sections

[1]Department of Electrical Engineering and Computer Science, University of Missouri, Columbia, MO, USA. [2]NextGen Precision Health, University of Missouri, Columbia, MO, USA. [3]These authors equally contributed to this work: Zhiye Guo, Jian Liu, Yanli Wang, Mengrui Chen. e-mail: chengji@missouri.edu

Key points

- Diffusion models are a generative artificial intelligence technology that can be applied in natural language processing, image synthesis and bioinformatics.
- Diffusion models have contributed greatly to computational protein design and generation, drug and small-molecule design, protein–ligand interaction modelling, cryo-electron microscopy data enhancement and single-cell data analysis.
- Many diffusion models are also available as open-source tools.
- Although diffusion models may potentially outperform other generative approaches, such as generative adversarial networks and variational auto-encoders, their computational resource requirements remain high.

Introduction

Deep learning[1] was introduced to the field of bioinformatics and computational biology in 2012 (ref. 2) (Box 1) and has been applied to many bioinformatics problems, such as protein structure prediction[3], protein function prediction[4–9], protein–ligand interaction prediction[10–14], gene-expression prediction[15–20] and gene regulatory network modelling[21–25]. Various deep learning architectures, including convolutional neural networks[26], long short-term memory networks[27], residual networks[28], generative adversarial networks (GAN)[29], graph neural networks (GNN)[30] (Box 2) and transformers[31] have been developed for bioinformatics data analysis.

Diffusion models leverage deep learning technology[32–35]; however, they outperform other deep learning methods in many domains, including in image generation[36–42], image inpainting[43,44] and speech synthesis[45]. Diffusion models are deep learning-based generative models[32–35] (Box 2) that aim to generate artificial yet realistic data (for example, a computer-generated Picasso painting or an answer to a user's question) from input parameters. Compared to other generative models, such as autoregressive models[46], normalizing flows[47], energy-based models[48], variational auto-encoders (VAEs)[49] or GANs[29], diffusion-based generative models have the ability to learn complex distributions, handle high-dimensional data and generate diverse data[50–55]. In particular, diffusion models can surpass GANs[29], which consist of a generator that generates data and a discriminator that can differentiate the generated data, in the challenging task of image synthesis[33,50]. In addition, diffusion models can be applied for computer vision[43,51,56–71], natural language processing[55,72–75], temporal data modelling[76–81], multi-modal modelling[36,37,82,83], and in medical image reconstruction[84–93].

Diffusion models were originally introduced[32] to address a central problem in machine learning, that of modelling complex datasets using highly flexible families of probability distributions while ensuring that learning, sampling, inference and evaluation remain analytically or computationally tractable (Fig. 1). Inspired by non-equilibrium statistical physics, this approach systematically and slowly destroys the structure of data through an iterative forward diffusion process. Then, a reverse diffusion process is applied to restore the structure in the data, yielding a highly flexible and tractable generative model of the data, thereby enabling rapid learning, data sampling and evaluating probabilities through deep generative models with up to thousands of layers or time steps as well as the computing of conditional and posterior probabilities under the learned model. Based on this concept, denoising diffusion probabilistic models (DDPMs)[33] can achieve performance comparable to or better than other generative models (for example, decoder, energy-based models and GANs)[46,94–96] in image generation tasks. The diffusion network structure and training strategy can further be improved to boost performance[50], surpassing GANs in image synthesis. For example, a multi-head attention mechanism and the BigGAN's residual module[95] can be applied for up-sampling and down-sampling of data to improve the resolution and quality of generated images. In addition, a denoising diffusion implicit model (DDIM)[97] can be used to increase sampling rate.

Importantly, diffusion models can be applied in bioinformatics, for example, for denoising cryo-electron microscopy (cryo-EM) data[98], single-cell gene-expression analysis[99,100], protein design and generation[84,91,101–107], drug and small-molecule design[54,108–113] and protein–ligand interaction modelling[114–118]. Diffusion models have the advantage of being able to handle high-dimensional data with high diversity and scalability.

In this Review, we provide a detailed survey of diffusion models, including denoising diffusion models, noise-conditioned score networks (NCSNs) and stochastic differential equations (SDEs), and discuss their applications in bioinformatics. We further highlight possible future developments of diffusion models, aiming to propose some challenging bioinformatics problems that may be tackled by creative diffusion models.

The concept of diffusion models

Diffusion models learn to reverse the process of data destruction or corruption (for example, introduced by noise), allowing the generation of realistic, clean data samples (for example, restoration of uncorrupted data). Thus, diffusion models can learn from data that has been progressively destroyed or degraded to generate new samples from a given distribution or to estimate the distribution from which a given sample is drawn (Box 2).

Diffusion models are based mainly on three frameworks, each with a different formulation of the forward and reverse processes (Fig. 2), that is, DDPMs[32,33], NCSNs[34,119] and score SDEs[35,120].

Denoising diffusion probabilistic models

DDPMs, which were the first diffusion models able to generate high-resolution data, typically contain two Markov chains (Box 2): the forward chain gradually adds noise to scramble the original data, followed by a reverse chain that removes the noise from the data to recover the original data. If $q(x_0)$ denotes the distribution of the original data, in which x_0 denotes uncorrupted data, the transition kernel $q(x_t|x_{t-1})$ of the forward Markov process adding Gaussian perturbation at time t is denoted $\mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I})$, in which $t\epsilon\{1, \ldots, T\}$. Here T represents the number of diffusion steps; $\beta_t\epsilon[0,1)$ is the hyperparameter denoting the variance schedule across diffusion steps; $\mathbf{I}$ is the identity matrix; and $N(x;\mu,\sigma)$ is the normal distribution of x with mean μ and covariance σ. If $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t}\alpha_s$, a noisy sample x_t can be obtained directly from the distribution conditioned on the original input x_0:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \quad (1)$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}) \quad (2)$$

The forward process gradually introduces noise into the original data until it is completely replaced by noise. The reverse process is the opposite operation, resulting in the generation of new samples. This process typically starts with unstructured noise obeying the prior distribution, and then, by applying a model — typically a trainable neural network — that has learning ability, noise is removed step by step to restore the original data. The neural network N can be formulated as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t)) \tag{3}$$

Given the starting point data of the reverse process as $p(X_T) = \mathcal{N}(X_T; 0, \mathrm{I})$, the distribution of X_0 conditioned on X_T is given by:

$$p_\theta(X_{0:T}) = p(X_T) \prod_{t=1}^{T} p_\theta(X_{t-1}|X_t) \tag{4}$$

Eventually, a marginal distribution of X_0 close to the original data x_0 can be obtained by $p_\theta(x_0) = \int p_\theta(x_{0:T}) \mathrm{d}x_{1:T}$.

To train the model parameterized with θ so that it can learn the pattern of the original data and make $p(x_0)$ close to the true data distribution $q(x_0)$, the loss function to be minimized is set as the negative log-likelihood (equation (5)). We note that the process of minimizing the negative log-likelihood of the observed data under the model is equivalent to minimizing the Kullback–Leibler (KL) divergence between the empirical distribution defined by the original data $q(x_0, x_1, \cdots, x_T)$ and the model distribution $p_\theta(x_0, x_1, \cdots, x_T)$:

$$\begin{aligned} \mathbb{E}[-\log p_\theta(x_0)] &\leq \mathrm{KL}(q(x_0, x_1, \cdots, x_T) \| p_\theta(x_0, x_1, \cdots, x_T)) \\ &= \mathbb{E}_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_T)}\right] \\ &= \mathbb{E}_q\left[-\log p(x_T) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] \\ &= -L_{\mathrm{VLB}} \end{aligned} \tag{5}$$

The objective of DDPM training is to minimize L_{VLB}, also known as the variational lower bound of the log-likelihood. L_{VLB} can also be parameterized to increase the quality of sample generation[33].

Noise-conditioned score networks

In NCSNs, the score function of a probability density function $p(x)$ is represented by the gradient of the log density with respect to the input as $\nabla_x \log p(x)$. To learn and estimate the score function, a score-matching neural network s_θ is trained. The goal of this neural network is to make $s_\theta(x) \approx \nabla_x \log p(x)$. Therefore, the objective function of the scoring network can be defined as:

$$\mathbb{E}_{x\sim p(x)} \| s_\theta(x) - \nabla_x \log p(x) \|_2^2 \tag{6}$$

Even though the problem is well defined, optimizing equation (6) is numerically impossible because the value of $\nabla_x \log p(x)$ cannot be known. However, score functions can be learned from data by applying score matching[121], denoising score matching[122–124] or sliced score matching[125].

Moreover, training remains difficult because the trained score functions are unreliable in low-dimensional manifold, because low-dimensional data is typically embedded in a high-dimensional space (the manifold hypothesis)[34]. This challenge can be addressed by introducing Gaussian noise to the data at various scales, which improves the data distribution's suitability for score-based generative modelling. Thus, a single NCSN can be applied to estimate the score corresponding to each noise level. If $0 < \sigma_1 < \sigma_2 < \ldots < \sigma_t < \ldots < \sigma_T$ is a sequence of Gaussian noise levels, $p_{\sigma_t}(x_t|x) = \mathcal{N}(x_t; x, \sigma_t^2 \mathbf{I})$, $p_{\sigma_t}(x) \approx p(x_0)$, and $p_{\sigma_T}(x) \approx \mathcal{N}(0, \mathbf{I})$. The NCSN $s_\theta(x, \sigma_t)$ with the denoising score matching can then approximate the gradient log density function, making $s_\theta(x, \sigma_t) \approx \nabla_x \log(p_{\sigma_t}(x))$, $\forall\, t \in \{1, \ldots, T\}$. And for x_t, $\nabla_x \log(p_{\sigma_t}(x))$ is derived as:

$$\nabla_{x_t} \log p_{\sigma_t}(x_t|x) = -\frac{x_t - x}{\sigma_t} \tag{7}$$

Consequently, the optimization objective function in equation (6) can be transformed into:

$$\frac{1}{T} \sum_{t=1}^{T} \lambda(\sigma_t) \mathbb{E}_{p(x)} \mathbb{E}_{x_t \sim p_t(x_t|x)} \| s_\theta(x_t, \sigma_t) + \frac{x_t - x}{\sigma_t} \|_2^2 \tag{8}$$

in which $\lambda(\sigma_t)$ is a weighting function.

During the sampling phase, NCSNs use the annealed Langevin dynamics algorithm, which employs a Markov Chain Monte Carlo procedure (Box 2) to sample from a distribution according to its score function $\nabla_x \log p(x)$. The Langevin method recursively computes x_i as follows:

$$x_i = x_{i-1} + \frac{\gamma}{2} \nabla_x \log p(x) + \sqrt{\gamma}\, \omega_i \tag{9}$$

Box 1

Deep learning

Deep learning is a machine learning technology that applies artificial neural networks with many layers of neurons (hence, 'deep') to model and extract complex patterns in data. Deep learning can then learn patterns and features from complex data to perform intelligent tasks, such as speech and image recognition, natural language processing and protein structure prediction. The artificial neurons in each layer receive input from the neurons in the previous layers until the final output layer produces a prediction (for example, classifying an image into a category or generating a sentence of text). During training of a deep learning model, the weights associated with the connections between neurons are adjusted to fit the training data. A major advantage of deep learning models over other machine learning methods is their ability to automatically learn hierarchical representations from raw data through multiple layers of abstraction. This enables deep learning models to achieve high prediction accuracy in many domains, such as precision medicine and healthcare (for example, medical image segmentation[237,258–261] and disease diagnosis[262–265]), finance (for example, algorithmic trading[266,267] and risk management[268]) and agriculture (for example, crop monitoring[269,270] and pest detection[271]). Some notable applications of deep learning are ChatGPT[272] for natural language processing, DALL-E-2 (ref. 83) and GLIDE[273] for image generation, and AlphaFold2 (ref. 163) for protein structure prediction.

Box 2

Key concepts relevant to diffusion models

Diffusion: the movement of molecules, atoms, ions or energy from a region of higher concentration to a region of lower concentration along a concentration gradient until the concentration becomes equal in both regions. Diffusion, which is driven by a gradient in Gibbs free energy or chemical potential, is a stochastic process owing to the inherent randomness in the movement of the diffusing entities.

Generative model: a type of machine learning model that aims at learning the underlying distribution of data to generate new, similar data. These models can approximate the joint probability distribution of input features and labels, if available, and generate new data points by sampling from the learned distribution.

Markov chain: a stochastic model that describes a sequence of possible states, in which the probability of a state depends (or is conditioned) only on its previous state.

Markov chain Monte Carlo: a statistical or computational simulation method that constructs a Markov chain to iteratively generate a sequence of samples according to a conditional probability distribution between two consecutive states. After running the Markov chain for enough iterations, the generated samples converge to the desired posterior distribution.

Graph neural network (GNN)[30]: a type of deep learning model for processing graph-structured data (for example, molecular graphs and biological networks). Each node in a GNN receives messages from its neighbouring nodes, which are used to update its hidden representation. By iteratively updating node representations, the GNN can aggregate information from both the local neighbourhood and remotely connected nodes in the graph.

Equivariant GNN[161]: a special type of GNN that is equivariant to a transformation (for example, translation and rotation) in the input data (for example, of a three-dimensional object, such as a protein structure). For example, the translation of an object in the input space leads to the translation of the same output of the object generated by the equivariant GNN in the output space without changing the value of the output.

SE(3)-equivariant networks[162]: a special equivariant GNN model that preserves the symmetry of the special Euclidean group SE(3). If a SE(3) transformation is applied to the input, the output generated by the networks undergoes an equivalent transformation. Achieving SE(3) equivariance allows the model to capture the inherent symmetries and geometric properties of the input 3D data.

SE(3)-transformer[198]: a specific implementation of SE(3)-equivariant networks using the transformer's self-attention mechanism to achieve SE(3) symmetry, including three-dimensional rotations and translations. The SE(3)-transformer is particularly useful for tasks involving three-dimensional structures, such as protein structure prediction and protein design, where different (x, y, z) coordinates of the same protein structure appearing in different orientations and positions can be treated as the exact same object.

where γ determines the amplitude of the update in the score's direction; x_0 is sampled from the prior distribution; and the noise is drawn according to $\omega_i \sim \mathcal{N}(0, \mathbf{I})$.

NCSNs and DDPMs both operate on the principle of converting a basic noise distribution into a more intricate data distribution by collecting information during the introduction of noise, which is then reapplied when removing the noise. Both models are trained to tackle a noise regression problem, based on the principle of maximum likelihood estimation. Notably, the objective formulation of score matching with Langevin dynamics in NCSN aligns with that of the re-weighted variant of the evidence lower bound of DDPM[35,126,127]. In terms of sample generation, both models employ ancestral sampling, which progressively transforms a noise sample into a data sample, guided by data distribution gradients.

Score stochastic differential equations

With unlimited time steps or noise levels, DDPMs and NCSNs can be further generalized to a situation in which the perturbation and denoising processes can be described as SDEs. This generalized approach[35] of gradually transforming data into noise is called score SDE. The forward process of score SDE uses SDEs and requires an estimated score function of the noisy data distribution. It is equivalent to the Itô SDE[128] solution, which consists of a drift component for mean transformation and a diffusion coefficient for describing noise:

$$\mathrm{d}x = f(x,t)\,\mathrm{d}t + g(t)\,\mathrm{d}w,\ t \in [0,T] \tag{10}$$

where w represents the standard Wiener process known as Brownian motion, and $f(x,t)$ and $g(t)$ are the drift and diffusion coefficients of SDE, respectively. The forward process in DDPMs and score-based generative models is a special case of the discretizational SDE.

The formulation of the reverse diffusion process of SDE is given by equation (11)[129], also called reverse-time SDE:

$$\mathrm{d}x = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]\,\mathrm{d}t + g(t)\,\mathrm{d}\overline{w} \tag{11}$$

where $\overline{w}$ is the standard Brownian motion running backward time, and $\mathrm{d}t$ represents the infinitesimal negative time step. The reverse SDE and forward SDE share the same marginal densities but in the opposite time direction[35]. As in DDPMs and NCSNs, to numerically solve reverse-time SDE, a trainable neural network $s_\theta(x,t)$ is employed to estimate the actual score function $\nabla_x \log p_t(x)$. The objective function can be defined as:

$$\mathbb{E}_{x(t)\sim p(x(t)|x(0)),x(0)\sim p_{\mathrm{data}}}\left[\frac{\lambda(t)}{2}\left\|s_\theta(x(t),t) - \nabla_{x(t)}\log p_t(x(t)|x(0))\right\|_2^2\right] \tag{12}$$

where $t \sim \mathcal{U}([0, T])$ denotes the uniform distribution over $[0, T]$ and λ is a weighting function. In addition, several sampling techniques, such as the predictor–corrector sampler, can be employed to generate good samples. This procedure uses a score-based method (that is, annealed Langevin dynamics) as a corrector after using a numerical approach to sample data from the reverse-time SDE.

Improving diffusion models

The aforementioned diffusion models can be further improved through extension in training speed[126,130–133], increasing data sampling (data generation) speed[97,134–139], integration with other neural networks[38,120,140–142], and applications to different data types[53,73,143–151]. Many of these improvement strategies are available as open-source tools[152] (Table 1), which has opened up their application to a diverse range of bioinformatics problems (Box 3). Importantly, diffusion models can handle different data types, such as one-dimensional (1D) DNA and protein sequences, two-dimensional (2D) biomedical images, three-dimensional (3D) protein structures and vectorized gene-expression data.

Protein design and generation

The computational generation of new, physically foldable protein structures allows the design of proteins with specific functions or structural properties for protein engineering and drug discovery. However, deep generative models (Box 2), such as VAEs and GANs[153–159], are limited to generating only small proteins or domains of large proteins (for example, of immunoglobulins). Alternatively, diffusion models can be applied to protein design and generation, because large and diverse proteins can be generated by guiding the model at each step of the iterative generation process.

Protein structures in protein generation[153,154] are typically described by a 2D matrix (map) that contains the pairwise distances and angles between all the residues in the protein. For example, ProteinSGM, based on a score-based generative model[91], applies a diffusion model of 2D image generation using such a representation to create protein structures: a score-based generation diffusion model with SDEs is used to generate a series of 2D matrices that include inter-residue pairwise distances d, and the ω, θ and φ angles between two residues. These constraints are then fed into Rosetta[160] to build native-like protein structures. For unconditional protein structure generation, ProteinSGM can generate proteins from random noise. For conditional protein structure generation, such as scaffold inpainting and functional site inpainting, the tool can generate protein structures that satisfy user-defined constraints, similar to solving an image inpainting problem. However, ProteinSGM requires post-processing by Rosetta using Markov Chain Monte Carlo (Box 2), which makes the prediction computationally expensive.

Unlike ProteinSGM, Foldingdiff[101] represents the protein backbone structures (only N–Cα–C atoms for each residue) with a series of consecutive angles to capture the relative orientation of the constituent atom acid residues. A simple language transformer model[31] with DDPM can then be applied to generate protein structures unconditionally, as the angles are invariant to translation and rotation. However, using a transformer to predict sequence-like consecutive angles has the drawback that errors from the early prediction accumulate and considerably affect the final structure, including collisions between atoms. In addition, the approach cannot be generalized to generate complex structures with more than one chain.

Inspired by Foldingdiff, DiffSDS[102] introduces a 1D directional representation derived from invariant atom features, similar to torsion angle representation, which enables an encoder–decoder language model to perform the diffusion process. In the language model, the encoder (with a hidden atom-direction-space layer) transforms the invariant features into equivalent direction vectors, whereas the decoder reverses the transformation. By performing the diffusion process in this direction and by conditioning angle spaces on geometric restraints, DiffSDS can restore protein backbone structures of higher quality than the deep-learning-based protein design method RFDesign[156]: DiffSDS is two times better at generating proteins that resemble natural proteins (protein likeness), as measured by Rosetta energies, about 18 times better in terms of connectivity errors and 60% better at generating non-overlapping scores with existing backbones than RFDesign.

The integration of diffusion models with GNNs[30] (Box 2) enables the direct generation of 3D protein coordinates, resulting in an end-to-end generative model. SE(3)-equivariant[161,162] (Box 2) DDPMs, which are usually used in small-molecule generation, can also be applied to generate protein structures in a representation-frame-independent manner[163]. For example, independent DDPM models equipped with invariant point attention[163] structural modules can be trained with the distribution of atom features (for example, coordinates in a canonical frame with respect to backbone atoms, residue type and side-chain angles) to generate a protein's backbone, sequence and side-chain rotamers[84]. By jointly diffusing the structure and sequence, while incorporating coarse structural constraints, the model can gradually generate the fully atomistic protein structure and sequence, allowing controllable protein backbone generation and protein structure inpainting. The sequence recovery rate of this method is comparable to that of other machine-learning-based and physics-based methods, such as 3DConv[164], RosettaFixBB and RosettaReIBB[165]. Similarly, Genie[103] makes use of the SE(3)-equivariant feature from the invariant point attention module in conjunction with DDPM to generate protein backbones unconditionally, also introducing geometric asymmetry with an invariant encoder to directly inject noise into residue coordinates, as well as an SE(3) equivariant decoder with an invariant point attention module to predict noise.

SMCDiff[104] applies a similar deep learning architecture (that is, an SE(3)-equivariant GNN) (Box 2) to the motif-scaffolding generation problem, dividing the problem into two parts: unconditional protein backbone generation (ProtDiff) and conditional sampling in diffusion models based on a protein motif (SMCDiff), similar to inpainting. Unconditional protein generation is achieved by training a SE(3)-equivariant GNN (Box 2), built from residue coordinates and embedded features from the protein sequence, to generate protein backbones. By contrast, conditional sampling is formulated on an unconditional diffusion model as a sequential Monte Carlo simulation problem, which may be solved by particle filtering. However, the network does not include torsion angles as features and may therefore generate unnatural proteins (for example, left-handed helices). SMCDiff was the first deep generative model that leveraged the power of diffusion models to address the motif-scaffolding generation problem.

RFdiffusion[105], which integrates a conditional DDPM diffusion model with the pre-trained protein 3D structure prediction model RoseTTAFold[166], can directly generate final 3D coordinates. Inspired by the recycling process in AlphaFold2, a self-conditioning prediction strategy is applied, in which the current prediction is conditioned on the prediction from the previous timesteps, thereby considerably improving the performance of the model. Starting from random noise,

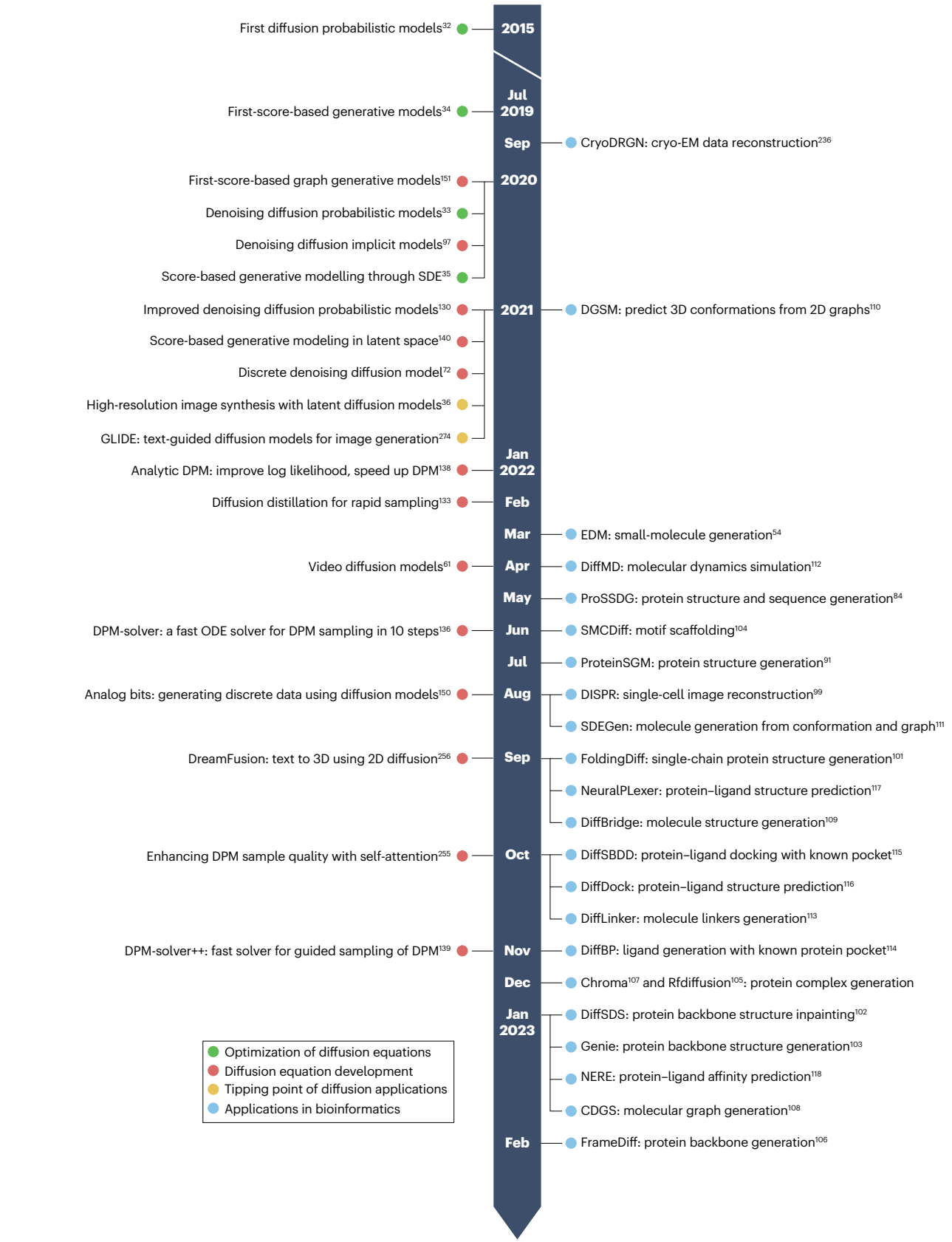
2015
First diffusion probabilistic models[32]
Jul 2019
First-score-based generative models[34]
Sep
CryoDRGN: cryo-EM data reconstruction[236]
2020
First-score-based graph generative models[151]
Denoising diffusion probabilistic models[33]
Denoising diffusion implicit models[97]
Score-based generative modelling through SDE[35]
2021
Improved denoising diffusion probabilistic models[130]
DGSM: predict 3D conformations from 2D graphs[110]
Score-based generative modeling in latent space[140]
Discrete denoising diffusion model[72]
High-resolution image synthesis with latent diffusion models[36]
GLIDE: text-guided diffusion models for image generation[274]
Jan 2022
Analytic DPM: improve log likelihood, speed up DPM[138]
Feb
Diffusion distillation for rapid sampling[133]
Mar
EDM: small-molecule generation[54]
Apr
Video diffusion models[61]
DiffMD: molecular dynamics simulation[112]
May
ProSSDG: protein structure and sequence generation[84]
Jun
DPM-solver: a fast ODE solver for DPM sampling in 10 steps[136]
SMCDiff: motif scaffolding[104]
Jul
ProteinSGM: protein structure generation[91]
Aug
Analog bits: generating discrete data using diffusion models[150]
DISPR: single-cell image reconstruction[99]
SDEGen: molecule generation from conformation and graph[111]
Sep
DreamFusion: text to 3D using 2D diffusion[256]
FoldingDiff: single-chain protein structure generation[101]
NeuralPLexer: protein–ligand structure prediction[117]
DiffBridge: molecule structure generation[109]
Oct
Enhancing DPM sample quality with self-attention[255]
DiffSBDD: protein–ligand docking with known pocket[115]
DiffDock: protein–ligand structure prediction[116]
DiffLinker: molecule linkers generation[113]
Nov
DPM-solver++: fast solver for guided sampling of DPM[139]
DiffBP: ligand generation with known protein pocket[114]
Dec
Chroma[107] and Rfdiffusion[105]: protein complex generation
Jan 2023
DiffSDS: protein backbone structure inpainting[102]
Genie: protein backbone structure generation[103]
NERE: protein–ligand affinity prediction[118]
CDGS: molecular graph generation[108]
Feb
FrameDiff: protein backbone generation[106]
Optimization of diffusion equations
Diffusion equation development
Tipping point of diffusion applications
Applications in bioinformatics

Fig. 1 | Timeline of advances in diffusion models and their applications in bioinformatics. Data are taken from refs. 32–36,54,61,72,84,91,97,99, 101–118,130,133,136,138–140,150,151,256,257,273. Cryo-EM, cryogenic electron microscopy; SDE, stochastic differential equations; DPM, diffusion probabilistic model; ODE, ordinary differential equations.

RFdiffusion can generate large protein structures unconditionally, which can then be used in the design of protein monomers. Using protein motif coordinates as input, RFdiffusion can also construct scaffolds conditionally for functional motif and enzyme active site scaffolding[105]. Given a point group symmetry, RFdiffusion can maintain the symmetry during the prediction owing to the equivariance design of RoseTTAFold. Therefore, this approach can be applied to symmetric protein oligomer and motif scaffolding (for example, for the design of therapeutic[167] and metal-binding proteins[168,169]). We note that compared to the other methods discussed in this section, some proteins designed by RFdiffusion have not only been validated in silico, but also by biochemical and biophysical experiments[170,171], making it one of the first generative artificial intelligence methods of protein design that have been experimentally validated. Furthermore, RFdiffusion outperforms other methods, such as RFDesign, in the design of large protein structures and high-order protein oligomers, demonstrating the advantage of diffusion models.

FrameDiff[106] applies diffusion models to explore whether a pre-trained protein structure predictor is necessary for protein backbone generation. Here, using denoising score matching, a principled SE(3) diffusion model can better formulate the protein backbone generation problem, achieving comparable performance with four-fold fewer network weights and without the need to train another protein structure prediction network, compared to RFdiffusion.

Chroma[107] is a GNN[30]-based conditional diffusion model designed to generate large single-chain proteins and protein complexes with desired properties and functions. This model can generate protein structures that are over 3,000 residues in size, which surpasses the size limit for proteins generated by several other networks (that is, ProteinSGM, Foldingdiff, DiffSDS and SMCDiff) (<2,000 residues). To reduce computational complexity, Chroma uses a random graph generation procedure that preserves both short- and long-range interactions. As a result, Chroma can produce high-quality, diverse new protein structures, and enables the programmable generation of proteins that are conditioned on several different properties, such as residue–residue distances, symmetry and shape.

Small-molecule generation and drug design

Drug discovery involves the identification and optimization of small molecules that can interact with specific biological targets, such as enzymes or receptors, to modulate their activity and ultimately achieve a therapeutic effect. Deep learning, particularly deep generative models, enables the rapid generation and evaluation of a large number of such potential drug candidates[172–175].

The conditional diffusion model, which is a deep learning method based on discrete graph structures (CDGS), allows the generation of molecular graphs of small molecules with similar data distributions to real-number molecular graphs[108]. This method employs a hybrid message-passing block architecture, which comprises a standard message-passing layer for collecting local features, such as node-edge dependencies, and an attention-based message-passing layer for extracting and transmitting global information in the architecture. The molecular graphs are embedded with distinct components for node features and edge matrices, with channels for edge existence and edge types. The CDGS model has enabled the application of diffusion models in the molecular graph domain, which is crucial for drug discovery and material science. This approach accurately models the complex dependency between graph structures and features during the generative process, using SDEs to describe the graph diffusion process. The continuous forward process is applied directly to edge existence variables, and the reverse process first decodes discrete graph structures, which serve as the condition for each sampling step. A specialized hybrid graph noise prediction model is used to extract global and local node-edge dependencies from intermediate graph states. This diffusion-based model can obtain high-fidelity samples in 200 steps of network evaluations using the Euler–Maruyama method[176]. In addition, a fast ordinary differential equation solver, which applies the semi-linear structure of probability flow ordinary differential equations for graphs, promotes rapid, high-quality graph sampling. CDGS outperforms other methods in molecular graph generation, including flow-based methods (for example, GraphAF[177], GraphDF[178], MoFlow[179] and GraphCNF[180]) and other diffusion models (EDP-GNN[151], GraphEBM[181] and GDSS[148]). CDGS also performs better in generic graph generation than ER[182], VGAE[183], GpraphRNN[184] and GRAN[185], demonstrating its potential to facilitate drug discovery and material design by representing molecular structures and restricting the molecule search space.

The E(3)-equivariant diffusion model (EDM)[54] (Box 2) performs the diffusion process on atom coordinates and atom types in the Euclidean space to generate small molecule structures with up to 29 atoms, compared to nine heavy atoms that can be achieved with equivariant normalizing flows[186]. An EDM represents each small molecule as a point cloud that can be described by a graph with nodes $v_i \in V$ representing atoms in the molecule based on an equivalent transformation, thereby combining the equivariant GNN and the diffusion process. The former contains L layers of equivariant graph convolutional layers that take each atom's 3D coordinates and features as input to model molecule structures with geometric symmetries, whereas the latter gradually adds Gaussian noise to both the coordinates and features of the atom, thereby improving training, performance and scalability, compared to other E(3)-equivariant models, such as G-Schnet[187] and equivariant normalizing flows[186] as well as graph-based molecule-generative models, such as GraphVAE[188], GraphTransformer[189] and Set2GraphVAE[190].

Based on the equivariant GNN architecture and inspired by the physics governing the formation of small molecules, the Lyapunov function applies physical and statistics prior information (diffusion informative prior bridge)[109] to guide the diffusion process in model training and generate high-quality and realistic molecules. In this approach, problem-dependent prior information, in particular, physical and statistics information, is injected into the diffusion process instead of imposing or improving deep learning architectures. Several energy functions, integrated with the physical and statistical prior information, are then used as a prior bridge to guide the model training without any extra modification of the equivariant GNN architecture. Thereby, the Lyapunov function shows better molecule-generation performance in terms of physical energy and molecule stability[109] and better uniformity-promoted 3D point cloud generation compared

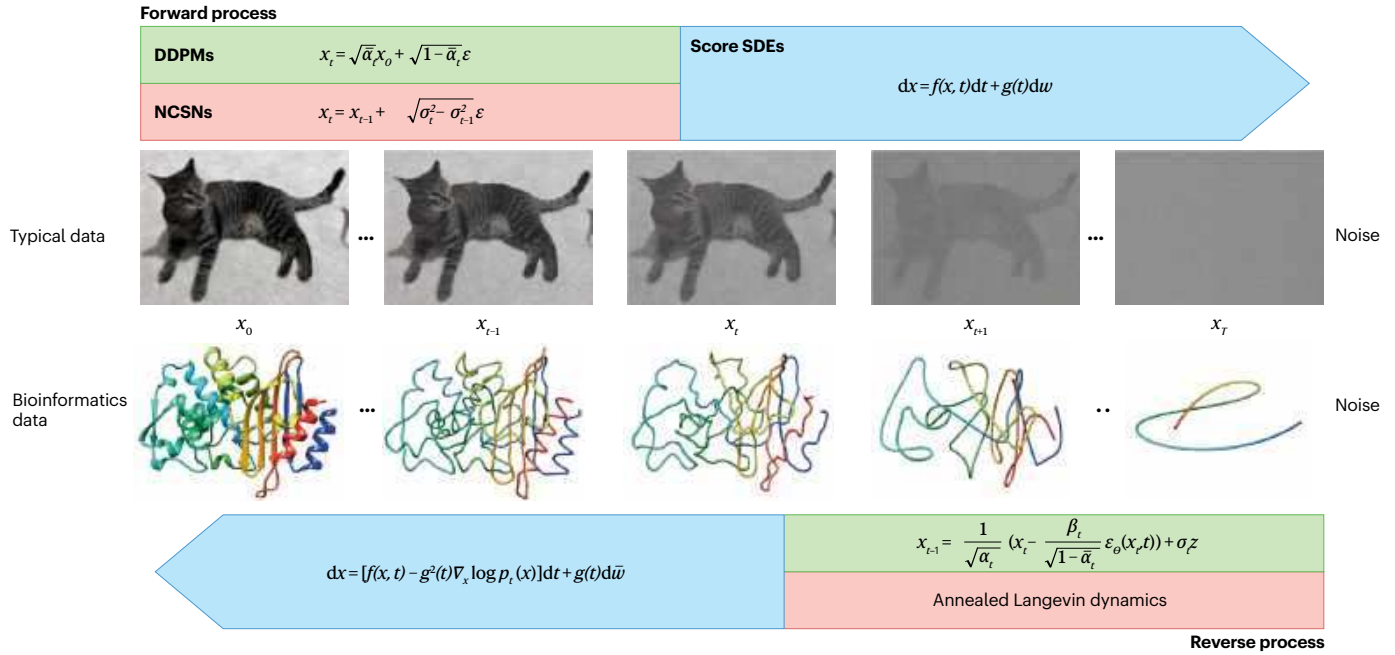

Fig. 2 | Forward and reverse processes of diffusion models. Forward and reverse processes are shown for denoising diffusion probabilistic models (DDPMs), noise-conditioned score networks (NCSNs) and score stochastic differential equations (score SDEs). The forward process progressively (from time 0 to T) adds noise to data (for example, to an image of a cat or a three-dimensional protein structure). The reverse process generates cleaner data from noisier data (that is, it denoises data) from time T to 0. x_0 denotes uncorrupted data, $t \in \{1, \ldots, T\}$ where T represents the number of diffusion steps; $\beta_t \in [0,1)$ is the hyperparameter denoting the variance schedule across the diffusion steps; $\mathbf{I}$ is the identity matrix; $\mathcal{N}(x;\mu,\sigma)$ is the normal distribution of x with mean μ and standard deviation σ; $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t}\alpha_s$; w represents the standard Wiener process known as Brownian motion; $f(x,t)$ and $g(t)$ are the drift and diffusion coefficients of SDE, respectively; and $p(x)$ is the probability density function.

to EDM[54] and point cloud diffusion[143], which apply the traditional Gaussian noise in model training, as well as equivariant normalizing flows[186].

Dynamic graph score matching (DGSM)[110] is a deep learning model developed for predicting stable 3D conformations from 2D molecular graphs, primarily used in computational chemistry. The model can also be extended to protein sidechain conformation prediction and complex multi-molecular prediction (for example, predicting the interaction of more than three small molecules without explicit bonds)[110]. Deep learning methods often consider only the local interactions between bonded atoms, while neglecting the long-range interactions among unbound atoms, which are crucial for constructing accurate 3D molecular structures. To overcome this limitation, DGSM treats each molecule as a graph $\boldsymbol{g} = <\boldsymbol{v},\boldsymbol{e}>$, where a node in v represents an atom and its features (for example, coordinates), and an edge in e represents a bond between two atoms. The distance $\boldsymbol{D_{ij}}$ between each pair of atoms, that is, the edge length in the graph, can then be computed from their coordinates. For each pair of unbound atoms, the distance $\boldsymbol{D_{ij}}$ can be perturbed by a Gaussian noise level at each training step. A message passing neural network[191] is then applied, using edge length and edge type in the graph as inputs to dynamically embed the molecular 2D graph by adding Gaussian noise to the distance between pairs of unbound atoms. Using the score-matching method, the model can then directly estimate the gradient fields of the logarithm density of atomic coordinates. Importantly, the model can be trained in an end-to-end fashion, thereby addressing the limitation of physics-based simulation methods that do not account for long-range interactions between non-bounded atoms. Thus, DGSM outperforms other methods, including RDKit[192], CGCF[193] and ConfGF[144] in terms of matching score and coverage score, confirming the benefit of modelling long-range interactions.

SDEGen[111] is a multi-stage diffusion model that can generate molecules by adopting multiple architectures in different stages with different purposes; here, molecular conformations, including distances between two atoms within three-hop edges, edge type and atom type, and their corresponding graphs, are used as inputs for three different multilayer perceptrons to generate their embeddings. The distance embeddings are corrupted by Gaussian noise and the atom-type embeddings are then updated by a GNN (Box 2). The noisy distance embeddings, edge-type embeddings and the updated atom-type embeddings are then combined into final bond embeddings. Finally, the SDE network is parameterized. This multi-stage model is not as streamlined as end-to-end models, but it outperforms several other models, including DGSM[194], CGCF[193], ConfGF[144], CVGAE[195] and DMCG[196], by multiple metrics, such as coverage score and matching score, in particular, when considering long-range interactions in molecules.

DiffMD[112] is a score-based denoise diffusion model that can be applied to improve molecular dynamics simulations. Deep-learning-based molecular dynamics models typically depend on intermediate force fields and can thus only be applied to static molecules, not considering thermodynamics. DiffMD addresses this problem by applying score-based conditional diffusion models, employing the equivariant geometric transformer to take atomic coordinates, velocity

and features embedded in molecular dynamics trajectories directly as input. In each layer, the model introduces velocities, directions and other geometric information using the spherical Fourier–Bessel transformation to update the input information. During the diffusion process, the conditional noise, based on the accelerations of atoms in previous frames, is added to the inputs for the equivariant geometric

Table 1 | Open-source tools for implementing and improving diffusion models

Tool Name	Year	Model type or improvement	Diffusion framework[a]	Network architecture	GitHub
NCSN[34]	2019	Foundational model	NCSN	Variant of U-Net	CODE
DDPM[33]	2020	Foundational model	DDPM	U-Net[237]	CODE
Score SDE[35]	2020	Foundational model	Score SDE	Variant of U-Net	CODE
EDP-GNN[151]	2020	Diffusion on graph data	NCSN	Variant of GNN	CODE
Improved diffusion[130]	2021	Speed up training	DDPM	U-Net	CODE
Cold diffusion[131]	2021	Speed up training	Score SDE	U-Net	CODE
DDIM[97]	2021	Speed up sampling	DDPM	U-Net	CODE
Diffusion-point-cloud[143]	2021	Point cloud	DDPM	Autoencoder	CODE
LSGM[140]	2021	Mixed modelling	Score SDE	Autoencoder	CODE
ConfGF[144]	2021	Diffusion on graph data	Score SDE	GNN	CODE
PVD[145]	2021	Point cloud	DDPM	Autoencoder	CODE
sdeflow-light[252]	2021	SDE unification	Score SDE	VAE	CODE
Gotta Go Fast[134]	2021	Speed up sampling	Score SDE	Variant of U-Net	CODE
Score-flow[120]	2021	Mixed modelling	Score SDE	U-Net	CODE
DiffFlow[141]	2021	Mixed modelling	Score SDE	U-Net	CODE
FastDPM[132]	2021	Speed up training	Score SDE	U-Net	CODE
argmax_flows[73]	2021	Categorical data	DDPM	VAE, DenseNet	CODE
Soft Truncation[253]	2021	Likelihood optimization	Score SDE	Variant of U-Net	CODE
ARDM[146]	2022	Categorical data	DDPM	Variant autoregressive models	CODE
k-Diffusion[135]	2022	Speed up sampling	Score SDE	Variant of U-Net	CODE
DPM-Solver[136]	2022	Speed up sampling	Score SDE	Plugin method	CODE
VQ-diffusion[254]	2022	Vector quantized	DDPM	Vector quantized VAE	CODE
Improved VQ-Diff[255]	2022	Vector quantized	DDPM	Vector quantized VAE	CODE
Diffusion GAN[38]	2022	Mixed modelling	DDPM	GANs	CODE
DiffuseVAE[142]	2022	Mixed modelling	DDPM	VAE	CODE
PNDM[137]	2022	Speed up sampling	Score SDE	Variant of U-Net	CODE
GeoDiff[147]	2022	Diffusion on graph data	DDPM	Variant of VAE	CODE
VDM[126]	2022	Speed up training	DDPM	Variant of VAE	CODE
Analytic-DPM[138]	2022	Speed up sampling	NCSN	U-Net	CODE
Point Diffusion-Refinement[53]	2022	Point cloud	DDPM	Variant of U-Net	CODE
GDSS[148]	2022	Diffusion on graph data	Score SDE	Multiple stage architecture	CODE
Riemannian-score-sde[149]	2022	Diffusion on manifold data	Score SDE	Multilayer perceptron	CODE
Diffusion Distillation[133]	2022	Speed up training	DDPM	Variant of U-Net	CODE
bit-diffusion[150]	2022	Diffusion on discrete data	DDPM	Variant of U-Net	CODE
DreamFusion[256]	2022	Generate 3D data	DDPM	Multi-stage architecture	CODE
diffusers[257]	2022	Improve sample quality	DDPM	Multiple network architectures	CODE
DPM-Solver++[139]	2022	Speed up sampling	DDPM	Multiple network architectures	CODE

[a]The diffusion frameworks are general models that are typically combined with specific deep learning architectures (network architectures) to generate or denoise a specific type of data. Tools intended to increase sampling speed and the quality of diffusion are tested with various network architectures, and their associated network architectures are categorized as 'multiple network architectures'. NCSN, noise-conditioned score network; U-Net, U-shaped neural network; DDPM, denoising diffusion probabilistic model; score SDE, score stochastic differential equation; GNN, graph neural network; VAE, variational autoencoder; GAN, generative adversarial network.

Box 3

A practical guide for applying diffusion models in bioinformatics

Diffusion models are particularly useful in the generation, design or analysis of small molecules, proteins and biological images. To decide which diffusion model to apply to a specific problem, the representation of the specific data type (for example, small molecules) needs to be considered to be suitable for processing by a deep learning model in the diffusion process. The conformations of small molecules and drugs can be represented in several ways to facilitate the diffusion process; for example, they can be treated as a string, such as the SELF-referencing embedded string, which can be converted into a two-dimensional (2D) matrix. This matrix can be used as input for graph neural networks (GNNs) under a diffusion model framework to generate three-dimensional (3D) molecular graphs, as exemplified by dynamic graph score matching [110]. Alternatively, they can be presented as 3D graphs that contain spatial direction and torsion angles between atoms, which can be used by a combination of SE(3)-equivariant GNNs[162] and diffusion models, such as the E(3)-equivariant diffusion model[54] to capture their essential properties. In addition, small molecules can be represented as 3D atomic point clouds to be processed by equivariant GNNs, as in DiffLinker[113]. Proteins can be represented as either one-dimensional (1D) sequential features suitable for a 1D transformer or 2D contact and distance maps suitable for processing by convolutional neural networks. The 3D structure of proteins is usually represented as graphs that consist of nodes denoting residues and edges that represent residue pairs in contact, which can be handled by both standard GNNs and SE(3)-equivariant GNNs in combination with diffusion models. For imaging data, such as cryo-electron microscopy images, various diffusion models initially developed for image generation, such as CascadedDiff[60], can be applied. Biomolecules or cell shapes may also be represented by 3D images, which can be reconstructed from 2D images by a combination of autoencoder or U-Net[237] with a diffusion model, as in CryoDRGN[98] and DISPR[99]. These can model the distribution of ground-truth data to generate higher-quality 3D images than other generative artificial intelligence methods. For example, DISPR outperforms a VAE-based deep generative model SHAPR[238] in the context of 3D cell shape reconstruction.

transformer to estimate the score function, that is, the gradient of the log density of the biomolecule conformations. DiffMD outperforms several deep-learning-based molecular dynamics methods, including tensor field networks[162], radial fields[197], SE(3)-transformers[198], graph mechanics networks[199] and SCFNN[200] in terms of average root-mean-squared error.

Fragment-based drug design can also be used for the discovery of new small molecules in a 3D space. Here, the aim is to design linkers consisting of atoms that can connect molecular fragments into a complete molecule. DiffLinker[113] uses an E(3)-equivariant 3D conditional diffusion model to generate these molecular linkers and to connect multiple molecular fragments to form a single connected molecule. The prediction is made by applying a GNN to predict the linker size (the atom number of the linker) and atom types. The coordinates of the atoms are sampled from the normal distribution, followed by a reverse diffusion process of the atom features conditioned on the input fragments. Compared to DeLinker[201] and 3DLinker[202], DiffLinker can perform better in terms of average quantitative estimation of drug-likeness, synthetic accessibility, the average number of rings in the linker, and the validity, uniqueness and novelty of the samples, thereby generating more realistic molecules.

Protein–ligand interaction modelling

Predicting the conformation of a ligand bound to a protein is important in the investigation of protein–ligand interactions and protein function as well as for the discovery of new drugs. Various protein–ligand docking, machine learning and auto-regressive models have been developed to address this problem[10,203–206]; however, these approaches are limited by their low geometrical accuracy. Alternatively, DiffBP[114] can generate ligands that bind to a specific protein pocket without requiring the ligand structure as input; here, a pre-generation network is used to generate the centre of mass and atom number of the ligand, followed by diffusion models in conjunction with equivariant GNNs[161,207] to generate high-quality ligand candidates[33,35]. Compared to auto-regressive methods, such as 3DSBDD[205], Pocket2Mol[206] and GraphBP[203], which generate one atom at a time without considering interactions among all atoms, DiffBP can generate all atoms of a ligand that bind to a target protein, exhibiting high binding affinities (for example, 41.07%[114] for DiffBP, compared to 12.22%[114] for 3DSBDD, 23.98%[114] for Pocket2Mol and 29.54%[114] for GraphBP) on the CrossDocked[208] dataset curated from protein–ligand complex structures in the Protein Data Bank (PDB).

DiffSBDD[115] adopts a DDPM equipped with an E(3)-equivariant neural network to generate new ligands, including atomic features binding to specific protein pockets; here, ligand generation can either be protein-conditioned, based on the binding site to the protein, or the ligand can be impainted after learning the joint distribution of the protein–ligand complexes. Compared to 3DSBDD and Pocket2Mol, DiffSBDD can generate more diverse ligands with higher affinity on the CrossDocked dataset[115].

Unlike diffusion models applied for protein pocket docking, DiffDock[116] uses the structure of the protein and ligand as input and does not require knowledge of the location of the binding site (that is, blind docking); here, the diffusion process is applied to ligand positions, represented by ligand translation and rotation, sampling multiple positions, which are then ranked based on a confidence score using a trained scoring model and a trained confidence model, which are built on top of SE(3)-equivariant GNNs (Box 2). The scoring model samples different positions of the ligand, and the confidence model selects the ligand positions with the highest confidence score, similar to the structural and scoring modules of AlphaFold2[163] for protein structure prediction. DiffDock has been tested on the PDBBind dataset, outperforming search-based methods, such as SMINA[209], QuickVina-W[210], GLIDE[211] and GNINA[212], and the deep learning methods EquiBind[213] and TANKBind[214]. Specifically, DiffDock achieved a top-1 success rate of 38.2% (the percentage of top-1 predictions with root-mean-square deviation <2 Å),

which is significantly better than the energetics-based method GLIDE (21.8%; $P = 2.7 \times 10^{-7}$) and the geometric deep-learning-based method TANKBind (20.4%; $P = 1.0 \times 10^{-12}$)[116].

Similar to DiffDock, NeuralPLexer[117] is a deep generative network that leverages SDEs to predict complex protein–ligand structures based on the protein structure and molecular graphs of the ligand as input in blind docking. The key component in the model is an equivariant structure diffusion module, which predicts the atomic coordinates on a heterogeneous graph formed by protein atoms, ligand atoms, protein backbone frames and ligand local frames. Using SDEs, the model can handle unbound or predicted protein structure inputs and can automatically accommodate changes in the protein structure in response to ligand binding. Compared with the deep learning method EquiBind[213] and the physics-based method CB-Dock[215] on the PDBBind[216] dataset, NeuralPLexer can generate a more accurate ligand structure with higher geometrical accuracy, with an approximately 70% success rate for a ligand with root-mean-square deviation <2 Å, which is higher than that of EquiBind (about 40%) and CB-Dock (about 38%) and has a lower steric clash rate of 0.105.

Finally, a deep generative energy-based diffusion model can predict the binding affinity for a protein–ligand pair, if trained with a set of protein–ligand complexes, without requiring labels for binding affinities[118]. During training, the network first predicts the rotation score for the perturbed ligand with respect to the protein pocket using an equivariant rotation prediction network, called Neural Euler's Rotation Equation (NERE). By training the model with the SE(3) denoising score matching, the log-likelihood is considered to be the binding affinity between the protein and ligand in a pair. Tested on the protein–ligand dataset PDBbind[216] and the structural antibody database SAbDab[217], the model achieves an accuracy of 0.656 in predicting protein–ligand binding affinity, which is better than that of other unsupervised methods: 0.647 for Molecular Mechanics Generalized Born Surface Area[218] (MM/GBSA), 0.617 for Astex Statistical Potential[219] (ASP) and 0.602 for DrugScore2018 (ref. 220). This model further performs comparably to other supervised methods in predicting antibody–antigen binding[118]: Zlab RerANK[221], ZRANK2[222], RosettaDock[223], PyDock[224], Scoring by Intermolecular Pairwise Propensities of Exposed Residues (SIPPER)[225], Atomic Potential Protein Interactions Scored Atomically (AP_PISA)[226], Coarse Grained Protein Interaction Energy (CP_PIE)[227] and FIREDOCK[228].

Cryo-electron microscopy data analysis

Single-particle cryo-electron microscopy (cryo-EM)[229–235] is a key imaging technique for determining and visualizing the 3D conformation (structure) of large biomolecular complexes (for example, protein complexes) at atomic resolution; here, the images of protein complexes obtained by cryo-EM are used to reconstruct their 3D conformation represented by 3D density maps.

The protein structure reconstruction method CryoDRGN[236] introduces a latent variable Z to define a conformational space V for a protein complex on cryo-EM density maps. CryoDRGN is based on a VAE framework that learns a continuous distribution in the latent space for protein structures from cryo-EM data. However, although CryoDRGN can simulate complicated structural dynamics, the Gaussian prior distribution of VAE does not match the posterior aggregate approximation, which limits the generative capability of the model[236]. Alternatively, a continuous-time diffusion model (that is, score SDEs) can be implemented in CryoDRGN to learn a high-quality generative model for capturing protein conformations directly from cryo-EM imaging data. This CryoDRGN[98] model is first trained with the standard VAE model using cryo-EM images in Fourier space. The latent space Z, which is predicted by the encoder of the trained VAE, is then fed into the denoise diffusion model based on a ResNet architecture[28] to approximate the distribution of the latent variable Z. Finally, the synthesized latent variable Z, which is sampled from the diffusion model and is similar to the target protein's distribution, is used as input for the decoder of the VAE to generate protein structures with better quality (higher similarity with the target proteins' distribution) than a VAE, which directly reconstructs protein structures by learning continuous distribution in latent space.

Single-cell image and gene-expression analysis

Reconstructing the 3D shape of a cell from a single-cell 2D microscopy image using computational methods is useful for studying the morphological features of cells. However, each 2D image may permit multiple 3D reconstructions, and therefore, different 2D slices may lead to different predictions of the 3D shape. To tackle this issue, DISPR[99] employs the U-net architecture[237] and a diffusion process to generate a single-cell 3D shape from 2D images. During training and evaluation, this approach uses a 2D image of an individual cell as an inductive bias. The 2D image is then concatenated with its 3D Gaussian noisy segmentation mask as input for the diffusion-based model to predict realistic 3D cell shapes. DISPR benefits from this training approach and its stochastic property. Unlike VAE-based architectures used in SHAPR[238] and its variants[239], which produce a single, deterministic reconstruction, DISPR employs a stochastic model trained on Gaussian noise and is thus capable of predicting an infinite number of cell shapes, providing a more comprehensive representation of dynamic cell structures. DISPR represents the first use of a diffusion model in the context of 3D cell shape reconstruction, outperforming VAE-based deep generative models, such as SHAPR[238], in terms of volume, surface area and roughness reconstruction[99].

Single-cell RNA sequencing can assess the expression of genes in individual cells. However, cells typically contain low quantities of RNA, which may cause noisy measurements (for example, varied measurements and experimental bias) of gene expression; moreover, values may be missed (dropouts). Therefore, it is important to denoise single-cell RNA-sequencing data and impute missing values. DEWAKSS[100] applies a diffusion model with a K-nearest-neighbour (KNN) graph to select denoising hyperparameters using the noise2self self-supervision method, thereby not depending on an explicit noise model but on an invariant function of data features. Unlike heuristic-based methods, such as MAGIC[240] and KNN-smoothing[241], which also use KNN graph architecture but may lead to over-smoothing of data variance, DEWAKSS can preserve variances across multiple gene-expression dimensions.

Open-source diffusion model tools

Some diffusion models that can be applied to bioinformatics have been implemented as open-source tools (Table 2). However, these tools do not use NCSNs[34] as the diffusion framework, mainly because NSCNs face problems in terms of sampling and training and can thus not achieve high accuracy in image generation. Therefore, NCSNs are less adopted in bioinformatics and computational biology than DDPMs[33] and score SDEs[35], which are equipped with efficient sampling and training methods[32] for high-definition image generation. Nevertheless, as the first diffusion model, NSCN has made substantial contributions to the development of the field. Furthermore, many

Table 2 | Open-source diffusion model tools for bioinformatics

Applications	Tool name	Denoising condition	Diffusion framework	Network architecture	Github
Protein design and generation	ProteinSGM[91]	Conditioned and unconditioned	Score SDE	Convolutional neural network	CODE
	FoldingDiff[101]	Unconditioned	DDPM	Transformer	CODE
	Genie[103]	Unconditioned	DDPM	Variant SE(3)-transformer	CODE
	SMCDiff[104]	Conditioned	Score SDE	EGNN	CODE
	FrameDiff[106]	Unconditioned	Score SDE	SE(3)-transformer	CODE
	RFdiffusion[105]	Conditioned and unconditioned	DDPM	SE(3)-transformer	CODE
	Chroma[107]	Conditioned	Score SDE	GNN	CODE
Small-molecule generation and drug design	EDM[54]	Conditioned	DDPM	EGNN	CODE
	SDEGen[111]	Conditioned	Score SDE	GNN	CODE
	DiffLinker[113]	Conditioned	DDPM	EGNN	CODE
Protein–ligand interaction modelling	DiffBP[114]	Conditioned	DDPM	EGNN	CODE
	DiffSBDD[115]	Conditioned	DDPM	EGNN	CODE
	DiffDock[116]	Conditioned	Score SDE	Variant EGNN	CODE
Cryo-EM data analysis	CryoDRGN[98]	Conditioned	Score SDE	VAE	CODE
Single-cell image and gene-expression data analysis	DISPR[99]	Conditioned	DDPM	U-Net	CODE

Score SDE, score stochastic differential equation; DDPM, denoising diffusion probabilistic model; EGNN, equivariant GNN; GNN, graph neural network; cryo-EM, cryogenic electron microscopy; VAE, variational autoencoder; U-Net, U-shaped neural network.

bioinformatics applications also include deep learning components to deal with data generation and denoising challenges specific to their application.

Outlook

Diffusion models can be applied in several bioinformatics applications and may be further extended to other computational biology areas owing to their ability to denoise data and generate realistic new data (Table 3).

3D genomics data analysis

High-throughput chromosome conformation capture (Hi-C) is a key technology for studying 3D conformations of chromosomes and genomes, applying next-generation sequencing techniques to sequence chromosomal regions that are spatially close to each other (that is, in contact)[242]. Thus, Hi-C data captures the interactions between chromosomal regions of a genome to build 3D conformations of the genome[243,244] and study long-range gene-enhancer interactions. This approach typically requires the data to be converted into 2D chromosomal contact matrices (maps), which store the frequency at which chromosome region *i* interacts with chromosome region *j*, where *i* and *j* are the indices of chromosome regions. Therefore, a Hi-C contact matrix can be considered an image. However, Hi-C data, in particular, single-cell Hi-C data, are usually noisy and incomplete, so that chromosomal interactions in chromosomal contact matrices may be false positives or interactions may be missing in the matrices. Deep learning methods (for example, GANs) can be applied to denoise Hi-C data[245]; in addition, diffusion models (for example, DDPM) may enable denoising of Hi-C chromosomal contact matrices to improve 3D genome conformation modelling and to study spatial interactions between genes and regulatory elements (for example, enhancers). However, the deep learning architecture of DDPM is typically the U-Net, which may not be as powerful as the deep residual network used in the Hi-C data denoising method ScHiCEDRN[246]. Thus, if applied to Hi-C data denoising, the architecture of DDPM would have to be updated to deep residual networks to improve its denoising ability.

Single-cell reconstruction and inference

The activity of a single cell can be captured by various 'omics data, such as transcriptomics (RNA-seq), proteomics, chromosome accessibility (ATAC-Seq) and epigenetics (bisulfite sequencing), which may benefit from diffusion models; for example, data could be inferred to one modality (for example, RNA-Seq) from another modality, such as ATAC-Seq data and genome methylation data; missing spots in single-cell spatial transcriptomic data could be calculated; spots (each consisting of multiple cells) in 10× spatial transcriptomic data could be decomposed into single cell data (super-resolution); and single-cell data could be used to build 3D models of the spatial arrangement of cells. Moreover, diffusion models designed to denoise images could also be applied to denoise single-cell 'omics data, such as transcriptomics, proteomics, metabolomics and epigenetics data.

DNA regulatory element design

The expression of genes is modulated by short DNA sequences on genomes, called regulatory elements, such as enhancers and promoters. Designing regulatory elements is an important approach to designing synthetic cells using synthetic biology. Generative models, such as GANs[247], can be applied to design enhancers that regulate the expression of genes and the development of cell types. However, diffusion models have shown better performance in image synthesis than GANs[50] and may thus be more suitable for the design of enhancers and other gene regulatory elements.

Table 3 | Applications of diffusion models in bioinformatics

Applications	Tool name	Key function and strength	Input	Output	Diffusion target[a]
Protein design and generation	ProteinSGM[91]	Inpaints plausible backbones/domains; generates native-like structures; allows precise and modular design	Inter-residue 6D feature maps	Full-atomistic structure	Inter-residue 6D feature maps
	FoldingDiff[101]	Mirrors native folding process; alleviates the need for equivariant networks; unconditionally generates realistic protein structures	6 consecutive backbone angles	Protein backbone structure	6 consecutive backbone angles
	DiffSDS[102]	Reduces computational complexity and cost; efficiently imposes geometric constraints; outperforms previous strong baselines	6 consecutive backbone angles	Masked protein backbone structure	6 consecutive backbone angles
	ProSSDG[84]	Operates at large scales; generates realistic proteins structures with sequences; allows interactive structure generation	Secondary structure; coarse constraints	All-atomistic protein structure	Coarse constraints
	Genie[103]	Dual representation for protein residues; designability and diversity	Oriented reference frames	Protein backbone structure	Oriented reference frames
	SMCDiff[104]	Efficiently samples scaffolds; samples conditioned on given motif; theoretically guarantees conditional samples	Molecular graph structure	Scaffold structure given input motif	Molecular graph structure
	RFdiffusion[105]	Generates diverse outputs; can be guided toward specific design objectives; explicitly models 3D structure	Sequence, predicted structure	Diverse, complex, functional protein	RF frames from a predicted structure
	FrameDiff[106]	Generates designable monomers and diverse protein backbones; does not require pretrained structure predictor	Molecular graph structure	Designable monomer backbone structure	Molecular graph structure
	Chroma[107]	Jointly models structures and sequences; sub-quadratic computational scaling; arbitrary conditional sampling	Protein graph structure	Proteins with desired functions	Protein graph structure
Small-molecule generation and drug design	CDGS[108]	Incorporates discrete graph structures; specialized graph noise prediction model; similarity-constrained molecule optimization pipeline	Graph structures and inherent features	Molecular graphs	Discrete graph structure
	EDM[54]	Equivariant to Euclidean transformations; operates on continuous and categorical features; admits likelihood computation	Atom coordinates, atom types	3D molecular graphs	Coordinates and categorical features
	DiffBridge[109]	Incorporates prior information; generates realistic molecules; uses physically informed diffusion bridges	Masked points in 3D Euclidean space	Realistic molecules or point cloud	Molecular graph structure
	DGSM[110]	Models local and long-range interactions; dynamically constructs graph structures; estimates gradient fields of logarithm density	2D molecular graphs	Stable 3D conformations	Molecular graph structure
	SDEGen[111]	Captures multimodal conformation distribution; quickly searches low-energy conformations and higher efficiency	Small molecules	Representative conformations	Encoding of graph structure
	DiffMD[112]	No intermediate variables; directly estimates gradient of log density; incorporates directions and velocities of atomic motions	3D coordinates, velocities, and invariant features	Molecule simulation trajectories	Atomic position coordinates
	DiffLinker[113]	Links arbitrary number of fragments; generates diverse and synthetically accessible molecules; conditions on protein pockets	Molecule structure	A molecule incorporating all the input fragments	Molecular graph structure
Protein–ligand interaction modelling	DiffBP[114]	Non-autoregressive generation; generates molecules with high affinity to target proteins and desirable drug properties	Protein–ligand structure	Protein–ligand structure given input protein	Protein graph structure
	DiffSBDD[115]	Generates diverse drug-like ligands; efficient in silico experiments; uses experimentally determined binding data	Protein–ligand structure	High-affinity ligands given protein pockets	Protein graph structure
	DiffDock[116]	Maps manifold to product space; provides confidence estimates; maintains precision on computationally folded structures	Protein–ligand structure	Ranked ligand poses, confidence scores	Ligand poses
	NeuralPLexer[117]	Repacks failed AlphaFold2 sites; enables end-to-end design; generalizes to ligand-unbound or predicted protein structure inputs	Protein backbone template and ligand molecular graphs	Full-atom protein–ligand structure	Contact maps, geometry prior
	NERE[118]	Benchmarks on protein–ligand and antibody–antigen dataset; outperforms other unsupervised methods	Protein–ligand complex	Binding affinity of the protein–ligand structure	Ligand atom coordinates

Table 3 (continued) | Applications of diffusion models in bioinformatics

Applications	Tool name	Key function and strength	Input	Output	Diffusion target[a]
Cryo-EM data analysis	CryoDRGN[98]	Accurate data distribution sampling ; fast latent space traversal; unlocks generative modelling tools	Single-particle cryo-EM imaging	High-quality 3D structure	Latent space of image embeds
Single-cell image and gene-expression analysis	DISPR[99]	Realistic 3D reconstructions; data augmentation tool in single-cell classification task; inverse biomedical problems	2D microscopy images as a prior	3D cell shape predictions	3D microscopy point cloud
	DEWAKSS[100]	Maintains cellular identity; preserves data variance; maintains cluster homogeneity	Gene-expression data	Single-cell genomics data	Gene-expression matrix

[a]The term 'diffusion' refers to the application of the forward and reverse diffusion processes on a specific data representation, that is, the target. Cryo-EM, cryogenic electron microscopy. RF, RosettaFold.

Cryo-electron microscopy image denoising

Diffusion models can reconstruct complex protein structures from 3D cryo-EM density maps, which are typically made of noisy and low-contrast 2D cryo-EM protein particle images, isolated from large 2D cryo-EM protein images (also called cryo-EM micrographs). However, denoising original cryo-EM images to build better 3D cryo-EM density maps remains challenging. Although image preprocessing techniques, such as EMAN2 (ref. 248), can denoise cryo-EM images[249], diffusion models trained on many noisy images at various noise levels and their clean counterparts may allow the recovery of clean cryo-EM images more effectively than conventional image processing techniques that have not been trained to learn the noise distribution of cryo-EM images[250].

Peptide design

Peptides, which are short, typically unfolded amino acid sequences, can bind to proteins to modulate their function, which has been explored for drug design. Diffusion models can not only be designed to generate new proteins but could also be adapted to create peptides that can modulate protein function. For example, diffusion models pretrained for protein design may be retrained on a peptide dataset to design peptides through transfer learning.

Protein structure refinement and mutation prediction

The tertiary and quaternary structures of many proteins and protein complexes can be fairly accurately predicted by AlphaFold2 (ref. 163) and AlphaFold-multimer[251], respectively. However, such predicted structures may contain structural errors and may thus need to be refined. Conditioned on a predicted structure input, diffusion models may be able to remove noise from the predicted structure to bring it closer to the native structure. Similarly, it remains challenging to predict how an amino acid mutation alters the structure of a protein, which may affect protein function and phenotype. Diffusion models have generative capability, demonstrated in protein design, and may therefore be able to transform the known structure of a protein without mutation (wild-type protein) to the structure of the same protein with mutations (mutant) to predict structural changes induced by mutations.

Limitations of current diffusion models

Although diffusion models may be applied for various bioinformatics applications, potentially outperforming GANs and VAEs, some limitations remain to be addressed. First, the training process of diffusion models involves the introduction of Gaussian noise to the data, resulting in a long training time. Second, although considerable efforts have been directed towards increasing the sampling speed in diffusion models, the sampling time of most models still exceeds that of other deep generative models (for example, GANs and VAEs). The long sampling time hinders some real-time applications of diffusion models. Developing a streamlined approach of single-step noise addition and removal may reduce the training and sampling time. Third, the computational resource requirements of diffusion models are higher than those of GANs and VAEs. Therefore, the trade-off between performance improvement and computational resource demand needs to be evaluated when deciding which model to use. Furthermore, new applications of diffusion models are often non-trivial and may require validation of suitable data representations (embeddings), types of diffusion model and deep learning architectures.

Citation diversity statement

We acknowledge that papers authored by scholars from historically excluded groups are systematically under-cited. Here, we have made every attempt to reference relevant papers in a manner that is equitable in terms of racial, ethnic, gender and geographical representation.

Published online: 27 October 2023

References

1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015). **This article provides a comprehensive overview of the advances, challenges and potential of deep learning methods.**
2. Eickholt, J. & Cheng, J. Predicting protein residue–residue contacts using deep networks and boosting. *Bioinformatics* **28**, 3066–3072 (2012).
3. Baek, M. & Baker, D. Deep learning and protein structure modeling. *Nat. Methods* **19**, 13–14 (2022).
4. Kulmanov, M. & Hoehndorf, R. DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics* **36**, 422–429 (2020).
5. Aggarwal, D. & Hasija, Y. A review of deep learning techniques for protein function prediction. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2211.09705 (2022).
6. Gligorijević, V. et al. Structure-based protein function prediction using graph convolutional networks. *Nat. Commun.* **12**, 3168 (2021).
7. Bileschi, M. L. et al. Using deep learning to annotate the protein universe. *Nat. Biotechnol.* **40**, 932–937 (2022).
8. Cai, Y., Wang, J. & Deng, L. SDN2GO: an integrated deep learning model for protein function prediction. *Front. Bioeng. Biotechnol.* **8**, 391 (2020).
9. Ko, C. W., Huh, J. & Park, J.-W. Deep learning program to predict protein functions based on sequence information. *MethodsX* **9**, 101622 (2022).
10. Dhakal, A., McKay, C., Tanner, J. J. & Cheng, J. Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions. *Brief. Bioinform.* **23**, bbab476 (2022).
11. Verma, N. et al. Ssnet: a deep learning approach for protein–ligand interaction prediction. *Int. J. Mol. Sci.* **22**, 1392 (2021).
12. Gomes, J., Ramsundar, B., Feinberg, E. N. & Pande, V. S. Atomic convolutional networks for predicting protein–ligand binding affinity. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1703.10603 (2017).
13. Jiménez, J., Skalic, M., Martinez-Rosell, G. & De Fabritiis, G. Kdeep: protein–ligand absolute binding affinity prediction via 3D-convolutional neural networks. *J. Chem. Inf. Model* **58**, 287–296 (2018).
14. Öztürk, H., Özgür, A. & Ozkirimli, E. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics* **34**, i821–i829 (2018).

15. Zhou, J. et al. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nat. Genet.* **50**, 1171–1179 (2018).
16. Avsec, Ž. et al. Effective gene expression prediction from sequence by integrating long-range interactions. *Nat. Methods* **18**, 1196–1203 (2021).
17. Zrimec, J. et al. Deep learning suggests that gene expression is encoded in all parts of a co-evolving interacting gene regulatory structure. *Nat. Commun.* **11**, 6141 (2020).
18. Yuan, Y. & Bar-Joseph, Z. Deep learning for inferring gene relationships from single-cell expression data. *Proc. Natl Acad. Sci.* **116**, 27151–27158 (2019).
19. Khan, A. & Lee, B. Gene transformer: transformers for the gene expression-based classification of lung cancer subtypes. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2108.11833 (2021).
20. Singh, R., Lanchantin, J., Robins, G. & Qi, Y. DeepChrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics* **32**, i639–i648 (2016).
21. Shu, H. et al. Modeling gene regulatory networks using neural network architectures. *Nat. Comput. Sci.* **1**, 491–501 (2021).
22. Razaghi-Moghadam, Z. & Nikoloski, Z. Supervised learning of gene-regulatory networks based on graph distance profiles of transcriptomics data. *npj Syst. Biol. Appl.* **6**, 21 (2020).
23. Chen, C. et al. DeepGRN: prediction of transcription factor binding site across cell-types using attention-based deep neural networks. *BMC Bioinform.* **22**, 38 (2021).
24. Xu, R., Zhang, L. & Chen, Y. CdtGRN: Construction of qualitative time-delayed gene regulatory networks with a deep learning method. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2111.00287 (2021).
25. Kwon, M. S., Lee, B. T., Lee, S. Y. & Kim, H. U. Modeling regulatory networks using machine learning for systems metabolic engineering. *Curr. Opin. Biotechnol.* **65**, 163–170 (2020).
26. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324 (1998).
27. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
28. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* 770–778 (IEEE, 2016).
29. Goodfellow, I. et al. Generative adversarial networks. *Commun. ACM* **63**, 139–144 (2020).
30. Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. & Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **20**, 61–80 (2008).
31. Vaswani, A. et al. Attention is All you Need. In *Advances in Neural Information Processing Systems* Vol. 30 (eds Guyon, I. et al.) (Curran Associates, 2017).
32. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In *Proc. 32nd Int. Conference on Machine Learning* Vol. 37 (eds Bach, F. & Blei, D.) 2256–2265 (PMLR, 2015).
33. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **33**, 6840–6851 (2020).
This article introduces the denoising diffusion probabilistic model, which was the first diffusion model capable of generating high-resolution data.
34. Song, Y. & Ermon, S. Generative Modeling by Estimating Gradients of the Data Distribution. In *Advances in Neural Information Processing Systems* Vol. 32 (eds Wallach, H. et al.) (Curran Associates, 2019).
This article introduces the noise-conditioned score network, which is one of the three main diffusion model frameworks.
35. Song, Y. et al. Score-based generative modeling through stochastic differential equations. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2011.13456 (2020).
This article introduces score stochastic differential equations for unconditional image generation.
36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* 10684–10695 (2022).
This article reports stable diffusion for image inpainting, class-conditional image synthesis and other tasks, including text-to-image synthesis and unconditional image generation.
37. Saharia, C. et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 36479–36494 (Curran Associates, 2022).
38. Wang, Z., Zheng, H., He, P., Chen, W. & Zhou, M. Diffusion-GAN: Training GANs with diffusion. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2206.02262 (2022).
39. Zheng, H., He, P., Chen, W. & Zhou, M. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2202.09671 (2022).
40. Xie, P. et al. Vector quantized diffusion model with CodeUnet for text-to-sign pose sequences generation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2208.09141 (2022).
41. Kim, D., Kim, Y., Kang, W. & Moon, I.-C. Refining generative process with discriminator guidance in score-based diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2211.17091 (2022).
42. Zheng, G. et al. Entropy-driven sampling and training scheme for conditional diffusion generation. In *Eur. Conf. on Computer Vision* 754–769 (Springer, 2022).
43. Saharia, C. et al. Palette: image-to-image diffusion models. In *ACM SIGGRAPH '22 Conf. Proc.* https://doi.org/10.1145/3528233.3530757 (ACM, 2022).
44. Wang, Y., Yu, J. & Zhang, J. Zero-shot image restoration using denoising diffusion null-space model. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2212.00490 (2022).
45. Lam, M. W., Wang, J., Su, D. & Yu, D. BDDM: bilateral denoising diffusion models for fast and high-quality speech synthesis. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2203.13508 (2022).
46. van den Oord, A. et al. Conditional Image Generation with PixelCNN Decoders. In *Advances in Neural Information Processing Systems* Vol. 29 (eds Lee, D. et al.) (Curran Associates, 2016).
47. Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S. & Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.* **22**, 1–64 (2021).
48. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M. & Huang, F. A tutorial on energy-based learning. In *Predicting Structured Data* (eds Bakir, G., Hofman, T., Schölkopf, B., Smola, A. & Taskar, B.) Vol. 1 (MIT Press, 2006).
49. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1312.6114 (2013).
50. Dhariwal, P. & Nichol, A. Diffusion models beat GANs on image synthesis. *Adv. Neural Inf. Process. Syst.* **34**, 8780–8794 (2021).
51. Li, H. et al. SRDiff: single image super-resolution with diffusion probabilistic models. *Neurocomputing* **479**, 47–59 (2022).
52. Giannone, G., Nielsen, D. & Winther, O. Few-shot diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2205.15463 (2022).
53. Lyu, Z., Kong, Z., Xu, X., Pan, L. & Lin, D. A conditional point diffusion-refinement paradigm for 3d point cloud completion. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2112.03530 (2021).
54. Hoogeboom, E., Satorras, V. c. G., Vignac, C. & Welling, M. Equivariant Diffusion for Molecule Generation in 3D. In *Proc. 39th Int. Conference on Machine Learning* Vol. 162 (eds Chaudhuri, K. et al.) 8867–8887 (PMLR, 2022).
This article reports a foundational diffusion model that directly generates molecules in 3D space based on an equivariant graph neural network architecture.
55. Li, X., Thickstun, J., Gulrajani, I., Liang, P. S. & Hashimoto, T. B. Diffusion-LM Improves Controllable Text Generation. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 4328–4343 (Curran Associates, 2022).
56. Amit, T., Nachmani, E., Shaharbany, T. & Wolf, L. SegDiff: image segmentation with diffusion probabilistic models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2112.00390 (2021).
57. Baranchuk, D., Rubachev, I., Voynov, A., Khrulkov, V. & Babenko, A. Label-efficient semantic segmentation with diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2112.03126 (2021).
58. Brempong, E. A. et al. Denoising pretraining for semantic segmentation. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* 4175–4186 (IEEE, 2022).
59. Cai, R. et al. Learning gradient fields for shape generation. In *Eur. Conf. on Computer Vision* 364–381 (Springer, 2020).
60. Ho, J. et al. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.* **23**, 1–33 (2022).
61. Ho, J. et al. Video diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2204.03458 (2022).
62. Kawar, B., Vaksman, G. & Elad, M. Stochastic image denoising by sampling from the posterior distribution. In *Proc. IEEE/CVF Int. Conf. on Computer Vision* 1866–1875 (2021).
63. Kim, B., Han, I. & Ye, J. C. DiffuseMorph: unsupervised deformable image registration along continuous trajectory using diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2112.05149 (2021).
64. Luo, S. & Hu, W. Score-based point cloud denoising. In *Proc. IEEE/CVF Int. Conf. on Computer Vision* 4583–4592 (IEEE, 2021).
65. Meng, C. et al. Sdedit: Guided image synthesis and editing with stochastic differential equations. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2108.01073 (2021).
66. Özbey, M. et al. Unsupervised medical image translation with adversarial diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2207.08208 (2023).
67. Saharia, C. et al. Image super-resolution via iterative refinement. In *IEEE Trans. on Pattern Analysis and Machine Intelligence* 4713–4726 (IEEE, 2022).
68. Whang, J. et al. Deblurring via stochastic refinement. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* 16293–16303 (IEEE, 2022).
69. Yang, R. & Mandt, S. Lossy image compression with conditional diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2209.06950 (2022).
70. Zhao, M., Bao, F., Chongxuan, L. I. & Zhu, J. EGSDE: Unpaired Image-to-Image Translation via Energy-Guided Stochastic Differential Equations. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 3609–3623 (Curran Associates, 2022).
71. Zimmermann, R. S., Schott, L., Song, Y., Dunn, B. A. & Klindt, D. A. Score-based generative classifiers. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2110.00473 (2021).
72. Austin, J., Johnson, D. D., Ho, J., Tarlow, D. & van den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Adv. Neural Inf. Process. Syst.* **34**, 17981–17993 (2021).
73. Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P. & Welling, M. Argmax flows and multinomial diffusion: learning categorical distributions. *Adv. Neural Inf. Process. Syst.* **34**, 12454–12465 (2021).
74. Savinov, N., Chung, J., Binkowski, M., Elsen, E. & Oord, A. V. D. Step-unrolled denoising autoencoders for text generation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2112.06749 (2021).
75. Yu, P. et al. Latent diffusion energy-based model for interpretable text modeling. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2206.05895 (2022).
76. Alcaraz, J. M. L. & Strodthoff, N. Diffusion-based time series imputation and forecasting with structured state space models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2208.09399 (2022).
77. Chen, N. et al. WaveGrad: estimating gradients for waveform generation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2009.00713 (2020).

78. Kong, Z., Ping, W., Huang, J., Zhao, K. & Catanzaro, B. DiffWave: a versatile diffusion model for audio synthesis. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2009.09761 (2020).
79. Rasul, K., Sheikh, A.-S., Schuster, I., Bergmann, U. & Vollgraf, R. Multivariate probabilistic time series forecasting via conditioned normalizing flows. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2002.06103 (2020).
80. Tashiro, Y., Song, J., Song, Y. & Ermon, S. CSDI: conditional score-based diffusion models for probabilistic time series imputation. *Adv. Neural Inf. Process. Syst.* **34**, 24804–24816 (2021).
81. Yan, T., Zhang, H., Zhou, T., Zhan, Y. & Xia, Y. ScoreGrad: multivariate probabilistic time series forecasting with continuous energy-based generative models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2106.10121 (2021).
82. Avrahami, O., Lischinski, D. & Fried, O. Blended diffusion for text-driven editing of natural images. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* 18208–18218 (IEEE, 2022).
83. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2204.06125 (2022).
84. Anand, N. & Achim, T. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2205.15019 (2022).
85. Cao, C., Cui, Z.-X., Liu, S., Liang, D. & Zhu, Y. High-frequency space diffusion models for accelerated MRI. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2208.05481 (2022).
86. Chung, H., Lee, E. S. & Ye, J. C. MR image denoising and super-resolution using regularized reverse diffusion. *IEEE Trans. Med. Imaging* **42**, 922–934 (2022).
87. Chung, H., Sim, B. & Ye, J. C. Come-closer-diffuse-faster: accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* 12413–12422 (IEEE, 2022).
88. Chung, H. & Ye, J. C. Score-based diffusion models for accelerated MRI. *Med. Image Anal.* **80**, 102479 (2022).
89. Güngör, A. et al. Adaptive diffusion priors for accelerated MRI reconstruction. *Med. Image Anal.* **88**, 102872 (2023).
90. Jing, B., Corso, G., Chang, J., Barzilay, R. & Jaakkola, T. Torsional Diffusion for Molecular Conformer Generation. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 24240–24253 (Curran Associates, 2022).
91. Lee, J. S. & Kim, P. M. ProteinSGM: score-based generative modeling for de novo protein design. Preprint at *bioRxiv* https://doi.org/10.1101/2022.07.13.499967 (2022).
92. Luo, S. et al. Antigen-Specific Antibody Design and Optimization with Diffusion-Based Generative Models for Protein Structures. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 9754–9767 (Curran Associates, 2022).
93. Mei, S., Fan, F. & Maier, A. Metal inpainting in CBCT projections using score-based generative model. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2209.09733 (2022).
94. Du, Y. & Mordatch, I. Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems* Vol. 32 (eds Wallach, H. et al.) (Curran Associates, 2019).
95. Brock, A., Donahue, J. & Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1809.11096 (2018).
96. Karras, T. et al. Training generative adversarial networks with limited data. *Adv. Neural Inf. Process. Syst.* **33**, 12104–12114 (2020).
97. Song, J., Meng, C. & Ermon, S. Denoising diffusion implicit models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2010.02502 (2020).
98. Kreis, K., Dockhorn, T., Li, Z. & Zhong, E. Latent space diffusion models of cryo-EM structures. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2211.14169 (2022).
99. Waibel, D. J., Röell, E., Rieck, B., Giryes, R. & Marr, C. A diffusion model predicts 3D shapes from 2D microscopy images. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2208.14125 (2022).
100. Tjärnberg, A. et al. Optimal tuning of weighted kNN- and diffusion-based methods for denoising single cell genomics data. *PLoS Comput. Biol.* **17**, e1008569 (2021).
101. Wu, K. E. et al. Protein structure generation via folding diffusion. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2209.15611 (2022).
102. Gao, Z., Tan, C. & Li, S. Z. DiffSDS: a language diffusion model for protein backbone inpainting under geometric conditions and constraints. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2301.09642 (2023).
103. Lin, Y. & AlQuraishi, M. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2301.12485 (2023).
104. Trippe, B. L. et al. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2206.04119 (2022).
105. Watson, J. L. et al. De novo design of protein structure and function with RFdiffusion. *Nature* **620**, 1089–1100 (2023).
This article presents RFdiffusion, which can be applied to complex protein-generation tasks.
106. Yim, J. et al. SE(3) diffusion model with application to protein backbone generation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2302.02277 (2023).
107. Ingraham, J. et al. Illuminating protein space with a programmable generative model. Preprint at *bioRxiv* https://doi.org/10.1101/2022.12.01.518682 (2022).
This article reports the graph-neural-network-based conditional diffusion model Chroma, which can generate large single-chain proteins and protein complexes with programmable properties and functions.
108. Huang, H., Sun, L., Du, B. & Lv, W. Conditional diffusion based on discrete graph structures for molecular graph generation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2301.00427 (2023).
109. Wu, L., Gong, C., Liu, X., Ye, M. & Liu, Q. Diffusion-based Molecule Generation with Informative Prior Bridges. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 36533–36545 (Curran Associates, 2022).
110. Luo, S., Shi, C., Xu, M. & Tang, J. Predicting molecular conformation via dynamic graph score matching. *Adv. Neural Inf. Process. Syst.* **34**, 19784–19795 (2021).
111. Zhang, H. et al. SDEGen: learning to evolve molecular conformations from thermodynamic noise for conformation generation. *Chem. Sci.* **14**, 1557–1568 (2023).
112. Wu, F. & Li, S. Z. DIFFMD: a geometric diffusion model for molecular dynamics simulations. In *Proc. AAAI Conference Artificial Intelligence* **37**, 5321–5329 (2003).
113. Igashov, I. et al. Equivariant 3D-conditional diffusion models for molecular linker design. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2210.05274 (2022).
114. Lin, H. et al. DiffBP: generative diffusion of 3D molecules for target protein binding. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2211.11214 (2022).
115. Schneuing, A. et al. Structure-based drug design with equivariant diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2210.13695 (2022).
116. Corso, G., Stärk, H., Jing, B., Barzilay, R. & Jaakkola, T. DiffDock: diffusion steps, twists, and turns for molecular docking. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2210.01776 (2022).
This article presents the diffusion model DiffDock for protein pocket docking.
117. Qiao, Z., Nie, W., Vahdat, A., Miller III, T. F. & Anandkumar, A. Dynamic-backbone protein–ligand structure prediction with multiscale generative diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2209.15171 (2022).
118. Jin, W., Sarkizova, S., Chen, X., Hacohen, N. & Uhler, C. Unsupervised protein–ligand binding energy prediction via neural Euler's rotation equation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2301.10814 (2023).
119. Song, Y. & Ermon, S. Improved techniques for training score-based generative models. *Adv. Neural Inf. Process. Syst.* **33**, 12438–12448 (2020).
120. Song, Y., Durkan, C., Murray, I. & Ermon, S. Maximum likelihood training of score-based diffusion models. *Adv. Neural Inf. Process. Syst.* **34**, 1415–1428 (2021).
121. Hyvärinen, A. & Dayan, P. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.* **6**, 95–709 (2005).
122. Raphan, M. & Simoncelli, E. P. Least squares estimation without priors or supervision. *Neural Comput.* **23**, 374–420 (2011).
123. Raphan, M. & Simoncelli, E. Learning to be Bayesian without Supervision. In *Advances in Neural Information Processing Systems* Vol. 19 (eds Scholkopf, B., Platt, J. & Hoffman, T) (MIT Press, 2006).
124. Vincent, P. A connection between score matching and denoising autoencoders. *Neural Comput.* **23**, 1661–1674 (2011).
125. Song, Y., Garg, S., Shi, J. & Ermon, S. Sliced Score Matching: A Scalable Approach to Density and Score Estimation. In *Proc. 35th Uncertainty in Artificial Intelligence Conference* Vol. 115 (eds Adams R., & Gogate, V.) 574–584 (PMLR, 2020).
126. Kingma, D., Salimans, T., Poole, B. & Ho, J. Variational diffusion models. *Adv. Neural Inf. Process. Syst.* **34**, 21696–21707 (2021).
127. Luo, C. Understanding diffusion models: a unified perspective. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2208.11970 (2022).
128. Arnold, L. *Stochastic Differential Equations* (Wiley, 1974).
129. Anderson, B. D. Reverse-time diffusion equation models. *Stoch. Process. Appl.* **12**, 313–326 (1982).
130. Nichol, A. Q. & Dhariwal, P. Improved Denoising Diffusion Probabilistic Models. In *Proc. 38th Int. Conference on Machine Learning* Vol. 139 (eds Meila, M. & Zhang, T.) 8162–8171 (PMLR, 2021).
131. Bansal, A. et al. Cold diffusion: inverting arbitrary image transforms without noise. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2208.09392 (2022).
132. Kong, Z. & Ping, W. On fast sampling of diffusion probabilistic models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2106.00132 (2021).
133. Salimans, T. & Ho, J. Progressive distillation for fast sampling of diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2202.00512 (2022).
134. Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T. & Mitliagkas, I. Gotta go fast when generating data with score-based models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2105.14080 (2021).
135. Karras, T., Aittala, M., Aila, T. & Laine, S. Elucidating the Design Space of Diffusion-Based Generative Models. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 26565–26577 (Curran Associates, 2022).
136. Lu, C. et al. DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 5775–5787 (Curran Associates, 2022).
137. Liu, L., Ren, Y., Lin, Z. & Zhao, Z. Pseudo numerical methods for diffusion models on manifolds. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2202.09778 (2022).
138. Bao, F., Li, C., Zhu, J. & Zhang, B. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2201.06503 (2022).
139. Lu, C. et al. DPM-solver++: fast solver for guided sampling of diffusion probabilistic models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2211.01095 (2022).
140. Vahdat, A., Kreis, K. & Kautz, J. Score-based generative modeling in latent space. *Adv. Neural Inf. Process. Syst.* **34**, 11287–11302 (2021).

141. Zhang, Q. & Chen, Y. Diffusion normalizing flow. *Adv. Neural Inf. Process. Syst.* **34**, 16280–16291 (2021).
142. Pandey, K., Mukherjee, A., Rai, P. & Kumar, A. DiffuseVAE: efficient, controllable and high-fidelity generation from low-dimensional latents. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2201.00308 (2022).
143. Luo, S. & Hu, W. Diffusion probabilistic models for 3D point cloud generation. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* 2837–2845 (IEEE, 2021).
144. Shi, C., Luo, S., Xu, M. & Tang, J. Learning Gradient Fields for Molecular Conformation Generation. In *Proc. 38th Int. Conference on Machine Learning* Vol. 139 (ed Meila, M. & Zhang, T.) 9558–9568 (PMLR, 2021).
145. Zhou, L., Du, Y. & Wu, J. 3D shape generation and completion through point-voxel diffusion. In *Proc. IEEE/CVF Int. Conf. on Computer Vision* 5826–5835 (IEEE, 2021).
146. Hoogeboom, E. et al. Autoregressive diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2110.02037 (2021).
147. Xu, M. et al. GeoDiff: a geometric diffusion model for molecular conformation generation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2203.02923 (2022).
148. Jo, J., Lee, S. & Hwang, S. J. Score-based Generative Modeling of Graphs via the System of Stochastic Differential Equations. In *Proc. 39th Int. Conference on Machine Learning* Vol. 162 (eds Chaudhuri, K. et al.) 10362–10383 (PMLR, 2022).
149. De Bortoli, V. et al. Riemannian score-based generative modelling. *Adv. Neural Inf. Process.* **35**, 2406–2422 (2022).
150. Chen, T., Zhang, R. & Hinton, G. Analog bits: generating discrete data using diffusion models with self-conditioning. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2208.04202 (2022).
151. Niu, C. et al. Permutation Invariant Graph Generation via Score-Based Generative Modeling. In *Proc. 23rd Int. Conference on Artificial Intelligence and Statistics* Vol. 108 (eds Chiappa, S. & Calandra, R.) 4474–4484 (PMLR, 2020).
152. Yang, L. et al. Diffusion models: a comprehensive survey of methods and applications. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2209.00796 (2022).
153. Anand, N. & Huang, P. Generative modeling for protein structures. In *Advances in Neural Information Processing Systems* Vol. 31 (eds Bengio, S. et al.) (Curran Associates, 2018).
154. Lin, Z., Sercu, T., LeCun, Y. & Rives, A. Deep generative models create new and diverse protein structures. In *Machine Learning for Structural Biology Workshop, NeurIPS* (2021).
155. Eguchi, R. R., Choe, C. A. & Huang, P.-S. Ig-VAE: generative modeling of protein structure by direct 3D coordinate generation. *PLoS Comput. Biol.* **18**, e1010271 (2022).
156. Anishchenko, I. et al. De novo protein design by deep network hallucination. *Nature* **600**, 547–552 (2021).
157. Greener, J. G., Moffat, L. & Jones, D. T. Design of metalloproteins and novel protein folds using variational autoencoders. *Sci. Rep.* **8**, 16189 (2018).
158. Anand, N., Eguchi, R. & Huang, P.-S. Fully differentiable full-atom protein backbone generation. In *Proc. Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop* (OpenReview.net, 2019).
159. Karimi, M., Zhu, S., Cao, Y. & Shen, Y. De novo protein design for novel folds using guided conditional Wasserstein generative adversarial networks. *J. Chem. Inf. Model* **60**, 5667–5681 (2020).
160. Simons, K. T., Bonneau, R., Ruczinski, I. & Baker, D. Ab initio protein structure prediction of CASP III targets using ROSETTA. *Proteins Struct. Funct. Bioinform.* **37**, 171–176 (1999).
161. Satorras, V. c. G., Hoogeboom, E. & Welling, M. E(n) Equivariant Graph Neural Networks. In *Proc. 38th Int. Conference on Machine Learning* Vol. 139 (eds Meila, M. & Zhang, T.) 9323–9332 (PMLR, 2021).
162. Thomas, N. et al. Tensor field networks: rotation-and translation-equivariant neural networks for 3D point clouds. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1802.08219 (2018).
163. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
164. Anand, N. et al. Protein sequence design with a learned potential. *Nat. Commun.* **13**, 746 (2022).
165. Leaver-Fay, A. et al. Scientific benchmarks for guiding macromolecular energy function improvement. *Methods Enzymol.* **523**, 109–143 (2013).
166. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
167. Chène, P. Inhibiting the p53–MDM2 interaction: an important target for cancer therapy. *Nat. Rev. Cancer* **3**, 102–109 (2003).
168. Salgado, E. N., Lewis, R. A., Mossin, S., Rheingold, A. L. & Tezcan, F. A. Control of protein oligomerization symmetry by metal coordination: C2 and C3 symmetrical assemblies through CuII and NiII coordination. *Inorg. Chem.* **48**, 2726–2728 (2009).
169. Salgado, E. N. et al. Metal templated design of protein interfaces. *Proc. Natl Acad. Sci.* **107**, 1827–1832 (2010).
170. Cao, L. et al. Design of protein-binding proteins from the target structure alone. *Nature* **605**, 551–560 (2022).
171. Chevalier, A. et al. Massively parallel de novo protein design for targeted therapeutics. *Nature* **550**, 74–79 (2017).
172. Gómez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
173. De Cao, N. & Kipf, T. MolGAN: an implicit generative model for small molecular graphs. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1805.11973 (2018).
174. Wu, Z. et al. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).
175. You, J., Liu, B., Ying, Z., Pande, V. & Leskovec, J. Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation. In *Advances in Neural Information Processing Systems* Vol. 31 (eds Bengio, S. et al.) (Curran Associates, 2018).
176. Kloeden, P. E., Platen, E., Kloeden, P. E. & Platen, E. *Stochastic Differential Equations* (Springer, 1992).
177. Shi, C. et al. GraphAF: a flow-based autoregressive model for molecular graph generation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2001.09382 (2020).
178. Luo, Y., Yan, K. & Ji, S. GraphDF: A Discrete Flow Model for Molecular Graph Generation. In *Proc. 38th Int. Conference on Machine Learning* Vol. 139 (eds Meila, M. & Zhang, T.) 7192–7203 (PMLR, 2021).
179. Zang, C. & Wang, F. MoFlow: an invertible flow model for generating molecular graphs. In *Proc. 26th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining* 617–626 (2020).
180. Lippe, P. & Gavves, E. Categorical normalizing flows via continuous transformations. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2006.09790 (2020).
181. Liu, M., Yan, K., Oztekin, B. & Ji, S. G. Molecular graph generation with energy-based models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2102.00546 (2021).
182. Erdős, P. & Rényi, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **5**, 17–60 (1960).
183. Kipf, T. N. & Welling, M. Variational graph auto-encoders. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1611.07308 (2016).
184. You, J., Ying, R., Ren, X., Hamilton, W. & Leskovec, J. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. In *Proc. 35th Int. Conference on Machine Learning* Vol. 80 (eds Dy, J & Krause, A.) 5708–5717 (PMLR, 2018).
185. Liao, R. et al. Efficient Graph Generation with Graph Recurrent Attention Networks. In *Advances in Neural Information Processing Systems* Vol. 32 (eds Wallach, H. et al.) (Curran Associates, 2019).
186. Garcia Satorras, V., Hoogeboom, E., Fuchs, F., Posner, I. & Welling, M. E(n) Equivariant Normalizing Flows. In *Advances in Neural Information Processing Systems* Vol. 34 (eds Ranzato, M. et al.) 4181–4192 (Curran Associates, 2021).
187. Gebauer, N., Gastegger, M. & Schütt, K. Symmetry-adapted generation of 3D point sets for the targeted discovery of molecules. In *Advances in Neural Information Processing Systems* Vol. 32 (eds Wallach, H. et al.) (Curran Associates, 2019).
188. Simonovsky, M. & Komodakis, N. Graphvae: towards generation of small graphs using variational autoencoders. In *Artificial Neural Networks and Machine Learning (ICANN 2018) 27th Int. Conf. on Artificial Neural Networks Proc. Part I* 412–422 (Springer, 2018).
189. Mitton, J., Senn, H. M., Wynne, K. & Murray-Smith, R. A graph VAE graph transformer approach to generating molecular graphs. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2104.04345 (2021).
190. Vignac, C. & Frossard, P. Top-N: equivariant set and graph generation without exchangeability. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2110.02096 (2021).
191. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural Message Passing for Quantum Chemistry. In *Proc. 34th Int. Conference on Machine Learning* Vol. 70 (eds Precup, D. & Teh, Y.) 1263–1272 (PMLR, 2017).
192. Riniker, S. & Landrum, G. A. Better informed distance geometry: using what we know to improve conformation generation. *J. Chem. Inf. Model* **55**, 2562–2574 (2015).
193. Xu, M., Luo, S., Bengio, Y., Peng, J. & Tang, J. Learning neural generative dynamics for molecular conformation generation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2102.10240 (2021).
194. Simm, G. N. & Hernández-Lobato, J. M. A generative model for molecular distance geometry. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1909.11459 (2019).
195. Mansimov, E., Mahmood, O., Kang, S. & Cho, K. Molecular geometry prediction using a deep generative graph neural network. *Sci. Rep.* **9**, 20381 (2019).
196. Zhu, J. et al. Direct molecular conformation generation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2202.01356 (2022).
197. Köhler, J., Klein, L. & Noé, F. Equivariant flows: sampling configurations for multi-body systems with symmetric energies. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1910.00753 (2019).
198. Fuchs, F., Worrall, D., Fischer, V. & Welling, M. Se(3)-transformers: 3D roto-translation equivariant attention networks. *Adv. Neural Inf. Process. Syst.* **33**, 1970–1981 (2020).
199. Huang, W. et al. Equivariant graph mechanics networks with constraints. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2203.06442 (2022).
200. Gao, A. & Remsing, R. C. Self-consistent determination of long-range electrostatics in neural network potentials. *Nat. Commun.* **13**, 1572 (2022).
201. Imrie, F., Bradley, A. R., van der Schaar, M. & Deane, C. M. Deep generative models for 3D linker design. *J. Chem. Inf. Model* **60**, 1983–1995 (2020).
202. Huang, Y., Peng, X., Ma, J. & Zhang, M. 3DLinker: an E(3) equivariant variational autoencoder for molecular linker design. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2205.07309 (2022).
203. Liu, M., Luo, Y., Uchino, K., Maruhashi, K. & Ji, S. Generating 3D molecules for target protein binding. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2204.09410 (2022).
204. Masuda, T., Ragoza, M. & Koes, D. R. Generating 3D molecular structures conditional on a receptor binding site with deep generative models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2010.14442 (2020).
205. Luo, S., Guan, J., Ma, J. & Peng, J. A 3D generative model for structure-based drug design. *Adv. Neural Inf. Process. Syst.* **34**, 6229–6239 (2021).

206. Peng, X. et al. Pocket2Mol: Efficient Molecular Sampling Based on 3D Protein Pockets. In *Proc. 39th Int. Conference on Machine Learning* Vol. 162 (eds Chaudhuri, K. et al.) 17644–17655 (PMLR, 2022).
207. Jing, B., Eismann, S., Soni, P. N. & Dror, R. O. Equivariant graph neural networks for 3D macromolecular structure. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2106.03843 (2021).
208. Francoeur, P. G. et al. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. *J. Chem. Inf. Model.* **60**, 4200–4215 (2020).
209. Koes, D. R., Baumgartner, M. P. & Camacho, C. J. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *J. Chem. Inf. Model* **53**, 1893–1904 (2013).
210. Hassan, N. M., Alhossary, A. A., Mu, Y. & Kwoh, C.-K. Protein–ligand blind docking using QuickVina-W with inter-process spatio-temporal integration. *Sci. Rep.* **7**, 15451 (2017).
211. Friesner, R. A. et al. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J. Med. Chem.* **47**, 1739–1749 (2004).
212. McNutt, A. T. et al. GNINA 1.0: molecular docking with deep learning. *J. Cheminform.* **13**, 43 (2021).
213. Strk, H., Ganea, O., Pattanaik, L., Barzilay, D. R. & Jaakkola, T. EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction. In *Proc. 39th Int. Conference on Machine Learning* Vol. 162 (eds Chaudhuri, K. et al.) 20503–20521 (PMLR, 2022).
214. Lu, W. et al. TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 7236–7249 (Curran Associates, 2022).
215. Liu, Y. et al. CB-Dock: a web server for cavity detection-guided protein–ligand blind docking. *Acta Pharmacol. Sin.* **41**, 138–144 (2020).
216. Wang, R., Fang, X., Lu, Y. & Wang, S. The PDBbind database: collection of binding affinities for protein–ligand complexes with known three-dimensional structures. *J. Med. Chem.* **47**, 2977–2980 (2004).
217. Dunbar, J. et al. SAbDab: the structural antibody database. *Nucleic Acids Res.* **42**, D1140–D1146 (2014).
218. Miller, B. R. III et al. MMPBSA.py: an efficient program for end-state free energy calculations. *J. Chem. Theory Comput.* **8**, 3314–3321 (2012).
219. Mooij, W. T. & Verdonk, M. L. General and targeted statistical potentials for protein–ligand interactions. *Proteins* **61**, 272–287 (2005).
220. Dittrich, J., Schmidt, D., Pfleger, C. & Gohlke, H. Converging a knowledge-based scoring function: DrugScore2018. *J. Chem. Inf. Model* **59**, 509–521 (2018).
221. Pierce, B. & Weng, Z. ZRANK: reranking protein docking predictions with an optimized energy function. *Proteins* **67**, 1078–1086 (2007).
222. Pierce, B. & Weng, Z. A combination of rescoring and refinement significantly improves protein docking performance. *Proteins* **72**, 270–279 (2008).
223. Alford, R. F. et al. The Rosetta all-atom energy function for macromolecular modeling and design. *J. Chem. Theory Comput.* **13**, 3031–3048 (2017).
224. Grosdidier, S., Pons, C., Solernou, A. & Fernández-Recio, J. Prediction and scoring of docking poses with pyDock. *Proteins* **69**, 852–858 (2007).
225. Pons, C., Talavera, D., De La Cruz, X., Orozco, M. & Fernandez-Recio, J. Scoring by intermolecular pairwise propensities of exposed residues (SIPPER): a new efficient potential for protein–protein docking. *J. Chem. Inf. Model* **51**, 370–377 (2011).
226. Viswanath, S., Ravikant, D. & Elber, R. Improving ranking of models for protein complexes with side chain modeling and atomic potentials. *Proteins* **81**, 592–606 (2013).
227. Ravikant, D. & Elber, R. PIE—efficient filters and coarse grained potentials for unbound protein–protein docking. *Proteins* **78**, 400–419 (2010).
228. Andrusier, N., Nussinov, R. & Wolfson, H. J. FireDock: fast interaction refinement in molecular docking. *Proteins* **69**, 139–159 (2007).
229. Dubochet, J. et al. Cryo-electron microscopy of vitrified specimens. *Q. Rev. Biophys.* **21**, 129–228 (1988).
230. Frank, J. et al. SPIDER and WEB: processing and visualization of images in 3D electron microscopy and related fields. *J. Struct. Biol.* **116**, 190–199 (1996).
231. Ludtke, S. J., Baldwin, P. R. & Chiu, W. EMAN: semiautomated software for high-resolution single-particle reconstructions. *J. Struct. Biol.* **128**, 82–97 (1999).
232. Scheres, S. H. RELION: implementation of a Bayesian approach to cryo-EM structure determination. *J. Struct. Biol.* **180**, 519–530 (2012).
233. Nogales, E. & Scheres, S. H. Cryo-EM: a unique tool for the visualization of macromolecular complexity. *Mol. Cell* **58**, 677–689 (2015).
234. Fernandez-Leiro, R. & Scheres, S. H. Unravelling biological macromolecules with cryo-electron microscopy. *Nature* **537**, 339–346 (2016).
235. Merk, A. et al. Breaking cryo-EM resolution barriers to facilitate drug discovery. *Cell* **165**, 1698–1707 (2016).
236. Zhong, E. D., Bepler, T., Davis, J. H. & Berger, B. Reconstructing continuous distributions of 3D protein structure from cryo-EM images. Preprint at *arXiv* https://doi.org/10.48550/arXiv.1909.05215 (2019).
237. 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 Int. Conf. Proc. Part III* 234–241 (Springer, 2015).
238. Waibel, D. J. E. et al. SHAPR—an AI approach to predict 3D cell shapes from 2D microscopic images. *iScience* https://doi.org/10.1016/j.isci.2022.105298 (2022).
239. Waibel, D. J., Atwell, S., Meier, M., Marr, C. & Rieck, B. Capturing shape information with multi-scale topological loss terms for 3D reconstruction. In *Medical Image Computing and Computer Assisted Intervention (MICCAI 2022) 25th Int. Conf. Proc. Part IV* 150–159 (Springer, 2022).
240. Van Dijk, D. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729.e727 (2018).
241. Wagner, F., Yan, Y. & Yanai, I. *K*-nearest neighbor smoothing for high-throughput single-cell RNA-seq data. Preprint at *bioRxiv* https://doi.org/10.1101/217737 (2017).
242. Lieberman-Aiden, E. et al. Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**, 289–293 (2009).
243. Trieu, T. & Cheng, J. 3D genome structure modeling by Lorentzian objective function. *Nucleic Acids Res.* **45**, 1049–1058 (2017).
244. Trieu, T. & Cheng, J. MOGEN: a tool for reconstructing 3D models of genomes from chromosomal conformation capturing data. *Bioinformatics* **32**, 1286–1292 (2016).
245. Highsmith, M. & Cheng, J. VEHiCLE: a variationally encoded Hi-C loss enhancement algorithm for improving and generating Hi-C data. *Sci. Rep.* **11**, 1–13 (2021).
246. Wang, Y., Guo, Z. & Cheng, J. Single-cell Hi-C data enhancement with deep residual and generative adversarial networks. *Bioinformatics* **39**, btad458 (2023).
247. Taskiran, I. I., Spanier, K. I., Christiaens, V., Mauduit, D. & Aerts, S. Cell type directed design of synthetic enhancers. Preprint at *bioRxiv* https://doi.org/10.1101/2022.07.26.501466 (2022).
248. Tang, G. et al. EMAN2: an extensible image processing suite for electron microscopy. *J. Struct. Biol.* **157**, 38–46 (2007).
249. Al-Azzawi, A. et al. DeepCryoPicker: fully automated deep neural network for single protein particle picking in cryo-EM. *BMC Bioinform.* **21**, 1–38 (2020).
250. Kawar, B., Elad, M., Ermon, S. & Song, J. Denoising diffusion restoration models. *Adv. Neural Inf. Process. Syst.* **35**, 23593–23606 (2022).
251. Evans, R. et al. Protein complex prediction with AlphaFold-Multimer. Preprint at *bioRxiv* https://doi.org/10.1101/2021.10.04.463034 (2021).
252. Huang, C.-W., Lim, J. H. & Courville, A. C. A variational perspective on diffusion-based generative models and score matching. *Adv. Neural Inf. Process. Syst.* **34**, 22863–22876 (2021).
253. Kim, D., Shin, S., Song, K., Kang, W. & Moon, I.-C. Soft truncation: a universal training technique of score-based diffusion model for High Precision Score Estimation. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2106.05527 (2021).
254. Gu, S. et al. Vector quantized diffusion model for text-to-image synthesis. In *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* 10696–10706 (IEEE, 2022).
255. Tang, Z., Gu, S., Bao, J., Chen, D. & Wen, F. Improved vector quantized diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2205.16007 (2022).
256. Poole, B., Jain, A., Barron, J. T. & Mildenhall, B. DreamFusion: text-to-3D using 2D diffusion. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2209.14988 (2022).
257. Hong, S., Lee, G., Jang, W. & Kim, S. Improving sample quality of diffusion models using self-attention guidance. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2210.00939 (2022).
258. Li, W. Automatic segmentation of liver tumor in CT images with deep convolutional neural networks. *J. Comput. Commun.* **3**, 146 (2015).
259. Menze, B. H. et al. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Trans. Med. Imaging* **34**, 1993–2024 (2014).
260. Cheng, J. et al. Superpixel classification based optic disc and optic cup segmentation for glaucoma screening. *IEEE Trans. Med. Imaging* **32**, 1019–1032 (2013).
261. Wang, S. et al. Central focused convolutional neural networks: developing a data-driven model for lung nodule segmentation. *Med. Image Anal.* **40**, 172–183 (2017).
262. Srinivasu, P. N. et al. Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. *Sensors* **21**, 2852 (2021).
263. Swapna, G., Vinayakumar, R. & Soman, K. Diabetes detection using deep learning algorithms. *ICT Express* **4**, 243–246 (2018).
264. Das, A., Acharya, U. R., Panda, S. S. & Sabut, S. Deep learning based liver cancer detection using watershed transform and Gaussian mixture model techniques. *Cogn. Syst. Res.* **54**, 165–175 (2019).
265. Jo, T., Nho, K. & Saykin, A. J. Deep learning in Alzheimer's disease: diagnostic classification and prognostic prediction using neuroimaging data. *Front. Aging Neurosci.* **11**, 220 (2019).
266. Arévalo, A., Niño, J., Hernández, G. & Sandoval, J. High-frequency trading strategy based on deep neural networks. In *Int. Conf. on Intelligent Computing* 424–436 (Springer, 2016).
267. Bao, W., Yue, J. & Rao, Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS One* **12**, e0180944 (2017).
268. Xiao, Q., Li, K., Zhang, D. & Xu, W. Security risks in deep learning implementations. In *2018 IEEE Security and Privacy Workshops (SPW)* 123–128 (IEEE, 2018).
269. Halstead, M., Ahmadi, A., Smitt, C., Schmittmann, O. & McCool, C. Crop agnostic monitoring driven by deep learning. *Front. Plant. Sci.* **12**, 786702 (2021).
270. Feng, A., Zhou, J., Vories, E. & Sudduth, K. A. Evaluation of cotton emergence using UAV-based imagery and deep learning. *Comput. Electron. Agric.* **177**, 105711 (2020).
271. Liu, J. & Wang, X. Plant diseases and pests detection based on deep learning: a review. *Plant. Methods* **17**, 22 (2021).
272. Brown, T. et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).
273. Nichol, A. et al. Glide: towards photorealistic image generation and editing with text-guided diffusion models. Preprint at *arXiv* https://doi.org/10.48550/arXiv.2112.10741 (2021).

Acknowledgements
The work was partly supported by the US National Institutes of Health (grants R01GM146340 (to J.C.), R01GM093123 (to J.C.) and R35GM126985 (to D.X.)) and the US National Science Foundation (grant DBI2308699 (to J.C.)).

Author contributions
Z.G., J.L., Y.W. and M.C. collected data. J.C., D.X. and D.W. provided guidance on organizing the content. J.C. envisioned the developments for proteins, 3D genomics, single-cell Hi-C data analytics, cryo-EM, DNA design, peptide, proteomics and metabolomics. D.X. envisioned the developments in single-cell reconstruction and inference. Z.G., J.C., J.L., Y.W., D.X., D.W. and M.C. wrote and edited the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Peer review information *Nature Reviews Bioengineering* thanks Bonnie Berger, Daniel Lazarev and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.