Score-based Enhanced Sampling for Protein Molecular Dynamics

Jiarui Lu[1,2*] **Bozitao Zhong**[1,2*] **Jian Tang**[1,3,4]
[1]Mila - Québec AI Institute [2]Université de Montréal
[3]HEC Montréal [4]CIFAR AI Research Chair
<jiarui.lu, bozitao.zhong>@umontreal.ca, jian.tang@hec.ca

Abstract

The dynamic nature of proteins is crucial for determining their biological functions and properties, and molecular dynamics (MD) simulations stand as a predominant tool to study such phenomena. By utilizing empirically derived force fields, MD simulations explore the conformational space through numerically evolving the system along MD trajectories. However, the high-energy barrier of the force fields can hamper the exploration of MD, resulting in inadequately sampled ensemble. In this paper, we propose leveraging score-based generative models (SGMs) trained on general protein structures to perform protein conformational sampling to complement traditional MD simulations. We argue that SGMs can provide a novel framework as an alternative to traditional enhanced sampling methods by learning multi-level score functions, which directly sample a diversity-controllable ensemble of conformations. We demonstrate the effectiveness of our approach on several benchmark systems by comparing the results with long MD trajectories and state-of-the-art generative structure prediction models. Our framework provides new insights that SGMs have the potential to serve as an efficient and simulation-free methods to study protein dynamics.

1 Introduction

Understanding the dynamical properties of protein is crucial for elucidating the structural mechanism of their biological functions and regulations. Transitions can exist in the conformational ensembles which proteins populate, ranging from angstrom to nanometer in length, and from nanosecond to second in time. Experimental measurements, such as crystallographic B-factors and NMR spectroscopy, can be performed to probe such dynamics. However, these are limited in spatial and temporal scale. Despite the success of structure prediction methods [5, 32, 37], which enables the study of proteins based on their high-accuracy structures, the predicted conformational ensembles often lack diversity [12, 48].

Molecular dynamics (MD) serves as a physics-based tool for studying the protein dynamics by employing an empirical force field for simulation. MD simulations evolve the protein structure motion over time to generate ensemble according to the equilibrium distribution, which can be further analyzed to quantify functions of interest. A significant challenge encountered by MD simulations is the high energy-barriers, which forbid transitions within a limited number of simulation steps. Specifically, two typical conformational states with high probability density (low energy) often lie on either side of a high energy barrier. Consequently, transitions between these states can only be achieved with sufficiently long simulation time, otherwise the trajectories remain trapped within the same energy well, resulting in limited exploration.

*Equal contribution.

Preprint. Under review.

Over the past decades, enhanced sampling methods have been proposed to overcome the energy barrier and encourage more exploration of MD simulations [1]. These methods in general fall into two categories: **(1)** collective variables (CV)-based approaches, such as umbrella sampling [62] and metadynamics [35]; **(2)** tempering-based (or CV-free) approaches such as simulated tempering [41] and parallel tempering (synonymous replica exchange molecular dynamics, REMD) [24, 59, 60], where the term "tempering" refers to methods that involve increasing the temperature of the simulated system to overcome energy barriers [1]. The CV-based methods require pre-defined collective variables, or reaction coordinates, which are a low-dimension representation describing the motion of interest using particle coordinates of the system. However, defining proper collective variables is usually challenging for many real systems [74]. In contrast, tempering-based methods operate by scheduling the system's temperature to facilitate barrier-crossing transitions [1], which borrows the idea from classical simulated annealing approaches for optimization.

Inspired by tempering-based enhanced sampling methods [41], we leverage the score-based generative models (SGMs) [26, 55, 58] and present the **S**core-based **EN**hanced **S**ampling (**SENS**), a score-based framework for protein conformational sampling which is trained on general protein structures from Protein Data Bank (PDB) [8]. SENS operates by dispersing the input conformation to its geometric neighborhood and then annealing back the perturbed conformations into respective equilibrium states with the learned score functions. This approach enables directly sampling of diverse conformation ensembles, thus effectively circumventing the high-energy barrier issue in traditional MD simulations. Moreover, SENS does not rely on any specific simulation data for training, and can be transferred to perform zero-shot conformational sampling by an amortized training. To assess the effectiveness of the SENS, we evaluate it on several benchmark systems, comparing the results with long MD simulations and state-of-the-art generative structure prediction models. The performance of SENS demonstrates its potential as a promising tool for enhanced protein conformational sampling.

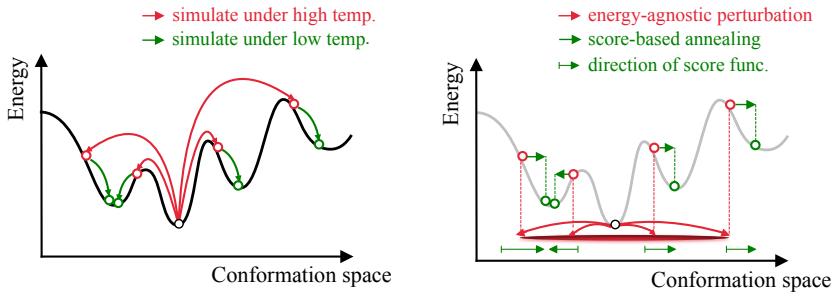


Figure 1: The illustrative comparison between (a) tempering-based enhanced sampling versus (b) sampling via SENS. **(a)** The initial conformations first go through high-temperature MD simulations such that they may "jump out" (red) with large kinetic energy, followed by low-temperature MD steps (green). The motion is highly dependent on the energy surface and hampered by potentially high-energy barrier; **(b)** The initial conformations are randomly dispersed by a perturbation kernel (red). Then the noisy conformations find their respective equilibrium guided by the learned score functions (green). By contrast with (a), these two operations are both energy-agnostic.

2 Preliminary: Score-based generative models

Score-based generative models (SGMs) can be represented by a diffusion process $\mathbf{x}(t) \in \mathbb{R}^d$ defined by the Itô stochastic differential equation(SDE):

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}, \tag{1}$$

with continuous time index $t \in [0, T]$, where the $\mathbf{f}(\mathbf{x}, t) \in \mathbb{R}^d$ is the time-dependent drift coefficient, $g(t) \in \mathbb{R}$ is the diffusion coefficient, and $\mathbf{w} \in \mathbb{R}^d$ is the standard Wiener process. Then, the corresponding backward SDE that describes the dynamics from $\mathbf{x}(T)$ to $\mathbf{x}(0)$ is [3, 58]:

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]dt + g(t)d\mathbf{w}, \tag{2}$$

where dt is negative infinitesimal timestep and $\mathbf{w}$ is the standard Wiener process as continuous time t flows back from T to 0.

Different drift and diffusion coefficient can be defined to steer the diffusion process. Two widely used SDE-based schemes [58] are (1) Variance Exploding (VE) SDE which is derived from score matching with Langevin dynamics (SMLD) [57]:

$$d\mathbf{x} = \sqrt{\frac{d[\sigma^2(t)]}{dt}} d\mathbf{w}, \tag{3}$$

and (2) Variance Preserving (VP) SDE from denoising diffusion probabilistic models (DDPM) [26, 55]:

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}, \tag{4}$$

where $\sigma(t)$ and $\beta(t)$ are pre-defined noise schedules to perturb the data.

3 Proposed method: Score-based enhanced sampling

Suppose we have an observed conformation of a protein (we shall identify it with subscript $[p]$) denoted by $\mathbf{x}_{[p]} \in \mathbb{R}^{3N_{[p]}}$, where $N_{[p]}$ is the total number of atoms of interest. Our goal is to obtain a diverse conformation ensemble $\tilde{\mathbf{X}}^*_{[p]} = \{\tilde{\mathbf{x}}^{(i)}_{[p]}\}$ (members indexed by $i \in \Lambda$) that well captures the dynamics of p with $\mathbf{x}_{[p]}$ as starting point. To achieve this, consider two stochastic operators: (1) *heat* operator $\mathbf{H}_\epsilon : \mathbb{R}^{3N_{[p]}\times|\Lambda|} \to \mathbb{R}^{3N_{[p]}\times|\Lambda|}$. and (2) *anneal* operator $\mathbf{A}_\eta : \mathbb{R}^{3N_{[p]}\times|\Lambda|} \to \mathbb{R}^{3N_{[p]}\times|\Lambda|}$, where ϵ, η are random variables governing the randomness, and $|\Lambda|$ is the size of ensemble to be sampled. Inspired by tempering based enhanced sampling [24, 41, 59, 60], the problem can be described as follows:

$$\tilde{\mathbf{X}}_{[p]} = \mathbf{A}_\eta(\mathbf{H}_\epsilon(\mathbf{X}_{[p]})), \tag{5}$$

where $\mathbf{X}_{[p]} = \text{repeat}(\mathbf{x}_{[p]}; |\Lambda|) \in \mathbb{R}^{3N_{[p]}\times|\Lambda|}$ is an operation that repeats $\mathbf{x}_{[p]}$ for $|\Lambda|$ times to create a set of replicas before simulation. The above formulation indicates that the conformational ensemble can be obtained as a composition of *heat* and *anneal* operators. For example, in simulated tempering [41], $\mathbf{H}_\epsilon(\cdot)$ amounts to perform (independently for each replica) high-temperature MD simulations while the *anneal* operator $\mathbf{A}_\eta(\cdot)$ proceeds the simulations with decreased temperature, thus dispersing the initial state into different local minima (as shown in Fig. 2(a)). Similar formulation can be found in [6], which generalized the diffusion kernels to more abstract degradation-restoration operators for accommodation of arbitrary image transformation beyond Gaussian noise.

We further characterize these two operators. The ideal *heat* operator should perturb the replicas to let $\mathbf{H}_\epsilon(\mathbf{X}_{[p]})$ cover a sufficiently wide neighborhood of $\mathbf{x}_{[p]}$. Correspondingly, the *anneal* operator $\mathbf{A}_\eta$ should be able to equilibrate the "activated" ensemble $\mathbf{H}_\epsilon(\mathbf{X}_{[p]})$ to the local minima such that the final ensemble is reasonable. The ideal capacity of *anneal* operator lies in that it captures so abundant local minima that any properly dispersed replica can be guided to one of the low-energy conformations close to it. Note that $\mathbf{A}_\eta(\cdot)$ and $\mathbf{H}_\epsilon(\cdot)$ are not necessarily defined with a set (ensemble) as input. Sampling can be performed independently as in classical MD and simulated tempering [41]; while for methods such as REMD [59, 60], communications between different trajectories can benefit the simulation. For simplicity, we consider independent operators such that Eq. (5) can be replaced by its point-to-point versions, which are adopted in later paragraphs.

Based on this, we propose a novel score-based framework that realizes these two operators. We reason that, as a generative model: (1) The perturbing-denoising nature of SGMs transforms the input within the same space; (2) The continuous diffusion process allows us to partially perturb the input instead of projecting into a latent space; (3) SGMs have proved their effectiveness for modeling multimodal distribution [19] and are seldom haunted by the mode collapse issue.

3.1 Sampling via forward-backward dynamics

One key assumption is the diversity of conformational ensemble varies across different systems. To elaborate, starting from an initial conformation $\mathbf{x}(0)$, the sampler-accessible states are likely encompassed within an unknown $\delta-$neighborhood of $\mathbf{x}(0)$, where δ can vary between different systems by a large degree. An underestimated δ results in inadequate sampling, failing to reach distant modes within finite computational time, which is a challenge confronted by classical MD simulations or Markov chain Monte Carlo (MCMC); on the other hand, setting the allowable region to

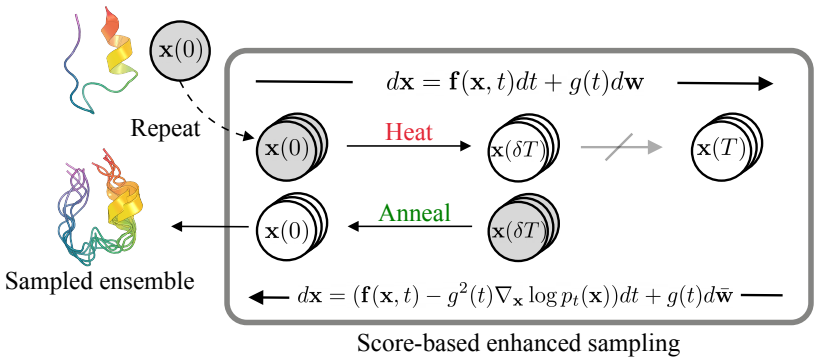


Figure 2: Illustration of SENS. Given an initial conformation (Trp-cage as example, PDB ID: 2JOF), a number of replicas are created (repeat) and fed to the *heat* process. The *heat* process respectively perturbs each replica until a specific time δT, where δ $(0 < \delta < 1)$ is a parameter controlling the range of perturbation. Afterwards, the *anneal* process yields the sampled ensemble.

nearly whole space can however lead to unrealistic conformational ensembles due to the prevalence of numerous local minima in high-dimension space. A random walker taking large steps might sample remote conformations while overlooking many closer optima, which deviates from real dynamics.

To better sample protein conformational ensembles, we propose a forward-backward dynamics that leverages the multi-level score functions learned by score matching. Firstly, consider the following realization of operators defined by integrals, given $\delta \in (0, 1)$ is the parameter controlling the perturbation scale:

$$\mathbf{H}_{\mathbf{w}}(\mathbf{x}(0)) \coloneqq \mathbf{x}(0) + \int_0^{\delta T} \left[\mathbf{f}(\mathbf{x}(t), t)dt + g(t)d\mathbf{w}\right], \tag{6}$$

where $\mathbf{x}(t)$ is the diffusion process as Eq. (1), with initial value $\mathbf{x}(0)$ being input geometry as in classical MD simulations. The $\mathbf{f}(\cdot, \cdot), g(\cdot)$ and $\mathbf{w} = \mathbf{w}(t)$ are defined over time domain $t \in [0, T]$ according to Eq. (1). And with positive dt:

$$\mathbf{A}_{\bar{\mathbf{w}}}(\mathbf{x}(\delta T)) \coloneqq \mathbf{x}(\delta T) + \int_{\delta T}^{2\delta T} \left\{\left[-\mathbf{f}(\bar{\mathbf{x}}(t), \tau(t)) + g^2(\tau(t))\nabla_{\bar{\mathbf{x}}(t)} \log p_t(\bar{\mathbf{x}}(t))\right] dt + g(\tau(t))d\bar{\mathbf{w}}\right\}, \tag{7}$$

where $\tau(t) \coloneqq 2\delta T - t$ for brief notation. $\bar{\mathbf{x}}(t) \coloneqq \mathbf{x}(2\delta T - t)$, $\bar{\mathbf{w}} \coloneqq \mathbf{w}(2\delta T - t)$ are identical processes as Eq. (1) and (2) with change of variable; the rest of symbols are denoted the same as above. Then, the sampled conformation $\tilde{\mathbf{x}}$ by the forward-backward dynamics take the following form by composing two operators:

$$\tilde{\mathbf{x}} = \mathbf{A}_{\bar{\mathbf{w}}}(\mathbf{H}_{\mathbf{w}}(\mathbf{x}(0))) \tag{8}$$

Intuitively, Eq. (8) is composed of both forward and backward SDEs (Eq. (1) and (2)), first enforcing proper perturbation and then mapping the perturbed examples into the annealed conformations. When δ is small, $\mathbf{H}_{\mathbf{w}}(\cdot)$ induces minor difference from the input structure; as δ increases, the initial conformation $\mathbf{x}(0)$ can become indistinguishable due to large-scale noise injection, thereby encouraging a diverse ensemble. The definitions above realize the *heat* and *anneal* operator respectively by pushing forward a new SDE from $t = 0$ to $2\delta T$.

Similar to the likelihood computation in probability flow ODE [58], we can also derive from Eq. (8) the (pseudo) free energy with the pretrained score functions. This can be used for energy reweighting [42] of the sampled ensemble to Boltzmann distribution, which is useful for computing thermodynamics quantities. More details can be found in the appendix.

3.2 Score matching objective

To approximate the score functions in Eq. (8), we can train a time-dependent score network $\mathbf{s}_\theta(\mathbf{x}, t)$ via the following objectives:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{t\in[0,t_m]} \left\{ \lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)} \left[\| \mathbf{s}_\theta(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}(0)) \|^2 \right] \right\}, \quad (9)$$

where $\lambda(t) > 0$ is a positive loss reweighting function, $\mathbf{x}(0) \sim p(\mathbf{x})$ and $\mathbf{x}(t) \sim p_{t|0}(\mathbf{x}(t)|\mathbf{x}(0))$ are defined by the corresponding perturbation kernel, and the time t is uniformly sampled over cropped time domain $[0, t_m]$ ($\delta T \leq t_m \leq T$ is the pre-specified upper bound of δT used during inference). Note that in Eq. (8), the diffusion process $\mathbf{x}(t)$ are not necessarily defined over the whole $[0, T]$, where $\mathbf{x}(T)$ indicates pure Gaussian noise. By setting $t_m = T$, it recovers the original objective of SGMs, which allows unconditionally generating conformation from pure noise. While small t_m enjoys reduced training cost, the optimal δ is unknown for unseen protein system during inference. For highly structural proteins, we may choose small t_m; for disordered proteins, we can set larger t_m to ensure the distant modes are accessible by the anneal operator. We set $t_m = T$ in our experiments for convenience and adopt $\lambda(t) \propto 1/\mathbb{E}\left[\|\nabla_{\mathbf{x}(t)} \log p_{t|0}(\mathbf{x}(t)|\mathbf{x}(0))\|\right]$ as suggested in [58].

3.3 Control sampling with temperature

Temperature plays an important role during sampling by effectively trade-off diversity and fidelity (i.e. exploration and exploitation). Given a probability density $p(\mathbf{x})$, the tempering distribution by an inverse temperature factor $\beta > 0$ is denoted as $p^{(\beta)}(\mathbf{x}) = (p(\mathbf{x}))^\beta / Z_\beta$, where Z_β is the normalizing constant $\int_{\mathbf{x}} (p(\mathbf{x}))^\beta d\mathbf{x}$, which only depends on β. The tempering strategy is commonly used for categorical distributions, such as language modeling, via rescaling the logits. Unfortunately, for SGMs, the temperature coefficient does not explicitly appear in the backward SDE.

Consider some probability distribution $p(\mathbf{x})$ to which we do not have access. We have done approximating its corresponding score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ using a pre-trained network $\mathbf{s}_\theta(\mathbf{x}, t)$. Our goal is to find the $\beta-$tempering version of the score $\nabla_{\mathbf{x}} \log p_t^{(\beta)}(\mathbf{x})$ such that the backward SDE can be mediated by β. To derive its expression, we firstly consider the case of Gaussian data distribution, i.e., $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$, and introduce the following lemma:

Lemma 3.1. *Let $\mathbf{x}(0) \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ be the data sample from some gaussian distribution, then the marginal distribution of perturbed data $\mathbf{x}(t)$ of Eq. (3) and (4) is subject to the following gaussian process:*

$$\mathbf{x}^{(VE)}(t) \sim \mathcal{N}(\mathbf{x}(t)|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0 + (\sigma^2(t) - \sigma^2(0))\mathbf{I}), \quad (10)$$

and

$$\mathbf{x}^{(VP)}(t) \sim \mathcal{N}(\mathbf{x}(t)|\alpha(t)\boldsymbol{\mu}_0, \alpha^2(t)\boldsymbol{\Sigma}_0 + (1 - \alpha(t))^2\mathbf{I}), \quad (11)$$

where $\alpha(t) = \exp(-\frac{1}{2}\int_0^t \beta(s)ds)$. Then we can obtain the following score rescaling theorem:

Theorem 3.2. *(Score rescaling) Suppose $\mathbf{x}(0) \in \mathbb{R}^d$ is Gaussian distributed with covariance matrix $\boldsymbol{\Sigma}$, then sampling via the backward process (Eq. (2)) with temperature control suffices to apply a linear transformation per time the score function at time t, or as the following form:*

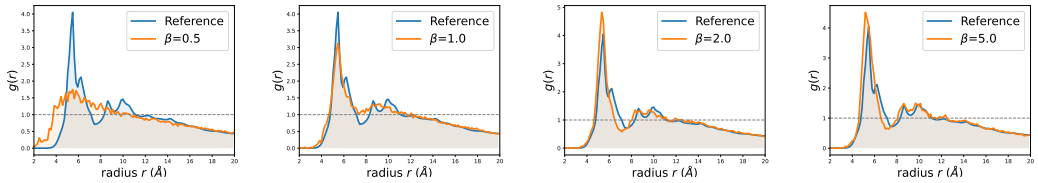
$$\nabla_{\mathbf{x}} \log p_t^{(\beta)}(\mathbf{x}) = \mathbf{R}(t; \boldsymbol{\Sigma}, \beta) \nabla_{\mathbf{x}} \log p_t(\mathbf{x}), \quad (12)$$

where $\mathbf{R}(t; \boldsymbol{\Sigma}, \beta) \in \mathbb{R}^{d\times d}$ is a time-dependent positive semi-definite matrix describing the tempering transformation, which only depends the covariance of data $\mathbf{x}(0)$ as well as an inverse temperature scalar $\beta > 0$. Especially, $\mathbf{R}(t; \boldsymbol{\Sigma}, \beta = 1) \equiv \mathbf{I}$, i.e., sampling without tempering effect. See appendix for detailed proof of Lemma 3.1 and Theorem 3.2, along with their specific forms for different SDEs.

The score rescaling in Eq. (12) only assumes Gaussian, yet this assumption can already be too strong. In the following section, we show that even when the Gaussian assumption fails to be made, we can still trade-off sample quality and diversity with the rescaling transformation via Eq. (12).

3.4 Model architectures

To parameterize the score network $s_\theta(\mathbf{x}, t)$ for conformation sampling, we adopt the modified EGNN [50] with SE(3)-equivariant property. Following [30, 31], we focus on residue-level protein structures, where the representation of conformation is denoted as $\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^{N\times 3}$, where $x_i \in \mathbb{R}^3$ is the coordinate of C_α atom of the i's residue. For message passing, we construct a k-nearest-neighbors (kNN) graph with maximum number of neighbors set to $k = 64$. To attend to



(a) $\beta = 0.5$ (b) $\beta = 1.0$ (c) $\beta = 2.0$ (d) $\beta = 5.0$

Figure 3: RDF of unconditionally sampled protein conformations, with Cα radius ranging from 2Å to 20Å. RDF of reference is colored as blue line while RDF of samples of specified β as orange line.

the chain structure of protein, sinusoidal positional embedding [66] is appended to the node feature along with the time embedding, which is used to condition the score network. As proved in [27, 72], the marginal distribution $p_t(\mathbf{x}(t))$ is invariant if the denoising transition distributions $p_{s|t}(\mathbf{x}(s)|\mathbf{x}(t))$ are equivariant. See appendix for the detailed model implementations.

4 Experiments

The training and validation examples are acquired from Protein Data Bank (PDB). Following [31], we set the training split as all structures released before April 30th, 2020 while the validation split as structures between May 1st, 2020 and November 30th, 2020. Single chain structures with sequence length between 20 and 256 were used. On top of these, we removed structures if they meet any of the criteria: (1) have missing atom or coordinates; or (2) contain residue discontinuity; or (3) resolution >5Å. We also dropped out structures appearing in our evaluation benchmarks to prevent data leakage. Finally, we obtained a training set with 193,145 structures, plus a validation set containing 12,110 structures. For experiments, we adopt the VESDE diffusion as backbone model.

As for baselines, AlphaFold2 (AF2) [32] is notably powerful for its highly accurate structure prediction for unseen proteins. However, several recent studies [12, 48, 64, 69] have investigated its potential for conformational sampling, with their results underscoring the limitations in conformational diversity. In accordance with these studies, We use AF2 with either no MSA (sequence only) or reduced MSA [12] as input, running all 5 models under multiple random seed to encourage diversity; idpGAN [30] is a recently reported conditional generative method based on generative adversarial networks (GAN). idpGAN is trained with plenty of simulation data of intrinsically disordered peptides (IDP) and can generalize to generate unseen protein dynamics; Like idpGAN, EigenFold [31] is a recently developed generative structure prediction model that employs harmonic diffusion that enables diversity while keeping good accuracy. Specifically, EigenFold was trained on general PDB database and use the pre-computed embeddings from OmegaFold [71], which are different from idpGAN.

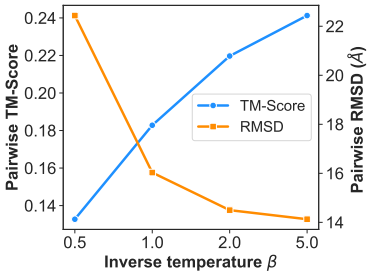


Figure 4: Pairwise TM-score (blue) and RMSD (orange) of unconditionally generated structures from SENS along with decreased temperature.

4.1 Effect of temperature on generation

In order to validate the proposed method, we firstly investigated the effect of inverse temperature β on the generative (backward) process. For evaluation, we generated 100 samples of length 128 from Gaussian noise under different β (ranging from 0.5 to 5.0) respectively. Note that when $\beta = 1.0$, sampling is not affected by temperature.

Firstly, we computed the radial distribution function (RDF) of Cα for the sampled conformations and reference set, which was curated from CATH v4.3 [53] with 40% sequence identity since it represents a good structural coverage of protein space. As shown in Fig. 3, as the temperature decreases, the

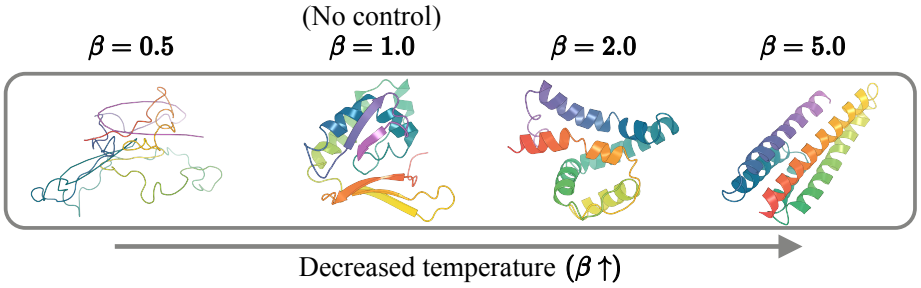


Figure 5: Visualization of protein examples unconditionally generated under different temperatures from noise. When temperature decreases (β increases), the structures exhibit an increased propensity for structural components.

RDF of generated samples becomes similar to RDF of reference. Moreover, as shown in Fig. 4, the pairwise TM-score increases/RMSD decreases while we decrease the temperature for sampling, which low temperature can uniformly lead to less diverse sampled ensembles. Lastly, we visualized the generated protein ribbon under different temperatures in Fig. 5. It demonstrates that decreasing temperature gives rise to samples with more structural components (more protein-like than random point clouds). Especially, when $\beta = 5.0$, the generated example exhibits the helical-bundle feature which is similar to high-quality *de novo* designed miniproteins [11].

4.2 Protein structural dynamics

To assess the performance of SENS on zero-shot conformational sampling, we set up the benchmark set consisting of 12 fast-folding proteins with up to 1ms long all-atom MD simulation trajectories as reference (named *reference* MD) [38]. Specifically, we generate 100 conformations for each protein from SENS and baseline models. The conformations from reference MD are uniformly sampled from MD trajectories in the same size with different timescales: 1μs, 10μs, full (the longest time in data). For example, 10μs means conformations are sampled from trajectories with the first 10μs. To better show the effect of temperature in MD simulation, we run two independent 0.1μs simulations (named *short* MD) with different temperatures for comparison as well.

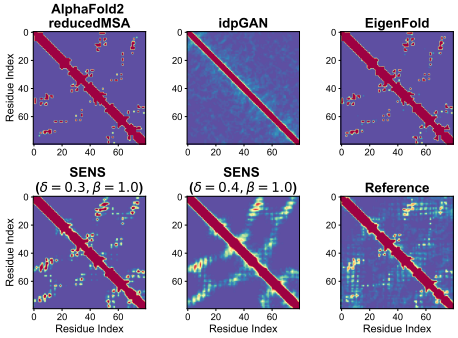


Figure 6: Contact probability distribution for protein *Lambda* (PDB ID: 1LMB) conformations from different methods.

Our evaluation metrics are categorized into: (a) *validity* assesses whether the sampled conformations obey basic physical constrain; (b) *fidelity* reflects the difference between generated conformations with reference MD simulations; (c) *diversity* evaluates the covering variety of generated ensemble. Each metric was averaged among different benchmark targets to give the finally evaluation as in Table 1. Metrics are briefly defined as below while the detailed definitions can be found in appendix.

Validity The validity is defined by clash ratio which is the number of clash-free samples divided by the number of samples. Clash is counted by examining whether a contact pair is within certain distance threshold.

Fidelity The fidelity compares the distribution divergence between sampled conformations and the full reference MD. We use the symmetric Jensen-Shannon (JS) divergence based on (i) pairwise distance distribution (JS-D) and (ii) radius of gyration distribution (JS-Rg) as in idpGAN [30].

Diversity The diversity of generated ensemble is defined the average pairwise dissimilarity scores based on root mean square deviation (RMSD, unit: nm) and TM-score [76]. For the latter, we apply the inverse TMscore ($1 - \text{TM}(i, j)$) to express "diversity" together with RMSD.

As shown in Table 1, SENS's performance validated its effectiveness in sampling diverse, yet physically plausible conformations when compared to other (generative) structure prediction models and traditional MD simulations. All structure generation models suffered from limited diversity and bad fidelity. Though idpGAN showed outstanding sampling diversity, its conformations tend to be disordered and deviate from true distribution (also as shown in Fig. 6). Notably for SENS, increasing temperature (w/ declined β) results in more diverse ensembles and better coverage of reference conformation space by trading off a bit validity.

Table 1: Benchmark results of different methods with MD references. The samples from reference MD trajectories are colored as brown while samples from other baselines are obtained by running their codes. Among these metrics, *validity*, Div-TM and Div-RMSD (*diversity*) are the high the better (↑); while JS-D and JS-Rg (*fidelity*) are the lower the better (↓). The cell color indicates the top three best scores in each column, and the best result of our methods are **bolded**.

	Validity(↑)	JS-D(↓)	JS-Rg (↓)	Div-TM(↑)	Div-RMSD(↑)
AlphaFold2 *reducedMSA* [65]	1.000	0.330	0.517	0.057	0.049
AlphaFold2 *noMSA*	0.995	0.321	0.431	0.194	0.224
EigenFold [31]	0.894	0.249	0.477	0.218	0.166
idpGAN [30]	1.000	0.251	0.530	0.839	1.346
SENS ($\delta = 0.3, \beta = 5.0$)	**1.000**	0.283	0.524	0.543	0.275
SENS ($\delta = 0.4, \beta = 5.0$)	**1.000**	0.252	0.422	0.647	0.401
SENS ($\delta = 0.3, \beta = 2.0$)	**1.000**	0.267	0.474	0.536	0.277
SENS ($\delta = 0.4, \beta = 2.0$)	0.998	0.217	0.401	0.693	0.440
SENS ($\delta = 0.3, \beta = 1.0$)	0.998	0.257	0.416	0.576	0.295
SENS ($\delta = 0.4, \beta = 1.0$)	0.993	**0.203**	**0.338**	**0.734**	**0.498**
Short MD 0.1μs	1.000	0.217	0.300	0.508	0.449
Short MD 0.1μs (high temp.)	1.000	0.204	0.285	0.536	0.495
Reference 1μs [38]	1.000	0.135	0.178	0.582	0.642
Reference 10μs [38]	1.000	0.100	0.156	0.650	0.736
Reference full [38]	1.000	0.000	0.000	0.659	0.756

4.3 Case study: BPTI dynamics

To better demonstrate the performance of our proposed method, we conducted a case study using the bovine pancreatic trypsin inhibitor (BPTI) protein. The dynamic properties of BPTI have been extensively studied using long molecular dynamics (MD) simulations [51]. Experiments and MD simulations revealed BPTI exhibits five distinct structural clusters [51]. Similar to our benchmark, full MD trajectory (1,013 μs) were used as reference and generative structure prediction methods were compared with ours. Principal Component Analysis (PCA) were employed for dimension reduction to illustrate the conformation distribution. As shown in the upper panel of Fig. 7, AlphaFold2 redusedMSA and EigenFold were only able to sample a constrained region, while idpGAN sampled an excessive amount of disordered conformations, which are unsuitable for structured proteins like BPTI. In contrast, our method was able to sample a reasonable and diverse range of conformations, outperform 1μs simulations, and approaching the full reference trajectory. In the lower panel of Fig. 7, we present the nearest sampled structure from SENS($\delta = 0.3, \beta = 1.0$, colored) to the five structural clusters. Similar structure can be found for all 5 five structure clusters with in RMSD 0.25 nm. These findings demonstrates a promising performance for SENS in generating reliable and diverse protein conformations.

5 Related works

Protein structure design A parallel research interest emerging recently focuses on the protein backbone structure design based on deep generative models. Early attempts include ProtDiff [63], which generates novel CA-only backbones; protein structure-sequence co-generation based on structural constraints [2]; and diffusion models tailored for antibody design [39]. FoldingDiff complements these by applying diffusion to the dihedral angles of backbones. Chroma [29] designs novel protein backbones with several conditional inputs including natural language and comprehensively evaluates the programmability. Meanwhile, RFdiffusion [68] pushed the diffusion-based protein design to the experimental side and validated the effectiveness of generative modeling for this task. More advanced

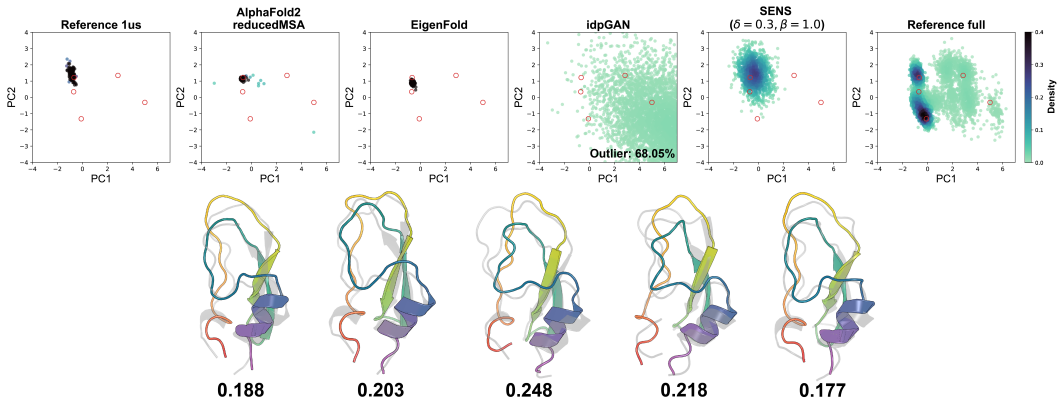


Figure 7: Visualization of sampled BPTI conformations and structure clusters. The upper panel illustrates a PCA-reduced map of the sampled conformations, pairwise distances were used as features. The color gradient represents the estimated density. Red circles indicate the locations of five structure clusters. 68.05% of the idpGAN samples are found outside the depicted area. The lower panel depicts the nearest sampled structure of SENS($\delta = 0.3, \beta = 1.0$, colored) to the five structure clusters (grey). RMSD (nm) is presented below.

methods including Genie [36] and FrameDiff [75] have been proposed very recently, leveraging the invariant point attention (IPA) modules proposed by AF2 [32] to enhance model capacity.

Learning for protein dynamics Due to the inefficiency of classical simulations for protein dynamics [9, 33], several works attempted to perform efficient sampling or learn neural force fields from simulation data. Boltzmann generators [42] were developed to generate equilibrium samples using normalizing flows [20, 45] trained on simulation data or ground-truth energy. CGNets [67] proposed learning coarse-grained (CG) force fields trained by a force-matching objective in a supervised learning manner. Flow-matching [34] improved this by complementing density estimation and sampling right before force-matching, thus not relying on ground-truth forces in simulation data. The most recent work [4] proposed to train diffusion model on samples from equilibrium distribution of a specific protein, and leveraged either the first-layer score function as learned force field or the entire backward diffusion to perform conformational sampling. However, these works can suffer from the transferability problem [67] and cannot be generalized to unseen proteins. While these methods were designed *ad hoc* and can only be applied to the proteins they are trained on, our method is distinguished from them by performing zero-shot conformation sampling, ready for modeling unseen protein dynamics.

6 Conclusion and limitations

In this paper, we presented SENS, a score-based framework for protein conformational sampling inspired by tempering-based enhanced sampling methods in molecular dynamics (MD) simulations. SENS firstly disperses the initial conformation within its neighborhood, followed by a score-based annealing process. We trained SENS on general protein structure data from the Protein Data Bank (PDB) using score matching objectives. Experimental results on several MD benchmarking systems demonstrate that SENS can effectively sample a diverse ensemble from the initial conformation and achieve comparable performance with long simulation results.

Limitations of SENS and potential future directions encompass several aspects: **(1)** The isotropic perturbation kernels could be improved by incorporating drift and diffusion terms specifically designed for protein structures [29, 31]. This modification would allow the perturbations based on non-Euclidean coordinates, drawing parallels to CV-based enhanced sampling techniques. **(2)** SENS has mainly been tested on VE/VPSDE diffusion [58] as proof of concept. The backward process, however, can be computationally intensive due to plenty of network evaluations. To accelerate SENS for efficient sampling, one might consider the use of distillation [49] or more advanced frameworks such as consistency models [56]. **(3)** An interesting application of SENS is to plug it in atom-level MD simulations by providing out-of-barrier conformation set as MD starting points. To achieve this,

one may either extend SENS for modeling all-atom structures, or develop back-mapping algorithms like in [73]. **(4)** The proposed method, in its current form, relies solely on structure data for training. As amino acid sequences was also reported to be used for inferring protein structure [37], it's worth exploring the feasibility to use sequence as conditional signal when modeling dynamics. For example, one can provide gradient guidance during the backward SDE [58] using an external classifier, which could be trained on sequence-structure pairing data using a CLIP-like architecture [43].

References

[1] Cameron Abrams and Giovanni Bussi. Enhanced sampling in molecular dynamics using metadynamics, replica-exchange, and temperature-acceleration. *Entropy*, 16(1):163–199, 2013.

[2] Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022.

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

[4] Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zuegner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *arXiv preprint arXiv:2302.00600*, 2023.

[5] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[6] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*, 2022.

[7] David Belanger and Andrew McCallum. Structured prediction energy networks. In *International Conference on Machine Learning*, pages 983–992. PMLR, 2016.

[8] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[9] Rafael C Bernardi, Marcelo CR Melo, and Klaus Schulten. Enhanced sampling techniques in molecular dynamics simulations of biological systems. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1850(5):872–877, 2015.

[10] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[11] Longxing Cao, Inna Goreshnik, Brian Coventry, James Brett Case, Lauren Miller, Lisa Kozodoy, Rita E Chen, Lauren Carter, Alexandra C Walls, Young-Jun Park, et al. De novo design of picomolar sars-cov-2 miniprotein inhibitors. *Science*, 370(6515):426–431, 2020.

[12] Devlina Chakravarty and Lauren L Porter. Alphafold2 fails to predict protein fold switching. *Protein Science*, 31(6):e4353, 2022.

[13] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[14] Giovanni Ciccotti and Jean-Paul Ryckaert. Molecular dynamics simulation of rigid molecules. *Computer Physics Reports*, 4(6):346–392, 1986.

[15] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. *arXiv preprint arXiv:2210.01776*, 2022.

[16] Tom Darden, Darrin York, and Lee Pedersen. Particle mesh ewald: An n log (n) method for ewald sums in large systems. *The Journal of chemical physics*, 98(12):10089–10092, 1993.

[17] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[18] Diego Del Alamo, Davide Sala, Hassane S Mchaourab, and Jens Meiler. Sampling alternative conformational states of transporters and receptors with alphafold2. *Elife*, 11:e75751, 2022.

[19] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

[20] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[21] Peter Eastman, Jason Swails, John D Chodera, Robert T McGibbon, Yutong Zhao, Kyle A Beauchamp, Lee-Ping Wang, Andrew C Simmonett, Matthew P Harrigan, Chaya D Stern, et al. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology*, 13(7):e1005659, 2017.

[22] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.

[23] Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

[24] Ulrich HE Hansmann. Parallel tempering algorithm for conformational studies of biological molecules. *Chemical Physics Letters*, 281(1-3):140–150, 1997.

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[27] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022.

[28] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

[29] John Ingraham, Max Baranov, Zak Costello, Vincent Frappier, Ahmed Ismail, Shan Tie, Wujie Wang, Vincent Xue, Fritz Obermeyer, Andrew Beam, et al. Illuminating protein space with a programmable generative model. *bioRxiv*, pages 2022–12, 2022.

[30] Giacomo Janson, Gilberto Valdes-Garcia, Lim Heo, and Michael Feig. Direct generation of protein conformational ensembles via machine learning. *Nature Communications*, 14(1):774, 2023.

[31] Bowen Jing, Ezra Erives, Peter Pao-Huang, Gabriele Corso, Bonnie Berger, and Tommi Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models. *arXiv preprint arXiv:2304.02198*, 2023.

[32] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[33] Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature structural biology*, 9(9):646–652, 2002.

[34] Jonas Köhler, Yaoyi Chen, Andreas Krämer, Cecilia Clementi, and Frank Noé. Flow-matching: Efficient coarse-graining of molecular dynamics without forces. *Journal of Chemical Theory and Computation*, 19(3):942–952, 2023.

[35] Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the national academy of sciences*, 99(20):12562–12566, 2002.

[36] Yeqing Lin and Mohammed AlQuraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. *arXiv preprint arXiv:2301.12485*, 2023.

[37] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[38] Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

[39] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models. *bioRxiv*, pages 2022–07, 2022.

[40] James A Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E Hauser, and Carlos Simmerling. ff14sb: improving the accuracy of protein side chain and backbone parameters from ff99sb. *Journal of chemical theory and computation*, 11(8):3696–3713, 2015.

[41] Enzo Marinari and Giorgio Parisi. Simulated tempering: a new monte carlo scheme. *Europhysics letters*, 19(6):451, 1992.

[42] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[44] Srinivas Ramachandran, Pradeep Kota, Feng Ding, and Nikolay V Dokholyan. Automated minimization of steric clashes in protein structures. *Proteins: Structure, Function, and Bioinformatics*, 79(1):261–270, 2011.

[45] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[46] Sam Roweis and Zoubin Ghahramani. A unifying review of linear gaussian models. *Neural computation*, 11(2):305–345, 1999.

[47] Jean-Paul Ryckaert, Giovanni Ciccotti, and Herman JC Berendsen. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of computational physics*, 23(3):327–341, 1977.

[48] Tadeo Saldaño, Nahuel Escobedo, Julia Marchetti, Diego Javier Zea, Juan Mac Donagh, Ana Julia Velez Rueda, Eduardo Gonik, Agustina García Melani, Julieta Novomisky Nechcoff, Martín N Salas, et al. Impact of protein conformational diversity on alphafold predictions. *Bioinformatics*, 38(10):2742–2748, 2022.

[49] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

[50] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[51] David E Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, Michael P Eastwood, Joseph A Bank, John M Jumper, John K Salmon, Yibing Shan, et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

[52] Chence Shi, Chuanrui Wang, Jiarui Lu, Bozitao Zhong, and Jian Tang. Protein sequence and structure co-design with equivariant translation. *arXiv preprint arXiv:2210.08761*, 2022.

[53] Ian Sillitoe, Nicola Bordin, Natalie Dawson, Vaishali P Waman, Paul Ashford, Harry M Scholes, Camilla SM Pang, Laurel Woodridge, Clemens Rauer, Neeladri Sen, et al. Cath: increased structural coverage of functional space. *Nucleic acids research*, 49(D1):D266–D273, 2021.

[54] John Skilling. The eigenvalues of mega-dimensional matrices. *Maximum Entropy and Bayesian Methods: Cambridge, England, 1988*, pages 455–466, 1989.

[55] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[56] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[57] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[58] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[59] Yuji Sugita and Yuko Okamoto. Replica-exchange molecular dynamics method for protein folding. *Chemical physics letters*, 314(1-2):141–151, 1999.

[60] Robert H Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. *Physical review letters*, 57(21):2607, 1986.

[61] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[62] Glenn M Torrie and John P Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, 1977.

[63] Brian L Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

[64] Bodhi P Vani, Akashnathan Aranganathan, Dedi Wang, and Pratyush Tiwary. From sequence to boltzmann weighted ensemble of structures with alphafold2-rave. *bioRxiv*, pages 2022–05, 2022.

[65] Bodhi P Vani, Akashnathan Aranganathan, Dedi Wang, and Pratyush Tiwary. Alphafold2-rave: From sequence to boltzmann ranking. *Journal of Chemical Theory and Computation*, 2023.

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[67] Jiang Wang, Simon Olsson, Christoph Wehmeyer, Adrià Pérez, Nicholas E Charron, Gianni De Fabritiis, Frank Noé, and Cecilia Clementi. Machine learning of coarse-grained molecular dynamics force fields. *ACS central science*, 5(5):755–767, 2019.

[68] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *bioRxiv*, pages 2022–12, 2022.

[69] Hannah K Wayment-Steele, Sergey Ovchinnikov, Lucy Colwell, and Dorothee Kern. Prediction of multiple conformational states by combining sequence clustering with alphafold2. *bioRxiv*, pages 2022–10, 2022.

[70] Kevin E Wu, Kevin K Yang, Rianne van den Berg, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *arXiv preprint arXiv:2209.15611*, 2022.

[71] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pages 2022–07, 2022.

[72] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

[73] Soojung Yang and Rafael Gómez-Bombarelli. Chemically transferable generative backmapping of coarse-grained proteins. *arXiv preprint arXiv:2303.01569*, 2023.

[74] Yi Isaac Yang, Qiang Shao, Jun Zhang, Lijiang Yang, and Yi Qin Gao. Enhanced sampling in molecular dynamics. *The Journal of chemical physics*, 151(7):070902, 2019.

[75] Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

[76] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

[77] Bozitao Zhong, Xiaoming Su, Minhua Wen, Sicheng Zuo, Liang Hong, and James Lin. Parafold: paralleling alphafold for large-scale predictions. In *International Conference on High Performance Computing in Asia-Pacific Region Workshops*, pages 1–9, 2022.

A Further details of SENS

A.1 Equivariant score network

In this section, we elaborate the equivariant model architecture used to parameterize the score function for SENS. Basically, we adopt the E(n)-equivariant graph neural networks (EGNN) [50] given its simple and computationally efficient form, with an additional SE(3)-equivariant convolutional layer to achieve the SE(3)-equivariant property during the backward process. The SE(3)-equivariant property of a function $(\boldsymbol{x}', \boldsymbol{h}') = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{h})$ with respect to group transformations operating on three-dimensional (3D) coordinates $\boldsymbol{x} \in \mathbb{R}^{N\times 3}$ and hidden features $\boldsymbol{h} \in \mathbb{R}^{N\times d}$, is defined below:

$$(\boldsymbol{x}'\boldsymbol{R}^\top + \boldsymbol{t}, \boldsymbol{h}') = \boldsymbol{f}(\boldsymbol{x}\boldsymbol{R}^\top + \boldsymbol{t}, \boldsymbol{h}), \tag{13}$$

where rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ (orthogonal with determinant 1) and translation vector $\boldsymbol{t} \in \mathbb{R}^3$ define the special Euclidean group (SE(3) group) transformations, where any rotation (yet not reflection) and translation in 3D space are included. The hidden features are invariant to the group actions and thus not affected by the transformation. The advantage of leveraging such equivariance is that the conformations generated via a group orbit, i.e., by applying all group actions on some specific input $\boldsymbol{x}$, can be equivalent (trivial) in 3D space because they belong to the same molecular species. Neural networks equipped with equivariant property can focus more on performing non-trivial mapping for the conformations.

The SE(3) convolutional layer is applied before the EGNN network, and adopts the definition in SE(3)-transformer [22] and tensor field networks (TFN) [61]. To simplify, we only consider up to type-1 features (scalar and vector field) from each node's neighborhood for the convolution, which means, the output type-l ($l = 0, 1$) feature for node i is:

$$\boldsymbol{f}^l_{\text{out},i} = w^{ll}\boldsymbol{f}^l_{\text{in},i} + \sum_{k=0}^{1}\sum_{j\in\mathcal{N}_i} \boldsymbol{W}^{lk}(\boldsymbol{x}_j - \boldsymbol{x}_i)\boldsymbol{f}^k_{\text{in},j}, \tag{14}$$

where $\boldsymbol{f}^l_{\text{in},i}$ is the type-l feature for node i, $\boldsymbol{W}^{lk} : \mathbb{R}^3 \to \mathbb{R}^{(2l+1)\times(2k+1)}$ is the learnable kernel mapping from type-k features to type-l features, and $w^{ll}\boldsymbol{f}^l_{\text{in},i}$ is the self-interaction term with learnable parameters $w^{ll} \in \mathbb{R}$. As shown in [61], such kernel is a combination of equivariant basis kernels $\{\boldsymbol{W}^{lk}_J\}^{k+l}_{J=|k-l|}$, or formally

$$\boldsymbol{W}^{lk}(\boldsymbol{x}) = \sum_{J=|k-l|}^{k+l} \phi^{lk}_J(\|\boldsymbol{x}\|)\boldsymbol{W}^{lk}_J(\boldsymbol{x}), \text{ where } \boldsymbol{W}^{lk}_J(\boldsymbol{x}) = \sum_{m=-J}^{J} Y_{Jm}(\boldsymbol{x}/\|\boldsymbol{x}\|)\boldsymbol{Q}^{lk}_{Jm}, \tag{15}$$

where $\boldsymbol{Q}^{lk}_{Jm}$ are the Clebsch-Gordan matrices of shape $(2l+1)\times(2k+1)$, Y_{Jm} indicates the m's dimension of spherical harmonic $Y_J : \mathbb{R}^3 \to \mathbb{R}^{2J+1}$, and $\phi_J(\cdot)$ are learnable radial functions with input radial $\|\boldsymbol{x}\|$.

EGNN is a type of message passing graph neural networks with the equivariant graph convolutional layers (EGCL) defined on type-1 and type-0 features $(\boldsymbol{x}^{l+1}, \boldsymbol{h}^{l+1}) = \phi^l_{\text{EGNN}}(\boldsymbol{x}^l, \boldsymbol{h}^l)$, or formally as follows (l is the index of layer):

$$\begin{aligned}
&\boldsymbol{m}_{ij} = \phi_m(\boldsymbol{h}^l_i, \boldsymbol{h}^l_j, d^2_{ij}, e_{ij}), \quad \boldsymbol{h}^{l+1}_i = \phi_h(\boldsymbol{h}^l_i, \sum_{j\in\mathcal{N}_i} a_{ij}\boldsymbol{m}_{ij}),\\
&\boldsymbol{x}^{l+1}_i = \boldsymbol{x}^l_i + \sum_{j\in\mathcal{N}_i}(\frac{\boldsymbol{x}^l_i - \boldsymbol{x}^l_j}{d_{ij}+1})\phi_x(\boldsymbol{h}^l_i, \boldsymbol{h}^l_j, d^2_{ij}, e_{ij}),
\end{aligned} \tag{16}$$

where $d_{ij} \coloneqq \|\boldsymbol{x}^l_i - \boldsymbol{x}^l_j\|_2$ denotes the Euclidean distance between two nodes w.r.t. their type-1 features, e_{ij} is the edge attributes from input, $a_{ij} = \phi_a(\boldsymbol{m}_{ij})$ indicates the attention mechanism that aggregates the message $\boldsymbol{m}_{ij}$, $\mathcal{N}_i$ is the neighborhood of node i, and the learnable mappings $(\phi_m, \phi_h, \phi_x, \phi_a)$ are parameterized by multi-layer perceptrons (MLP). The whole EGNN architecture is composed of L such layers to perform the non-linear and equivariant transformation.

Denoting the SE(3) convolutional layer as $\varphi_\theta(\cdot)$ and the EGNN network as $\text{EGNN}_\theta(\cdot)$, the equivariant score network used in our work has the composition form as $\mathbf{s}_\theta(\boldsymbol{x}, t) = \text{Out}^{(1)}\left[(\text{EGNN}_\theta \circ \varphi_\theta)(\boldsymbol{x}, \boldsymbol{h}^0(t))\right]$, where $\circ$ means the composition of two functions, $\text{Out}^{(1)}(\cdot)$ indicates extracting only the type-1 feature (vector field), $\boldsymbol{x}$ is the atom coordinates and $\boldsymbol{h}^0(t)$ is the time-dependent initial embedding (type-0 features). Since the special Euclidean group SE(3) is just a subgroup of the Euclidean group E(3) by excluding reflections, it is trivial to prove that the score network $\mathbf{s}_\theta(\mathbf{x}, t)$, composed of SE(3) and E(3)-equivariant functions, is SE(3)-equivariant w.r.t. $\mathbf{x}$ for any fixed t.

A.2 Implementation details

For equivariant score network, the nodes $\boldsymbol{x}^0$ (type-1) are initialized by the coordinates of the Cα atoms of the input conformation, while the node features $\boldsymbol{h}^0$ (type-0) are initialized by the sinusoidal time embedding and positional embedding [66]. To be specific, both embeddings having a dimension of 128 are concatenated and then passed through a affine transformation and a followed layer normalization. The size of node embedding is 128 and we do not consider edge feature to construct the denoising network. The whole EGNN network consists of 4 convolutional layers, 128 features per layer and SiLU activation functions. The score network is trained for a total 10,000,000 updates with batch size 2, learning rate 1e-5 using Adam optimizer and a warmup linear schedule, where learning rate are linearly increasing to the maximum value for the first 100,000 steps and linearly decreasing to zero for the rest of steps. The training was deployed on NVIDIA Tesla V100-SXM2 32GB and the training process approximately lasted 24 GPU days for a single run. To determine the best model checkpoint for sampling experiments, we employed early stopping strategy to avoid overfitting based on the validation loss per epoch. For diffusion, we used $T = 1000$ as the total number of time steps. During sampling, the reverse diffusion samplers [58] were adopted to discretize the reverse-time domain and calculate the integral.

A.3 Data processing

For each atom coordinate in the PDB dataset, we perform the whitening transformation by deducting a mean vector and element-wise rescaling by a factor. Specifically, for any protein conformation with N atoms $\mathbf{x} = (x_1, x_2, \ldots, x_N)$, we apply $x_i \leftarrow (1/\hat{\sigma}_{\text{train}}) \odot (x_i - 1/N \sum_i x_i), \forall i$, where $x_i \in \mathbb{R}^3$ is the Euclidean coordinate (xyz) of the ith atom, $\hat{\sigma}_{\text{train}}$ is the standard deviation vector calculated independently for each component in 3D space (we denote as $x[0], x[1], x[2]$) among all the atom coordinates in the training set, while $\odot$ is Hadamard (element-wise) product operation. In practice, we set $\hat{\sigma}_{\text{train}}$ = [10.08, 10.05, 10.74] and such statistic is computed from the whole training set.

A.4 Proof of Lemma 3.1

Below we give a detailed proof of 3.1. Consider we have a data sample $\mathbf{x}_0 \in \mathbb{R}^d$ is normally distributed, then its marginal distribution is:

$$p_0(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_0 - \boldsymbol{\mu}_0)^T {\boldsymbol{\Sigma}_0}^{-1}(\mathbf{x}_0 - \boldsymbol{\mu}_0)\right). \tag{17}$$

Because we need to derive both marginal distribution for both VESDE and VPSDE, we start with a more general case. Consider another conditional Gaussian distribution p_t for $\mathbf{x}_t$ given $\mathbf{x}_0$ in the form:

$$p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\mathbf{A}\mathbf{x}_0 + \mathbf{b}, \boldsymbol{\Sigma}_t), \tag{18}$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}, \mathbf{b} \in \mathbb{R}^d$ are parameters defining the linear function of its condition variable $\mathbf{x}_0$, and $\boldsymbol{\Sigma}_t$ is the covariance matrix of $\mathbf{x}_t$. This is exactly an example of a linear Gaussian model [46].

The derivation of $p_t(\mathbf{x}_t)$ involves computing the integral $\int_{\mathbf{x}} p_0(\mathbf{x}) p_{t|0}(\mathbf{x}_t|\mathbf{x}) d\mathbf{x}$, which in general can be as intractable as the evidence (normalizing constant of posterior) that appears in Bayesian inference. However, this can be solved in closed-form for Gaussian distributions. As proved in the Section 2.3.3 of [10], the marginal distribution $p_t(\mathbf{x}_t)$ is also normally distributed and take the following form:

$$p_t(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t|\mathbf{A}\boldsymbol{\mu}_0 + \mathbf{b}, \boldsymbol{\Sigma}_t + \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top), \tag{19}$$

Therefore, for VESDE, the perturbation kernel in Eq. (3) give the conditional distribution [58] as:

$$p_{t|0}^{(\text{VE})}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\mathbf{x}_0, (\sigma^2(t) - \sigma^2(0))\mathbf{I}), \tag{20}$$

where $\sigma(t)$ is noise scale function of VESDE and $t \in [0, T]$ is the continuous time variable. Then, plug in the conditional distribution back to Eq. (19), it finally yields

$$p_t^{(\text{VE})}(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0 + (\sigma^2(t) - \sigma^2(0))\mathbf{I}). \tag{21}$$

Also, for VPSDE, the conditional distribution based on the perturbation kernel is:

$$p_{t|0}^{(\text{VP})}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t|\alpha(t)\mathbf{x}_0, (1 - \alpha^2(t))\mathbf{I}), \tag{22}$$

where $\alpha(t) = \exp(-\frac{1}{2}\int_0^t \beta(s)ds)$ and $\beta(t)$ is the noise function in VPSDE. Similar to above, the marginal distribution of $\mathbf{x}_t$ is:

$$p_t^{(\text{VP})}(\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t|\alpha(t)\boldsymbol{\mu}_0, \alpha^2(t)\boldsymbol{\Sigma}_0 + (1 - \alpha^2(t))\mathbf{I}), \tag{23}$$

A.5 Proof of Theorem 3.2

Since we have obtained the marginal distributions $p_t(\mathbf{x}_t)$ as above, we can prove the Theorem 3.2 by giving the specific form of tempering transformation. Consider the general form of marginal distribution as Eq. (19), based on the density function of Gaussian distribution, the corresponding score function can be written as:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = -(\boldsymbol{\Sigma}_t + \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)^{-1} (\mathbf{x}_t - (\mathbf{A}\boldsymbol{\mu}_0 + \mathbf{b})), \tag{24}$$

Now suppose we want to "temper" the original data distribution $p_0(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ to a inverse temperature factor $\beta > 0$, i.e.,

$$p_0^{(\beta)}(\mathbf{x}_0) = \frac{(p_0(\mathbf{x}_0))^\beta}{Z_\beta} = \mathcal{N}(\mathbf{x}_0|\boldsymbol{\mu}_0, \beta^{-1}\boldsymbol{\Sigma}_0), \tag{25}$$

where Z_β is the normalizing constant. This indicates the inverse temperature β actually rescales the data covariance by a factor $1/\beta$. Plug-in back to Eq. (24), we have:

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log^{(\beta)} p_t(\mathbf{x}_t) &= -(\boldsymbol{\Sigma}_t + \beta^{-1}\mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)^{-1} (\mathbf{x}_t - (\mathbf{A}\boldsymbol{\mu}_0 + \mathbf{b})) \\
&= -(\boldsymbol{\Sigma}_t + \beta^{-1}\mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)^{-1} \left[(\boldsymbol{\Sigma}_t + \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)(\boldsymbol{\Sigma}_t + \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)^{-1}\right] (\mathbf{x}_t - (\mathbf{A}\boldsymbol{\mu}_0 + \mathbf{b})) \\
&= (\boldsymbol{\Sigma}_t + \beta^{-1}\mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)^{-1}(\boldsymbol{\Sigma}_t + \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top) \left[-(\boldsymbol{\Sigma}_t + \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)^{-1} (\mathbf{x}_t - (\mathbf{A}\boldsymbol{\mu}_0 + \mathbf{b}))\right] \\
&= (\boldsymbol{\Sigma}_t + \beta^{-1}\mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)^{-1}(\boldsymbol{\Sigma}_t + \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t).
\end{aligned} \tag{26}$$

Finally, let $\mathbf{R}(t; \boldsymbol{\Sigma}, \beta) \coloneqq (\boldsymbol{\Sigma}_t + \beta^{-1}\mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)^{-1}(\boldsymbol{\Sigma}_t + \mathbf{A}\boldsymbol{\Sigma}_0\mathbf{A}^\top)$, we have the result of Theorem 3.2:

$$\nabla_{\mathbf{x}} \log p_t^{(\beta)}(\mathbf{x}) = \mathbf{R}(t; \boldsymbol{\Sigma}, \beta)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}). \tag{27}$$

Note that for different SDEs, the tempering transformation $\mathbf{R}(t; \boldsymbol{\Sigma}, \beta)$ takes different forms since it is also determined by $\boldsymbol{\Sigma}_t$ and $\mathbf{A}$, which are further determined by the noise schedule functions. To be specific, for VESDE, $\mathbf{A} = \mathbf{I}, \boldsymbol{\Sigma}_t = (\sigma^2(t) - \sigma^2(0))\mathbf{I}$; for VPSDE, $\mathbf{A} = \alpha(t)\mathbf{I}, \boldsymbol{\Sigma}_t = (1 - \alpha^2(t))\mathbf{I}$. Therefore, by plugging in, the tempering transformation will be:

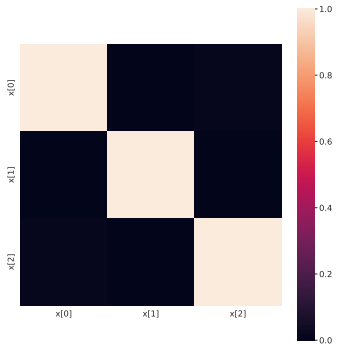


Figure S1: Covariance matrix $\mathbf{\Sigma}_{\text{a}}$ calculated among all atom coordinates in the training set after the whitening process as in Appendix A.3.

$$\mathbf{R}^{(\text{VE})}(t; \mathbf{\Sigma}, \beta) \coloneqq ((\sigma^2(t) - \sigma^2(0))\mathbf{I} + \frac{1}{\beta}\mathbf{\Sigma}_0)^{-1}((\sigma^2(t) - \sigma^2(0))\mathbf{I} + \mathbf{\Sigma}_0), \tag{28}$$

for VESDE and

$$\mathbf{R}^{(\text{VP})}(t; \mathbf{\Sigma}, \beta) \coloneqq ((1 - \alpha^2(t))\mathbf{I} + \frac{\alpha^2(t)}{\beta}\mathbf{\Sigma}_0)^{-1}((1 - \alpha^2(t))\mathbf{I} + \alpha^2(t)\mathbf{\Sigma}_0), \tag{29}$$

for VPSDE.

A.6 Covariance determination

According to Eq. (12), we need to compute the covariance matrix $\mathbf{\Sigma}$ in order to properly rescale the score function during sampling. Because covariance $\mathbf{\Sigma}$ for the test protein in benchmark set cannot assume to be accessible during inference, we estimate it using the covariance of training set $\mathbf{\Sigma} \approx \hat{\mathbf{\Sigma}}_{\text{train}}$ since they can be viewed as identically distributed. To accommodate structures with different length, we adopt a simple strategy by applying the unified covariance matrix for each atom and concatenate together, such that $\mathbf{\Sigma}$ has the following block diagonal form:

$$\begin{bmatrix} \mathbf{\Sigma}_{\text{a}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{\text{a}} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{\Sigma}_{\text{a}} \end{bmatrix}$$

where $\mathbf{\Sigma}_{\text{a}} \in \mathbb{R}^{3\times3}$ is the covariance matrix calculated among all the atoms in the training set, and the number of blocks in a row or column depends on the number of atoms in the test protein when applying the rescaling transformation. We then evaluate $\mathbf{\Sigma}_{\text{train}}$ for the training net after whitening. As shown in Fig. S1, the $\mathbf{\Sigma}_{\text{a}}$ is nearly a identity matrix in $\mathbb{R}^3$ and thus the $\mathbf{\Sigma}$ can be as simple as a identity matrix in $\mathbb{R}^{N\times3}$ when applying to a protein with N atoms.

A.7 Pseudo free energy computation

We firstly write the probability flow ODE that belongs to the backward diffusion in Eq. (2) by replacing the score function with time-conditioned score network $\mathbf{s}_\theta(\mathbf{x}, t)$ as follows (negative dt, same below):

$$d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}}\mathbf{s}_\theta(\mathbf{x}, t)]dt. \tag{30}$$

Let $\tilde{\mathbf{f}}_\theta(\mathbf{x}, t) \coloneqq \mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g^2(t) \nabla_{\mathbf{x}} \mathbf{s}_\theta(\mathbf{x}, t)$, and consider a sampled conformation $\tilde{\mathbf{x}}$ from input conformation $\mathbf{x}(0)$, we can compute the pseudo free energy (PFE) by employing the instantaneous change of variables formula [13]:

$$\text{PFE}(\tilde{\mathbf{x}}_0) = -\log p_{\delta T|0}(\mathbf{x}(\delta T)|\mathbf{x}(0)) - \int_0^{\delta T} \nabla \cdot \tilde{\mathbf{f}}_\theta(\mathbf{x}(t), t) dt, \tag{31}$$

where the perturbed sample $\mathbf{x}(\delta T)$ and $\log p_{\delta T|0}(\mathbf{x}(\delta T)|\mathbf{x}(0))$ are explicitly defined by the perturbation kernel in Eq. (6) and can be evaluated in closed-form. To reduce the expensive computation of the second term, we may used the Skilling-Hutchinson trace estimator [28, 54] as suggested in [23, 58]:

$$\nabla \cdot \tilde{\mathbf{f}}_\theta(\mathbf{x}, t) = \mathbb{E}_{p(\epsilon)}[\epsilon^\top \nabla \tilde{\mathbf{f}}_\theta(\mathbf{x}, t) \epsilon], \tag{32}$$

where $\nabla \tilde{\mathbf{f}}_\theta(\mathbf{x}, t)$ denotes the Jacobian of $\tilde{\mathbf{f}}_\theta(\cdot, t)$, and the $\epsilon \in \mathbb{R}^d$ is a white noise, i.e., $\mathbb{E}[\epsilon] = \mathbf{0}$ and $\text{Cov}[\epsilon] = \mathbf{I}$. In practice, we can sample $\epsilon \sim p(\epsilon)$ and compute the unbiased estimation using above equation up to arbitrarily small error [58] given sufficient computation.

A.8 Pseudocode of forward-backward dynamics

The pseudocode for the forward-backward dynamics is shown in Algorithm 1 and 2 for better illustration.

Algorithm 1 Forward-backward dynamics (VESDE)

Require: input conformation $\mathbf{x}$, constant perturbation scale δ; time upper bound for diffusion process T; score network $\mathbf{s}_\theta$; noise scale function σ; data covariance $\mathbf{\Sigma}$; inverse temperature τ.
$\mathbf{x}_0 \leftarrow \mathbf{x}$ *// initialize state*
$\{t_1, \dots, t_M\} \leftarrow \text{Discretize}([0, \delta T])$ *// discretize time domain*
for $i = 1$ to M **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}_{i-1} + \sqrt{\sigma^2(t_i) - \sigma^2(t_{i-1})}\mathbf{z}$
for $i = M - 1$ to 0 **do**
  $\mathbf{R}_{i+1} \leftarrow \mathbf{R}(t_{i+1}; \mathbf{\Sigma}, \tau)$ // in Eq. (28)
  $\mathbf{s}'_{i+1} \leftarrow \mathbf{R}_{i+1} \mathbf{s}_\theta(\mathbf{x}_{i+1}, \sigma(t_{i+1}))$ *// score rescaling*
  $\mathbf{x}'_i \leftarrow \mathbf{x}_{i+1} + \left[\sigma^2(t_{i+1}) - \sigma^2(t_i)\right] \mathbf{s}'_{i+1}$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{\sigma^2(t_{i+1}) - \sigma^2(t_i)}\mathbf{z}$
return $\mathbf{x}_0$

B Details of evaluation metrics

In this section, we elaborate the definition of the evaluation metrics introduced in the experiments.

Validity The Validity is defined as the ratio of clash conformations. It is calculated as the number of conformations that contain steric clashes divided by the number of all evaluating examples. Steric clash is determined by whether two contacting atoms is too close to each other. For an example conformational ensemble $\{\mathbf{x}_i\}_{i=1}^n$, we have:

$$\text{Validity}(\{\mathbf{x}_i\}_{i=1}^n) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{\exists j, k, \text{ s.t. } d(\mathbf{x}_i[j], \mathbf{x}_i[k]) < d_0 \text{ and } |j - k| > 2\}, \tag{33}$$

where $\mathbf{x}[j] \in \mathbb{R}^3$ indicates the coordinate of the j's atom in conformation $\mathbf{x}$, $d_0 > 0$ is the distance threshold to discriminate steric clash, and the contacting pairs are only considered to be those beyond the 2-hop neighborhood along the chain ($|j - k| > 2$). In practice, we choose d_0 according to the van der Waals radius of C_α (1.7Å) minus an allowable overlap δ_d, or formally $d_0 = 2 \times 1.7 - \delta_d$ (unit : Å).

Algorithm 2 Forward-backward dynamics (VPSDE)

Require: input conformation $\mathbf{x}$, constant perturbation scale δ; time upper bound for diffusion process T; score network $\mathbf{s}_\theta$; noise scale function β; data covariance $\mathbf{\Sigma}$; inverse temperature τ.
$\mathbf{x}_0 \leftarrow \mathbf{x}$ *// initialize state*
$\{t_1, \dots, t_M\} \leftarrow \text{Discretize}([0, \delta T])$ *// discretize time domain*
for $i = 1$ to M **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \sqrt{1 - \beta(t_i)}\mathbf{x}_{i-1} + \sqrt{\beta(t_i)}\mathbf{z}$
for $i = M - 1$ to 0 **do**
  $\mathbf{R}_{i+1} \leftarrow \mathbf{R}(t_{i+1}; \mathbf{\Sigma}, \tau)$ // in Eq. (29)
  $\mathbf{s}'_{i+1} \leftarrow \mathbf{R}_{i+1}\mathbf{s}_\theta(\mathbf{x}_{i+1}, t_{i+1})$ *// score rescaling*
  $\mathbf{x}'_i \leftarrow \left[2 - \sqrt{1 - \beta(t_{i+1})}\right]\mathbf{x}_{i+1} + \beta(t_{i+1})\mathbf{s}'_{i+1}$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}'_i + \sqrt{\beta(t_{i+1})}\mathbf{z}$
return $\mathbf{x}_0$

The default value of δ_d is set to be 0.4Å, which is a reasonable value when examining protein-protein interactions [44].

Fidelity The fidelity of a set of conformations is evaluated by measuring the similarity between the distribution of the reference ensemble $\mathbf{X}$ and the distribution of generated ensemble $\tilde{\mathbf{X}}$, which takes the idea of Fréchet inception distance (FID) [25] for the evaluation of image synthesis. Following [30], we adopt two distributions that can loyally reflect the ensemble characteristics: (1) pairwise distance distribution, a $N \times N \times C$ tensor $\boldsymbol{D}$ (N is the number of atoms of each conformation), whose element $\boldsymbol{D}[i, j] = (p_1^{(ij)}, \dots, p_C^{(ij)})$ records the discretized distance distribution $p^{(ij)} = \text{Cat}(p_1^{(ij)}, \dots, p_C^{(ij)})$ of a pair of atom i and j among conformations in ensemble. Following [30], the distance range between minimum and maximum values is equally divided by $C = 50$ bins to give the categorical distribution; (2) radius of gyration distribution, where similar discretization treatment is applied same as above. On top of these, the Jensen-Shannon (JS) divergences is applied for both distributions between the reference ensemble and generated ensemble due to its symmetric property. It takes the following form (D_{KL} denotes the Kullback–Leibler (KL) divergence, p, q are distribution):

$$D_{JS}(p \parallel q) = \frac{1}{2}D_{KL}(p \parallel m) + \frac{1}{2}D_{KL}(q \parallel m), \text{ where } m = \frac{1}{2}(p + q). \tag{34}$$

Note that there exists $N(N-1)/2$ pairs of atoms in a N-atom conformation. To derive a scalar divergence between two input ensembles, we use the averaged JS divergence [30] over all atom pairs, or formally:

$$D_{JS}^{\text{avg}}(\mathbf{X} \parallel \tilde{\mathbf{X}}) = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=i+1}^{N} D_{JS}(p^{(ij)} \parallel \tilde{p}^{(ij)}). \tag{35}$$

Since the distribution of radius of gyration is uni-dimensional, the JS divergence is applied as usual.

Diversity The diversity of the ensemble of interest can be derived from any structural similarity score by enumerating and averaging the pairwise scores between two members of that ensemble. Here we adopt two most commonly used scoring functions: root mean square deviation (RMSD) and TM-score [76]. RMSD reflects the deviation degree in length (here we use nanometer (nm) as the unit) and is thus unnormalized. TM-score, on the contrary, is a normalized score to evaluate the structural similarity between two input structures, ranging from 0 to 1 and unit-free. A higher TM-score indicates that two structures share more similarity. Given above, the diversity (Div) of an ensemble $\mathbf{X} \coloneqq \{\mathbf{x}_i\}_{i=1}^n$ is defined as follows:

$$\text{Div-RMSD}(\mathbf{X}) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{RMSD}(\mathbf{x}_i, \mathbf{x}_j), \tag{36}$$

and

$$\text{Div-TM}(\mathbf{X}) = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left(1 - \text{TM}(\mathbf{x}_i, \mathbf{x}_j)\right), \tag{37}$$

where we apply the inverse score $[1 - \text{TM}(\cdot, \cdot)]$ to express diversity (the higher score the more diverse) aligned with RMSD. During evaluation, both scores are calculated using the officially released binary from [76].

C Implementation of other methods

C.1 AlphaFold2 structure prediction

The AlphaFold2 (AF2) [32] predictions rely on an AF2 variant ParaFold [77], which exploits the CPU parallelism to accelerate the multiple sequence alignment (MSA) query during the feature preparation stage and keeps the rest intact. We adapted the original pipelines to exploit AlphaFold2's mechanisms for sampling protein conformations. The 12 fast-folding proteins and BPTI were processed using all five AlphaFold pTM models, enabling a broad capture of potential conformations to enhance the prediction accuracy. Two distinct input settings, *reducedMSA* and *noMSA*, were used; the former simplified the MSA input by limiting concurrent sequence alignments, while the latter omitted the MSA entirely and and replaced it with a single sequence. In both settings, structure templates were not included to enhance the diversity of sampled conformations. This protocol is inspired by the protocol of AlphaFold2-RAVE [65], which exploited the potential capacity of AlphaFold2 to generate many possible conformations.

Specifically, in the *reducedMSA* setting, JackHMMER and HHBlits were used for searching MSA from databases including UniRef90, MGnify, and BFD, following AlphaFold2's original pipeline [32]. Only a small portion of the found MSA was used as input to compute the MSA representation fed to Evoformer, consistent with the method in [18, 65], `'max_extra_msa'` and `'max_msa_clusters'` were set to be 16 and 8 respectively. Conversely, the *noMSA* setting created a dummy MSA file as precomputed MSA for AlphaFold2 and as a result only a single sequence is used to predict structure. For both settings, each pTM model was repeated 20 times with varying random seeds.

C.2 Full-atom molecular dynamics simulations

To compare SENS with traditional molecular dynamics methods, we employ OpenMM (version 8.0) [21] to conduct short MD simulations on our fast-folding protein benchmark systems as well as BPTI. Initial structures were taken from long simulations of [38] and [51]. AMBER ff14SB [40] was used as the protein force-field and TIP3P was used as solvent model. All systems were neutralized and solvated in the boxes of 10 Å. All bonds involving hydrogen atoms were constrained with the SHAKE algorithm [14, 47]. The particle mesh Ewald (PME) algorithm [16] was used to calculate the long-range electrostatic interactions. Initial structures were relaxed with minimization until convergence, then subjected to the equilibration stage with 10 ps time step size in the NVT and NPT ensemble sequentially. The simulation temperature of each system was set according to the original paper [38]. Another independent simulation, with a 30K higher temperature under the same setting, was used for comparison, which we named `'high temp.'`.

D Extended results

D.1 Per system pairwise distance and Rg distributions

In this section, we present the per system pairwise distance and radius of gyration distributions of all fast-folding protein systems, which are used to calculate the average fidelity scores shown in our

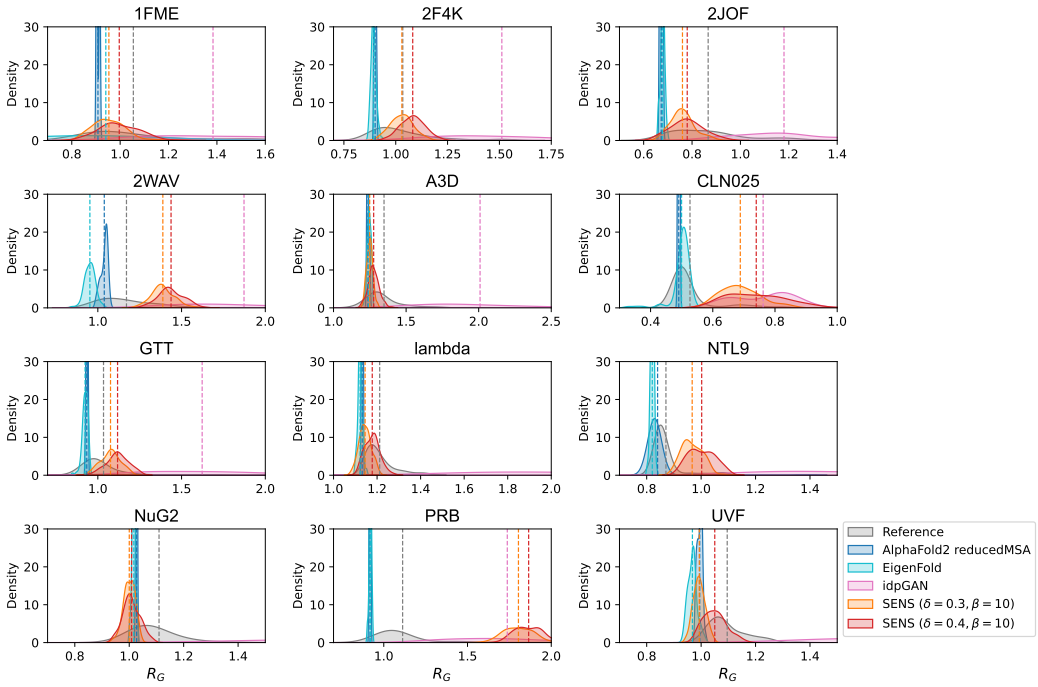


Figure S2: Rg distribution across all benchmarking systems. The shaded regions illustrate the spread of the Rg values for individual samples, while the dashed vertical lines mark the means of respective distributions.

main result. In Fig S2, the Rg distributions of selected methods are shown. In Fig. S3 and S4, the contacting probability distribution of different methods across all 12 benchmarking systems is shown, and each row indicates a specific protein system with the reference (full MD) result aligned in the rightmost column. The sampled conformations come from VESDE implementation of SENS as suggested in the main text.

D.2 More results for protein structural dynamics

We also tested the performance of SENS using VPSDE to construct the forward-backward dynamics proposed in the paper. The evaluation results under different hyperparameters are shown in Table S1, and are aligned with the VESDE counterpart. The results demonstrate that in general VPSDE can better capture the conformational distribution (lower JS divergence on average and higher diversity), yet cause more structural clashes (lower validity) in the sampled conformations. This scenario could result from the decaying drift term $-\frac{1}{2}\beta(t)\mathbf{x}$ in VPSDE and not in VESDE over-destroying the input structure during forward process. In particular, a β value of 5.0 can result in a temperature that is excessively low for VPSDE—though not for VESDE—impacting the generation of non-clashing conformations. In practice, one might need to carefully and heuristically tune the parameters δ, β for different systems and when using different diffusion dynamics to find the best value.

D.3 Apo/Holo conformational diversity

Following [31, 48], we tested SENS with other non-MD baseline methods (AlphaFold2, idpGAN, EigenFold) on the apo/holo pairs of conformers [48]. To accommodate the training protein lengths, we adopt a subset of 66 pairs with length less than or equal to 256 residues out of 91 structure pairs proposed in [48]. For each target pair, a conformational ensemble of size 10 sampled from each model is tested. For evaluation, we adopted the same metric *TM_ens* defined in [31], which evaluates to what degree the resulting ensemble captures both the apo and holo states. By definition,

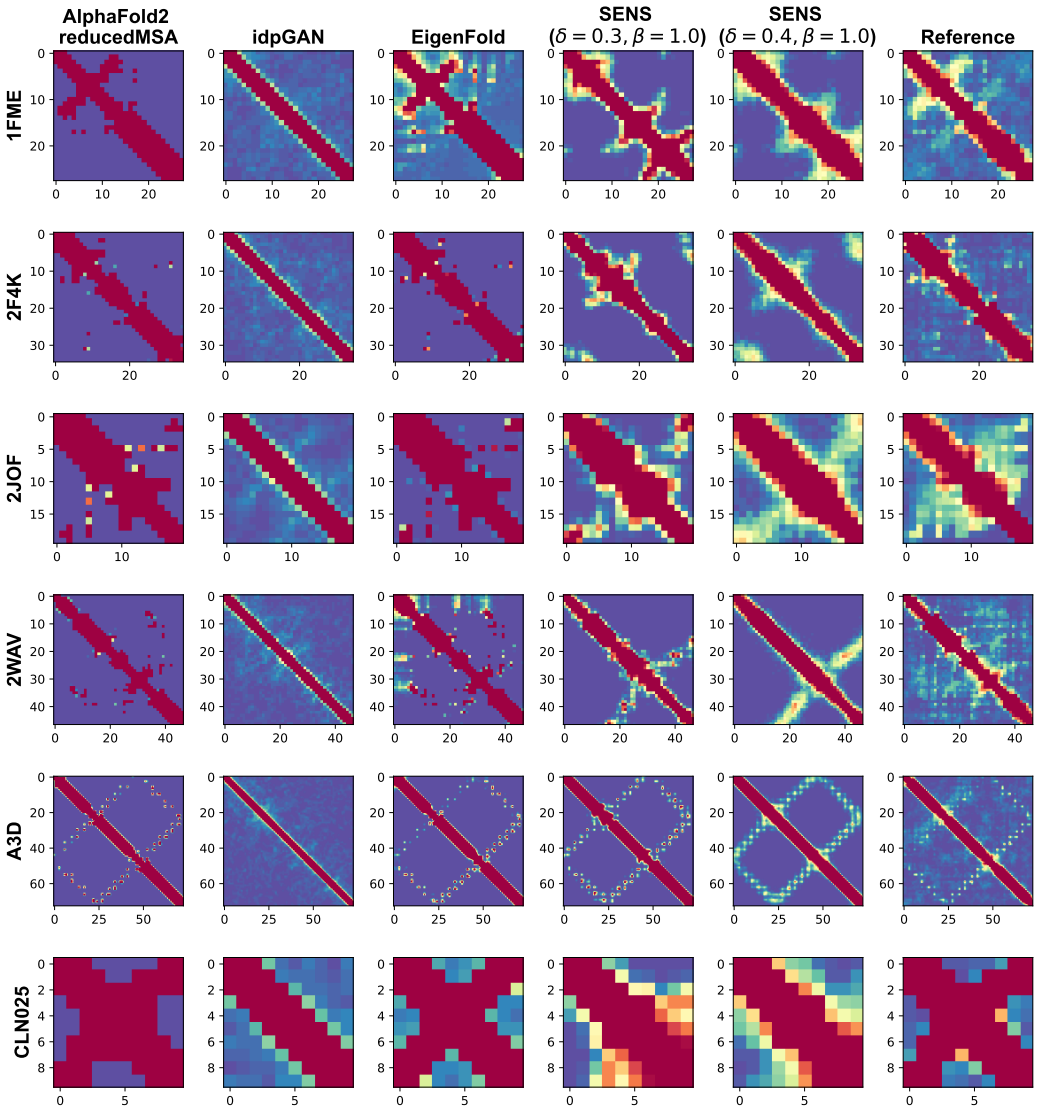


Figure S3: The contact probability distribution for all benchmarking systems. The color gradient illustrates the probability of contact, with red indicating a higher probability and blue a lower one. Contact is defined here as a Cα-Cα distance less than 8 Å.

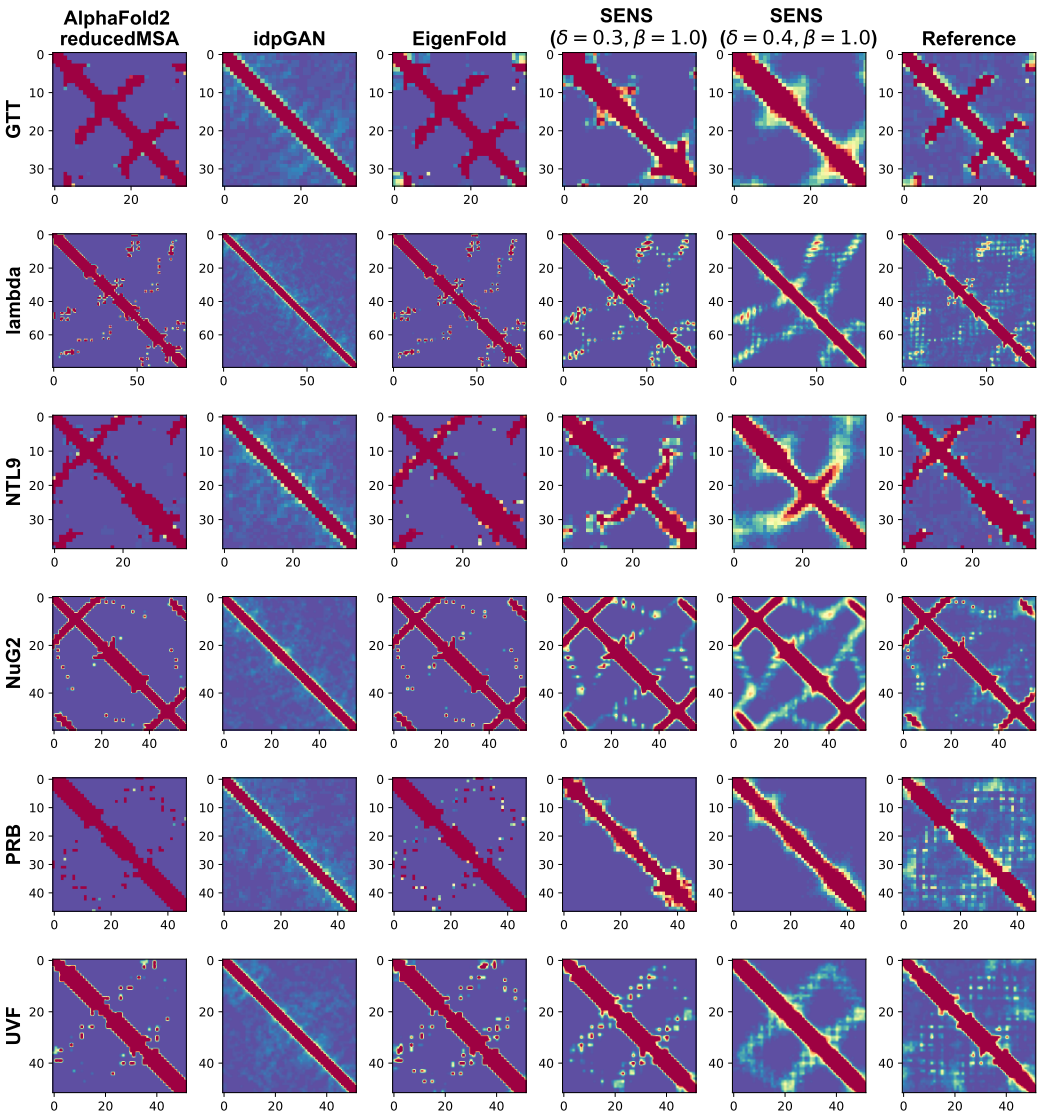


Figure S4: Extended part of Fig. S3.

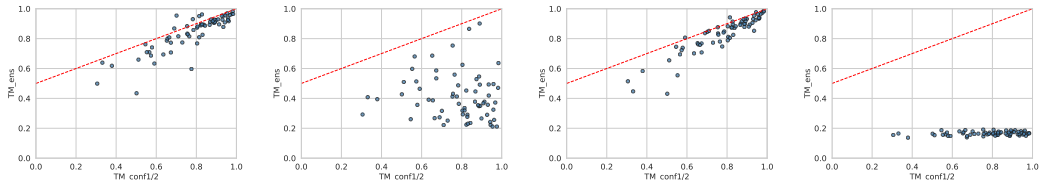
$$\mathrm{TM}_{\mathrm{ens}}(\mathbf{x}_{\mathrm{apo}}, \mathbf{x}_{\mathrm{holo}}, \{\mathbf{y}_i\}) = \frac{1}{2}\left[\max_i \mathrm{TM}(\mathbf{y}_i, \mathbf{x}_{\mathrm{apo}}) + \max_i \mathrm{TM}(\mathbf{y}_i, \mathbf{x}_{\mathrm{holo}})\right], \tag{38}$$

where $\mathbf{x}_{\mathrm{apo}}, \mathbf{x}_{\mathrm{holo}}$ denote the pair of ground truth apo/holo structures respectively, $\{\mathbf{y}_i\}$ indicates the generated ensemble. Similar to EigenFold, we plot the *TM_ens* scores for each apo/holo pair jointly with the intrinsic TM-score (calculated between apo/holo) denoted as *TM_conf1/2*. For baseline models, we input the protein sequence to obtain 10 different conformations; for SENS, we started with apo or holo structures and sampled 5 structures from either starting point.

As shown in Fig. S5, the results show that (generative) structure prediction models including AF2 and EigenFold can succeed in predicting only one form either apo or holo, and thus points are aggregated around the baseline result $y = 0.5x + 0.5$ (when the model can predict one single structure perfectly). Notably, when there is no MSA input, AF2 even failed to predict correctly single structure. For idpGAN, the generated ensembles show no relation with either apo or holo target such that the TM scores are all low. For the proposed SENS, since it had access to ground truth structure as starting point, the listed results demonstrate no improvements as well. As the δ increases, we find a declining

Table S1: Benchmark results of VE/VPSDE under different hyperparameters. Best results among those using VESDE or VPSDE are **bolded**.

	Validity(↑)	JS-D(↓)	JS-Rg (↓)	Div-TM(↑)	Div-RMSD(↑)
VPSDE ($\delta = 0.3, \beta = 5.0$)	0.521	0.275	0.549	0.769	0.730
VPSDE ($\delta = 0.4, \beta = 5.0$)	0.553	0.280	0.589	0.774	0.787
VPSDE ($\delta = 0.3, \beta = 2.0$)	0.989	0.222	0.430	0.761	0.743
VPSDE ($\delta = 0.4, \beta = 2.0$)	0.990	0.224	0.468	0.767	0.827
VPSDE ($\delta = 0.3, \beta = 1.0$)	0.959	**0.186**	**0.296**	0.814	0.878
VPSDE ($\delta = 0.4, \beta = 1.0$)	0.958	0.193	0.321	**0.819**	**0.932**
VESDE ($\delta = 0.3, \beta = 5.0$)	**1.000**	0.283	0.524	0.543	0.275
VESDE ($\delta = 0.4, \beta = 5.0$)	**1.000**	0.252	0.422	0.647	0.401
VESDE ($\delta = 0.3, \beta = 2.0$)	**1.000**	0.267	0.474	0.536	0.277
VESDE ($\delta = 0.4, \beta = 2.0$)	0.998	0.217	0.401	0.693	0.440
VESDE ($\delta = 0.3, \beta = 1.0$)	0.998	0.257	0.416	0.576	0.295
VESDE ($\delta = 0.4, \beta = 1.0$)	0.993	0.203	0.338	0.734	0.498
Short MD 0.1μs	1.000	0.217	0.300	0.508	0.449
Short MD 0.1μs (high temp.)	1.000	0.204	0.285	0.536	0.495
Reference 1μs	1.000	0.135	0.178	0.582	0.642
Reference 10μs	1.000	0.100	0.156	0.650	0.736
Reference full	1.000	0.000	0.000	0.659	0.756



(a) AF2 *reducedMSA* (b) AF2 *noMSA* (c) EigenFold (d) idpGAN

(e) SENS ($\delta = 0.2$) (f) SENS ($\delta = 0.3$) (g) SENS ($\delta = 0.4$) (h) SENS ($\delta = 0.5$)

Figure S5: The scatter plots of TM_ens versus TM_conf1/2 on the apo/holo pairing dataset from different methods. The red line in each subfigure is the plot of $y = 0.5x + 0.5$ for a reference baseline.

TM_ens score, which indicates the sampled ensembles are isolated from both apo and holo forms. Among all these methods, not a single model achieves significant improvement over baseline on the apo/holo task, the solving of which can be important and interesting to serve as a future research direction.

E Further discussion

In this paper, we have proposed the SENS for conformational sampling which leverages the bidirectional diffusion dynamics in SGMs. This method shares several aspects with (generative) structure prediction and protein backbone generation methods. Structure predictions, with representative methods AlphaFold2 [32] and RoseTTAFold [5], aim to recover the stablest protein folding state from its sequence. In terms of evaluation, crystal structures are used as ground truth to assess accuracy within the classical regression framework. Those methods that can predict highly accurate folding structure and generalize to unseen proteins (even quite a bit different from training data) are

encouraged and have a good potential to replace the experimental techniques such as X-ray diffraction and electron microscopy. The generative structure predictions, with the representative EigenFold [31], treat the protein folding task from discriminative prediction to conditional distribution learning, which shares the idea of DiffDock [15]. The step of "go generative" has a benefit as gaining predictive robustness yet may suffer from making additional assumption (discussion on this topic belongs to the structured prediction problem [7] in machine learning domains). One key assumption by employing generative protein structure prediction is that there exists multiple stable conformations from one single amino-acid sequence, which is common and reasonable to protein structures. However, one of the biggest concerns is the lack of ground truth multi-state structure data that belongs to a single sequence. We believe in the significance of exploring such topic not only for better augmenting the current folding methods but also for the potential application to promote the study of protein dynamics. Methods for generating protein backbones, such as ProtDiff [63], aim to complement the known protein space with *de novo*-designed backbone structures by generative models. The novel backbones are usually fed to an inverse folding model, for example ProteinMPNN [17], to further obtain the specific protein sequences that can potentially fold to those backbones. Such backbone generative models [29, 63, 68, 70, 75] effectively model the protein structure space and do yield protein-like decoys. The functionality of such generated decoys remains to be validated in the *in vitro* assays and seldom computational metrics can give good evaluations. Therefore, it is also promising to investigate such computational evaluation for the generated protein backbones. Moreover, since the goal of backbone generation is only to model the marginal distribution of backbone structures alone, one interesting question to be investigated is whether we can "go further generative" than the generative structure prediction models by modeling the joint distribution of both protein structures and sequences, generating structure and sequence in parallel similar to the frameworks of [2, 39, 52]. We consider the study of these as our potential future works.