PRE-TRAINING PROTEIN STRUCTURE ENCODER VIA SIAMESE DIFFUSION TRAJECTORY PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Inspired by the governing role of protein structures on protein functions, structure-based protein representation pre-training is recently gaining interest but remains largely unexplored. Along this direction, pre-training by mutual information (MI) maximization has shown some preliminary promise, while more in-depth studies are required to design more informative correlated views that share common information about a protein. Previous view designs focus on capturing structural motif co-occurrence in the same protein structure, while they cannot capture detailed atom- and residue-level interactions and also the statistical dependencies of residue types along protein sequences. To address these limitations, we propose the Siamese Diffusion Trajectory Prediction (**SiamDiff**) method. SiamDiff employs the **multimodal diffusion process** as a faithful simulation of the structure-sequence co-diffusion trajectory that gradually and smoothly approaches the folded structure and the corresponding sequence of a protein from scratch. Upon a native protein and its correlated counterpart obtained with random structure perturbation, we build two multimodal (structure and sequence) diffusion trajectories and regard them as two correlated views. A principled theoretical framework is designed to maximize the MI between such paired views, such that the model can acquire the atom- and residue-level interactions underlying protein structural changes and also the residue type dependencies. We study the effectiveness of SiamDiff on both residue-level and atom-level structural representations. Experimental results on EC and ATOM3D benchmarks show that the performance of SiamDiff is consistently competitive on all benchmark tasks, compared with existing baselines. The source code will be made public upon acceptance.

1 INTRODUCTION

In the past year, thanks to the rise of highly accurate and efficient protein structure predictors based on deep learning (Jumper et al., 2021; Baek et al., 2021), the gap between the number of reported protein sequences and corresponding (computationally) solved structures is greatly narrowed. These advances open the opportunity of self-supervised protein representation learning based on protein structures, *i.e.*, learning informative protein representations from massive protein structures without using any annotation. Compared to extensively studied sequence-based protein self-supervised learning like protein language models (Elnaggar et al., 2021; Rives et al., 2021), structure-based methods could learn more effective representations to indicate protein functions, since a protein sequence determines its structure, and the structure is the determinant of its diverse functions (Harms & Thornton, 2010).

To attain this goal, some recent works have explored different self-supervised learning strategies on protein structures, including contrastive learning (Zhang et al., 2022; Hermosilla & Ropinski, 2022), self-prediction (Zhang et al., 2022; Chen et al., 2022) and denoising score matching (Guo et al., 2022; Wu et al., 2022a). Among these works, mutual information (MI) maximization based methods (Zhang et al., 2022; Hermosilla & Ropinski, 2022) consistently achieve superior performance on predicting general protein functions and structural classes. At the core of these methods, correlated views are constructed to capture the co-occurrence of structural motifs on the same protein structure. However, such view construction scheme fails to capture detailed atom- and residue-level interactions and also the statistical dependencies of residue types along protein sequences. Therefore, the MI maximization based on these views may not produce effective representations for the tasks that require to model detailed local structures (*e.g.*, the Residue Identity task from ATOM3D (Townshend et al., 2020)) or

minor differences on structures and sequences due to point mutations (*e.g.*, the Mutation Stability Prediction task from ATOM3D), as demonstrated by the experimental results in Tabs. 1 and 2.

To tackle such limitation, in this work, we propose the Siamese Diffusion Trajectory Prediction (**SiamDiff**) method to jointly model fine-grained atom- and residue-level interactions and residue type dependencies. Specifically, given a native protein, we first derive its correlated counterpart by random structure perturbation which simulates its conformational fluctuation. We further extend the original protein and the generated counterpart respectively by the *multimodal diffusion process*, in which we transform both the protein structure and the protein sequence towards random distribution by gradually and smoothly adding noises. Such a diffusion process is verified as a faithful simulation of the structure-sequence co-diffusion trajectory by recent studies (Anand & Achim, 2022; Luo et al., 2022). We regard the diffusion trajectories of the original protein and the generated counterpart as two correlated views, and a principled theoretical framework is designed to maximize the MI between such paired views. Under such a learning framework, the model can acquire the atom- and residue-level interactions underlying protein structural changes and also the residue type dependencies. The learned protein representations are expected to boost diverse types of downstream tasks.

SiamDiff can be flexibly applied to both residue-level and atom-level structures for effective representation learning. We employ different self-supervised algorithms to pre-train residue-level and atom-level structure encoders, and the pre-trained models are extensively evaluated on Enzyme Commission number prediction (Gligorijević et al., 2021) and ATOM3D (Townshend et al., 2020) benchmarks. Experimental results verify that SiamDiff can consistently achieve competitive performance on all benchmark tasks and on both structure levels, in contrast to existing baselines.

2 Related Work

Protein Structure Representation Learning. The community witnessed a surge of research interests in learning informative protein structure representations using structure-based encoders and training algorithms. The encoders are designed to capture protein structural information on different granularity, including residue-level structures (Gligorijević et al., 2021; Wang et al., 2022b; Zhang et al., 2022), atom-level structures (Hermosilla et al., 2021; Jing et al., 2021a; Wang et al., 2022a) and protein surfaces (Gainza et al., 2020; Sverrisson et al., 2021; Somnath et al., 2021).

Recent works study pre-training on massive unlabeled protein structures for generalizable representations, covering contrastive learning (Zhang et al., 2022; Hermosilla & Ropinski, 2022), self-prediction of geometric quantities (Zhang et al., 2022; Chen et al., 2022) and denoising score matching (Guo et al., 2022; Wu et al., 2022a). All these methods only employ native proteins for pre-training. By comparison, the proposed SiamDiff uses the information from multimodal diffusion trajectories to better acquire atom- and residue-level interactions and residue type dependencies.

Diffusion Probabilistic Models (DPMs). DPM was first proposed in Sohl-Dickstein et al. (2015) and has been recently rekindled for its strong performance on image and waveform generation (Ho et al., 2020; Chen et al., 2020a). Recent works (Nichol & Dhariwal, 2021; Song et al., 2020; 2021) have improved training and sampling for DPMs. Besides the DPMs for continuous data, some works study discrete DPMs and achieve impressive results on generating texts (Austin et al., 2021; Li et al., 2022), images (Austin et al., 2021) and image segmentation data (Hoogeboom et al., 2021).

Inspired by these progresses, DPMs have been recently adopted to solve the problems in chemistry and biology domain, including molecule generation (Xu et al., 2022; Hoogeboom et al., 2022; Wu et al., 2022b), molecular representation learning (Liu et al., 2022), protein design (Anand & Achim, 2022; Luo et al., 2022) and motif-scaffolding (Trippe et al., 2022). In this work, we novelly study how DPMs can help protein representation learning, which aligns with a recent effort (Abstreiter et al., 2021) on diffusion-based image representation learning.

3 Preliminaries

3.1 Problem Definition

Notations. A protein with n_r residues (amino acids) and n_a atoms can be represented as a sequence-structure tuple $\mathcal{P} = (\mathcal{S}, \mathcal{R})$. We use $\mathcal{S} = [s_1, s_2, \cdots, s_{n_r}]$ to denote its sequence with s_i as the type of the i-th residue, while $\mathcal{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2..., \boldsymbol{r}_{n_a}] \in \mathbb{R}^{n_a \times 3}$ denotes its structure with $\boldsymbol{r}_i$ as the Cartesian

coordinates of the i-th atom. In the paper, we construct a graph for each protein with edges connecting atoms with the Euclidean distance lower than a threshold. Besides, we also consider residue-level protein graphs, a concise version of atom graphs that enable efficient message passing between nodes and edges. As in Zhang et al. (2022), we only keep the alpha carbon atom of each residue and add sequential, radius and K-nearest neighbor edges as different types of edges.

Equivariance. *Equivariance* is ubiquitous in machine learning for modeling the symmetry in physical systems (Thomas et al., 2018; Weiler et al., 2018; Batzner et al., 2022) and is shown to be critical for successful design and better generalization of 3D neural networks (Köhler et al., 2020). Formally, a function $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ is equivariant *w.r.t.* a group G if $\mathcal{F} \circ \rho_{\mathcal{X}}(x) = \rho_{\mathcal{Y}} \circ \mathcal{F}(x)$, where $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$ are transformations corresponding to an element $g \in G$ acting on $\mathcal{X}$ and $\mathcal{Y}$, respectively. The function is invariant *w.r.t* G if the transformations $\rho_{\mathcal{Y}}$ is identity. In this paper, we consider the SE(3) group, *i.e.*, rotations and translations in 3D space.

Problem Definition. Given a set of unlabeled proteins $\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, ...\}$, our goal is to train a protein encoder $\phi(\mathcal{S}, \mathcal{R})$ to extract representations that are SE(3)-invariant *w.r.t.* protein structures $\mathcal{R}$.

3.2 Diffusion Models on Proteins

Diffusion models are a class of deep generative models with latent variables encoded by a *forward diffusion process* and decoded by *a reverse generative process* (Sohl-Dickstein et al., 2015). There have been recent efforts on applying denoising diffusion models for protein generation (Luo et al., 2022; Anand & Achim, 2022). We use $\mathcal{P}^0$ to denote the ground-truth protein and $\mathcal{P}^t$ for $t = 1, \cdots, T$ to be the latent variables over T diffusion steps. Modeling the protein as an evolving thermodynamic system, the forward process gradually injects small noise to the data $\mathcal{P}^0$ until reaching a random noise distribution at time T. The reverse process learns to denoise the latent variable towards the data distribution. Both processes are defined as Markov chains:

$$q(\mathcal{P}^{1:T}|\mathcal{P}^0) = \prod_{t=1}^{T} q(\mathcal{P}^t|\mathcal{P}^{t-1}), \quad p_\theta(\mathcal{P}^{0:T-1}|\mathcal{P}^T) = \prod_{t=1}^{T} p_\theta(\mathcal{P}^{t-1}|\mathcal{P}^t), \tag{1}$$

where $q(\mathcal{P}^t|\mathcal{P}^{t-1})$ defines the forward process at step t and $p_\theta(\mathcal{P}^{t-1}|\mathcal{P}^t)$ with learnable parameters θ defines the reverse process at step t.

The generation of a protein relies on the joint diffusion process on structures and sequences. Following Luo et al. (2022), we assume 1) the separate definition of the forward process on structures and sequences and 2) the conditional independence of sequences and structures in the reverse process:

$$q(\mathcal{P}^t|\mathcal{P}^{t-1}) = q(\mathcal{R}^t|\mathcal{R}^{t-1}) \cdot q(\mathcal{S}^t|\mathcal{S}^{t-1}), \quad p_\theta(\mathcal{P}^{t-1}|\mathcal{P}^t) = p_\theta(\mathcal{R}^{t-1}|\mathcal{P}^t) \cdot p_\theta(\mathcal{S}^{t-1}|\mathcal{P}^t). \tag{2}$$

Next, we discuss how to define the diffusion models on protein structures and sequences, respectively.

Diffusion models on 3D structures. Since the coordinates of atoms are continuous variables in the 3D space, the forward process can be defined by adding random Gaussian noise. Then, the reverse process can be parameterized as a Gaussian with a learnable mean and user-defined variance. That is,

$$q(\mathcal{R}^t|\mathcal{R}^{t-1}) = \mathcal{N}(\mathcal{R}^t; \sqrt{1-\beta_t}\mathcal{R}^{t-1}, \beta_t I), \quad p_\theta(\mathcal{R}^{t-1}|\mathcal{P}^t) = \mathcal{N}(\mathcal{R}^{t-1}; \mu_\theta(\mathcal{P}^t, t), \sigma_t^2 I), \tag{3}$$

where $\beta_1, ..., \beta_T$ are a series of fixed variances and σ_t can be any user-defined variance. Since $\mathcal{R}^t$ is available as an input, following Ho et al. (2020), we reparameterize the mean $\mu_\theta(\mathcal{P}^t, t)$ as:

$$\mu_\theta(\mathcal{P}^t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathcal{R}^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathcal{P}^t, t) \right), \tag{4}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ and the neural network $\epsilon_\theta(\cdot)$ learns to decorrupt the data and should be translation-invariant and rotation-equivariant *w.r.t.* the protein structure $\mathcal{R}^t$.

Diffusion models on sequences. The key of diffusion models on discrete attributes is to define transition matrices for the Markov chain. Austin et al. (2021) adds an absorbing state [MASK] for the Markov chain and then each residue either stays the same or transits to [MASK] with some probability at each step. For the reverse process, a neural network $\tilde{p}_\theta$ is defined to predict the probability of $\mathcal{S}^0$ and then parameterize the diffusion trajectory with the probability $q(\mathcal{S}^{t-1}|\mathcal{S}^t, \tilde{\mathcal{S}}^0)$. That is,

$$q(\mathcal{S}^t|\mathcal{S}^{t-1}) = \text{Cat}\left(\mathcal{S}^t; \boldsymbol{p} = \boldsymbol{s}^{t-1}\boldsymbol{Q}^t\right), \quad p_\theta(\mathcal{S}^{t-1}|\mathcal{P}^t) \propto \sum_{\tilde{\mathcal{S}}^0} q(\mathcal{S}^{t-1}|\mathcal{S}^t, \tilde{\mathcal{S}}^0) \cdot \tilde{p}_\theta(\tilde{\mathcal{S}}^0|\mathcal{P}^t), \tag{5}$$

where $\boldsymbol{s}^t \in \mathbb{R}^{n_r \times 21}$ denotes the one-hot feature for the sequence $\mathcal{S}^t$, $\boldsymbol{Q}^t \in \mathbb{R}^{21 \times 21}$ denotes the corresponding transition matrix at step t, Cat$(\cdot)$ is the categorical distribution and $\tilde{\mathcal{S}}^0$ enumerates all possible residue types.

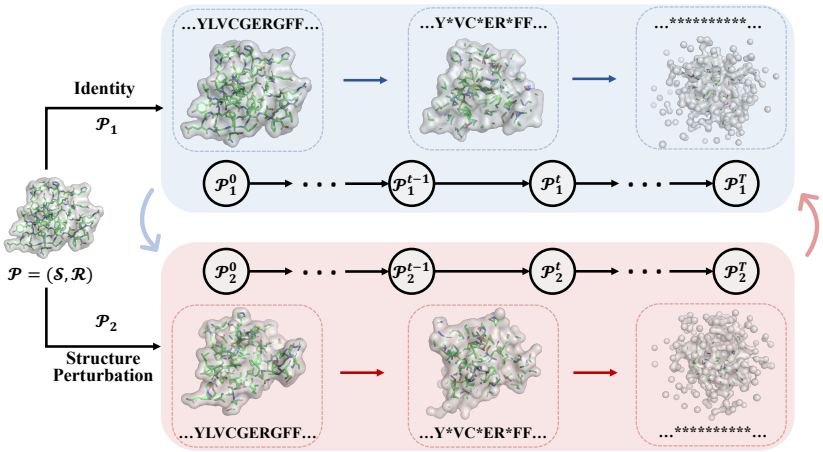

Figure 1: High-level illustration of SiamDiff. Given a protein $\mathcal{P} = (\mathcal{S}, \mathcal{R})$, we first obtain two correlated starting states $\mathcal{P}_1$ and $\mathcal{P}_2$ via identity and structure perturbation functions, respectively. Then, we apply a multimodal diffusion process on the starting states to construct two siamese trajectories, where we use * to denote masked residues. To maximize the mutual information between these trajectories, we predict the noises on one trajectory with representations from the other.

4 SiamDiff: Siamese Diffusion Trajectory Prediction

4.1 Overview

Recent self-supervised learning approaches learn informative representations by maximizing mutual information (MI) between representations of multiple views of the same object with shared information, whose effectiveness has been proved on images (Chen et al., 2020b), natural languages (Hjelm et al., 2018), graphs (Hassani & Khasahmadi, 2020), small molecules (Liu et al., 2021) and proteins (Zhang et al., 2022). For MI maximization based protein representation learning, previous works (Zhang et al., 2022; Hermosilla & Ropinski, 2022) only focus on capturing protein substructure co-occurrences and learning representations for the entire protein, but fail to capture detailed atom- and residue-level interactions and also the statistical dependencies of residue types along protein sequences. To tackle this limitation, in this work, we incorporate protein diffusion trajectories into the MI maximization framework. Because these trajectories can facilitate the modeling of atom- and residue-level interactions under structural changes and residue type dependencies.

Correlated protein diffusion trajectories construction. Our framework is based on the contrast of correlated protein trajectories, *a.k.a.*, siamese trajectories. These trajectories are generated by a *multimodal diffusion process*, which is verified to be effective for protein structure-sequence co-design (Anand & Achim, 2022; Luo et al., 2022) and thus can reflect physical and chemical principles underlying protein formation (Anfinsen, 1972; Dill & MacCallum, 2012).

Specifically, we first construct two correlated states from one protein as the starting points of the trajectories. Given the original protein $\mathcal{P} = (\mathcal{S}, \mathcal{R})$, we deem it as the native state $\mathcal{P}_1 = (\mathcal{S}_1, \mathcal{R}_1)$ and build a correlated state $\mathcal{P}_2 = (\mathcal{S}_2, \mathcal{R}_2)$ by randomly perturbing the protein structure, *i.e.*, $\mathcal{S}_2 = \mathcal{S}_1, \mathcal{R}_2 = \mathcal{R}_1 + \epsilon$, where $\epsilon \in \mathbb{R}^{n_a \times 3}$ is the noise drawn from a normal distribution. Such structure perturbation simulates the conformational fluctuation of a protein (Somero, 1995; Akasaka, 2006) and is thus a valid scheme for correlated states acquisition.

Then, for states $\mathcal{P}_1$ and $\mathcal{P}_2$, we sample their diffusion trajectories $\mathcal{P}_1^{0:T}$ and $\mathcal{P}_2^{0:T}$ with the mutlimodal diffusion process defined in Sec. 3.2. Take $\mathcal{P}_1 = (\mathcal{S}_1, \mathcal{R}_1)$ for example. We start the diffusion process from the state $\mathcal{P}_1^0 = \mathcal{P}_1$. The diffusion process is defined by the joint diffusion on structures and sequences, *i.e.*, $q(\mathcal{P}_1^{1:T}|\mathcal{P}_1^0) = q(\mathcal{R}_1^{1:T}|\mathcal{R}_1^0)q(\mathcal{S}_1^{1:T}|\mathcal{S}_1^0)$. We utilize the conditional Gaussian distributions in (3) to derive trajectories on structures $\mathcal{R}_1^{1:T}$ and the discrete Markov chain in (5) to derive the sequence diffusion processes $\mathcal{S}_1^{1:T}$. In this way, we define the trajectory $\mathcal{P}_1^{0:T} = \{(\mathcal{S}_1^t, \mathcal{R}_1^t)\}_{t=0}^T$ for $\mathcal{P}_1$ and can derive the siamese trajectory $\mathcal{P}_2^{0:T} = \{(\mathcal{S}_2^t, \mathcal{R}_2^t)\}_{t=0}^T$ similarly.

Maximizing mutual information (MI) between representations from siamese trajectories. The main focus of our approach is to maximize the MI between representations of siamese trajectories constructed as above (see App. A for related works about MI maximization). For notations, we use the bold symbol to denote the representation of an object and use $\boldsymbol{P}_1^{0:T}$ and $\boldsymbol{P}_2^{0:T}$ to denote the corresponding random variables of representations of the siamese trajectories $\boldsymbol{\mathcal{P}}_1^{0:T}$ and $\boldsymbol{\mathcal{P}}_2^{0:T}$. Because directly optimizing MI is intractable, we instead maximize an approximate lower bound of MI described in the following proposition (see proof in App. B.1).

Proposition 1 *With some approximations, the mutual information between representations of two siamese trajectories is lower bounded by:*

$$I(\boldsymbol{P}_1^{0:T}; \boldsymbol{P}_2^{0:T}) \geq -\frac{1}{2}(\mathcal{L}^{(2\to 1)} + \mathcal{L}^{(1\to 2)}) + C,$$

where C is a constant independent of our encoder and the term from trajectory $\mathcal{P}_b^{0:T}$ to $\mathcal{P}_a^{0:T}$ is defined as

$$\mathcal{L}^{(b\to a)} := \mathbb{E}_{\mathcal{P}_a^{0:T}, \mathcal{P}_b^{0:T}} \left[\sum_{t=1}^{T} D_{KL} \left(q(\mathcal{P}_a^{t-1} | \mathcal{P}_a^t, \mathcal{P}_a^0) || p(\mathcal{P}_a^{t-1} | \mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^{0:T}) \right) \right], \tag{6}$$

with $b \to a$ being either $2 \to 1$ or $1 \to 2$.

The two terms share the similar formula as the ELBO loss in diffusion models (Ho et al., 2020). Take $\mathcal{L}^{(2\to 1)}$ for example. Here $q(\mathcal{P}_1^{t-1} | \mathcal{P}_1^t, \mathcal{P}_1^0)$ is a posterior analytically tractable with our definition of each diffusion step $q(\mathcal{P}_1^t | \mathcal{P}_1^{t-1})$ in (3) and (5). The reverse process is learnt to generate a less noisy state $\mathcal{P}_1^{t-1}$ given the current state $\mathcal{P}_1^t$ and representations of the siamese trajectory $\boldsymbol{\mathcal{P}}_2^{0:T}$, which are extracted by the protein encoder to be pre-trained.

Essentially, we perform mutual prediction between two siamese trajectories, which is similar to the idea of mutual representation reconstruction in Grill et al. (2020); Chen & He (2021). In practice, though adding structural noises, $\mathcal{P}_1$ and $\mathcal{P}_2$ share information about the same protein and thus the whole trajectory of $\mathcal{P}_2$ may give too many clues for the denoising towards $\mathcal{P}_1^t$, which is harmful for pre-training. To address this issue, we parameterize $p(\mathcal{P}_1^{t-1} | \mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^{0:T})$ with $p_\theta(\mathcal{P}_1^{t-1} | \mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)$. For diffusion on sequences, we further guarantee that the same set of residues are masked in $\mathcal{S}_1^t$ and $\mathcal{S}_2^t$ to avoid the leakage of ground-truth residue types across views.

By optimizing the objective in (6), the model can acquire common information shared by siamese trajectories and also learn the atom- and residue-level interactions and residue type dependencies. The graphical overview of our method is illustrated in Fig. 1. Below we describe the parameterization of the generation process $p_\theta(\mathcal{P}_1^{t-1} | \mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)$ in Sec. 4.2, derive the pre-training objective from (6) in Sec. 4.3, and discuss advantages of our method and its connection with existing works in Sec. 4.4.

4.2 Modeling Protein Generation Process

Different from the generation process in traditional diffusion models, the parameterization of $p_\theta(\mathcal{P}_1^{t-1} | \mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)$ should inject information from $\mathcal{P}_2^t$. Therefore, we apply an SE(3)-invariant encoder $\phi(\cdot)$ to $\mathcal{P}_2^t$, and the extracted atom and residue representations (denoted as $\boldsymbol{a}_2^t$ and $\boldsymbol{h}_2^t$) are employed for this denoising step. Given the conditional independence in (2), this generation process can be decomposed into that on protein structures and sequences as shown below.

Generation process on protein structures. As in (4), modeling the generation process of protein structures is to model the noise on $\mathcal{R}_1^t$ and gradually decorrupt the noisy structure. This can be parameterized with a noise prediction network $\epsilon_\theta(\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t, t)$ that is translation-invariant and rotation-equivariant *w.r.t.* $\mathcal{R}_1^t$. Besides, the noise applied on $\mathcal{R}_1^t$ should not change with transformations on $\mathcal{R}_2^t$, so ϵ_θ should be SE(3)-invariant *w.r.t.* $\mathcal{R}_2^t$.

To achieve these goals, we define our noise prediction network with atom representations $\boldsymbol{a}_2^t$ (which is SE(3)-invariant *w.r.t.* $\mathcal{R}_2^t$) and atom coordinates $\boldsymbol{r}_1^t$ (which is SE(3)-equivariant *w.r.t.* $\mathcal{R}_1^t$). We draw inspirations from recent works (Satorras et al., 2021) to build an equivariant output based on normalized directional vectors between adjacent atom pairs. Each edge (i, j) is encoded by its length $\|\boldsymbol{r}_{1i}^t - \boldsymbol{r}_{1j}^t\|_2$ and the representations of two end nodes $\boldsymbol{a}_{2i}^t$, $\boldsymbol{a}_{2j}^t$, and the encoded score $m_{i,j}$ will be further used for aggregating directional vectors. Specifically, we have

$$[\epsilon_\theta(\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t, t)]_i = \sum_{j \in \mathcal{N}_1^t(i)} m_{i,j} \cdot \frac{\boldsymbol{r}_{1i}^t - \boldsymbol{r}_{1j}^t}{\|\boldsymbol{r}_{1i}^t - \boldsymbol{r}_{1j}^t\|_2}, \text{ with } m_{i,j} = \text{MLP}(\boldsymbol{a}_{2i}^t, \boldsymbol{a}_{2j}^t, \text{MLP}(\|\boldsymbol{r}_{1i}^t - \boldsymbol{r}_{1j}^t\|_2)),$$

where $\mathcal{N}_1^t(i)$ denotes the neighbors of the atom i in the corresponding graph of $\mathcal{P}_1^t$. Note that $\epsilon_\theta(\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t, t)$ achieves the equivariance requirement, as $m_{i,j}$ is SE(3)-invariant *w.r.t.* $\mathcal{R}_1^t$ and $\mathcal{R}_2^t$ while $\boldsymbol{r}_{1i}^t - \boldsymbol{r}_{1j}^t$ is translation-invariant and rotation-equivariant *w.r.t.* $\mathcal{R}_1^t$.

Generation process on protein sequences. As in (5), the generation process on sequences aims to predict masked residue types in $\mathcal{S}_1^0$ with a predictor $\tilde{p}_\theta$. In our setting of mutual prediction, we define the predictor based on representations of the same residues in $\mathcal{S}_2^t$, which are also masked. Hence, for each masked residue i in $\mathcal{S}_2^t$, we feed its representation $\boldsymbol{h}_{2i}^t$ to an MLP and predict the type of the corresponding residue type s_{1i}^0 in $\mathcal{S}_1^0$:

$$\tilde{p}_\theta(\mathcal{S}_1^0|\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t) = \textstyle\prod_i \tilde{p}_\theta(s_{1i}^0|\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t) = \prod_i \text{Softmax}(s_{1i}^0|\text{MLP}(\boldsymbol{h}_{2i}^t)),$$

where the softmax function is applied over all residue types.

4.3 Pre-training Objective

Given the defined forward and reverse process on two trajectories, we now derive the pre-training objective based on the mutual diffusion loss in (6). We take the term $\mathcal{L}^{(2\to1)}$ for example and its counterpart can be derived in the same way. It can be proved that with the independence assumptions in (2), the objective can be decomposed into a structure loss $\mathcal{L}^{(\mathcal{R},2\to1)}$ and a sequence loss $\mathcal{L}^{(\mathcal{S},2\to1)}$ (see proof in App. B.2):

$$\mathcal{L}^{(\mathcal{R},2\to1)} := \mathbb{E}\left[\textstyle\sum_{t=1}^T D_{\text{KL}}\left(q(\mathcal{R}_1^{t-1}|\mathcal{R}_1^t, \mathcal{R}_1^0)||p_\theta(\mathcal{R}_1^{t-1}|\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)\right)\right], \tag{7}$$

$$\mathcal{L}^{(\mathcal{S},2\to1)} := \mathbb{E}\left[\textstyle\sum_{t=1}^T D_{\text{KL}}\left(q(\mathcal{S}_1^{t-1}|\mathcal{S}_1^t, \mathcal{S}_1^0)||p_\theta(\mathcal{S}_1^{t-1}|\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)\right)\right]. \tag{8}$$

Structure loss $\mathcal{L}^{(\mathcal{R},2\to1)}$. It has been shown in Ho et al. (2020) that the loss function can be simplified under our parameterization by calculating KL divergence between Gaussians as weighted L2 distances between means ϵ_θ and ϵ (see details in App. B.3):

$$\mathcal{L}^{(\mathcal{R},2\to1)} = \textstyle\sum_{t=1}^T \gamma_t \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\|\epsilon - \epsilon_\theta(\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t, t)\|_2^2\right], \tag{9}$$

where the coefficients γ_t are determined by the variances $\beta_1, ..., \beta_t$. In practice, we follow the suggestion in Ho et al. (2020) to set all weights $\gamma_t = 1$ for the simplified loss $\mathcal{L}_{\text{simple}}^{(\mathcal{R},2\to1)}$.

Since ϵ_θ is designed to be rotation-equivariant *w.r.t.* $\mathcal{R}_1^t$, to make the loss function invariant *w.r.t.* $\mathcal{R}_1^t$, the supervision ϵ is also supposed to achieve such equivariance. So we adopt the chain-rule approach proposed in Xu et al. (2022), which decomposes the noise on pairwise distances to obtain the modified noise vector $\hat{\epsilon}$ as supervision. We refer readers to Xu et al. (2022) for more details.

Sequence loss $\mathcal{L}^{(\mathcal{S},2\to1)}$. Since we parameterize $p_\theta(\mathcal{S}_1^{t-1}|\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)$ with $\tilde{p}_\theta(\tilde{\mathcal{S}}_1^0|\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)$ and $q(\mathcal{S}_1^{t-1}|\mathcal{S}_1^t, \tilde{\mathcal{S}}_1^0)$ as in (5), it can be proved that the t-th KL divergence term in (8) reaches zero when $\tilde{p}_\theta(\tilde{\mathcal{S}}_1^0|\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)$ puts all mass on the ground truth $\mathcal{S}_1^0$ (see proof in App. B.4). Therefore, for the purpose of pre-training, we can simplify the KL divergence to the cross entropy loss between the correct residue type s_{1i}^0 and the prediction:

$$\mathcal{L}_{\text{simple}}^{(\mathcal{S},2\to1)} = \textstyle\sum_{t=1}^T \sum_i \text{CE}\left(s_{1i}^0, \tilde{p}_\theta(s_{1i}^0|\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)\right). \tag{10}$$

Final pre-training objective. To summarize, the ultimate training objective for our method is

$$\mathcal{L}_{\text{total}} = \frac{1}{2}\left(\mathcal{L}_{\text{simple}}^{(\mathcal{R},2\to1)} + \mathcal{L}_{\text{simple}}^{(\mathcal{S},2\to1)} + \mathcal{L}_{\text{simple}}^{(\mathcal{R},1\to2)} + \mathcal{L}_{\text{simple}}^{(\mathcal{S},1\to2)}\right), \tag{11}$$

where $\mathcal{L}_{\text{simple}}^{(\cdot,b\to a)}$ is the loss term defined by predicting trajectory $\mathcal{P}_a^{0:T}$ from trajectory $\mathcal{P}_b^{0:T}$.

4.4 Discussion

Now we discuss the advantages of our method over previous works. The benefits of these critical designs will be empirically demonstrated by experiments in Sec. 5.3.

Advantages of multimodal denoising. Compared with previous works focusing on either protein sequences (Yang et al., 2022) or structures (Guo et al., 2022), in this work, we perform joint denoising

on both modalities. Note that given a sequence $\mathcal{S}$ and a structure $\mathcal{R}$ that exist in the nature with high probability, the sequence-structure tuple $\mathcal{P} = (\mathcal{S}, \mathcal{R})$ may not be a valid state of this protein. Consequently, instead of modeling the marginal distribution, we are supposed to model the joint distribution of protein sequences and structures, which can be achieved by our multimodal denoising.

Connection with diffusion models. Diffusion models have achieved outstanding performance on image and text generation tasks (Dhariwal & Nichol, 2021; Li et al., 2022) and recently been applied on unsupervised representation learning (Abstreiter et al., 2021). The key to its success is the denoising objective at different noise levels, which has also been used for self-supervised learning in earlier works of scheduled denoising autoencoders (Geras & Sutton, 2014; Chandra & Sharma, 2014). Our method differs from these existing works by incorporating the idea of mutual prediction of two siamese diffusion trajectories so as to maximize the mutual information between them.

5 EXPERIMENTS

To comprehensively evaluate different pre-training algorithms based on protein structures, we select a set of diverse tasks that cover function prediction, structure ranking, mutation effect prediction and local structural role modeling. We conduct experiments on both residue and atom levels to prove the effectiveness of our method on coarse- and fine-grained structures.

5.1 EXPERIMENTAL SETUPS

Pre-training datasets. Following Zhang et al. (2022), we pre-train our models with the AlphaFold protein structure database (Jumper et al., 2021; Varadi et al., 2021), including both 365K proteome-wide predicted structures and 440K Swiss-Prot (Consortium, 2021) predicted structures. In this way, we compare our pre-training algorithms with the ones in Zhang et al. (2022) under a fair setting.

Downstream benchmark tasks. For downstream evaluation, we adopt the EC protein function prediction task (Gligorijević et al., 2021) and four ATOM3D tasks (Townshend et al., 2020).

1. **Enzyme Commission (EC) number prediction** task aims to predict EC numbers of proteins which describe their catalysis behavior in biochemical reactions. This task is formalized as 538 binary classification problems based on the third and fourth levels of EC tree (Webb et al., 1992). We adopt the dataset splits from Gligorijević et al. (2021) and use the test split with 95% sequence identity cutoff following Zhang et al. (2022).
2. **Protein Structure Ranking (PSR)** task predicts global distance test scores of protein structure predictions submitted to the Critical Assessment of Structure Prediction (CASP) (Kryshtafovych et al., 2019) competition. This benchmark dataset is split according to the competition year.
3. **Mutation Stability Prediction (MSP)** task seeks to predict whether a mutation will increase the stability of a protein complex or not (binary classification). The benchmark dataset is split upon a 30% sequence identity cutoff among different splits.
4. **Residue Identity (RES)** task studies the structural role of an amino acid under its local environment. A model predicts the type of the center amino acid based on its surrounding atomic structure. The environments in different splits are with different protein topology classes.

Baseline methods. We evaluate our method on both residue- and atom-level structures. GearNet-Edge (Zhang et al., 2022) and GVP (Jing et al., 2021a) are employed as backbone models for residue and atom levels, respectively. Based on these two encoders, we compare the proposed method with previous protein structural pre-training algorithms including multiview contrastive learning (Zhang et al., 2022), denoising score matching (Guo et al., 2022) and four self-prediction methods (Zhang et al., 2022), *i.e.*, residue type, distance, angle and dihedral prediction.

Training and evaluation. For fair comparison, we pre-train our model for 50 epochs on the AlphaFold protein structure database, following Zhang et al. (2022). For downstream evaluation, we fine-tune the pre-trained models for 200 epochs on EC and 50 epochs on Atom3D tasks except RES. Due to the large size of the RES dataset, we set the time limit as 24 hours and thus only fine-tine each model for 12 epochs. More training hyperparameters are stated in App. D.

For EC prediction, we employ F_{max} and AUPR as evaluation metrics, following the original benchmark (Gligorijević et al., 2021). For PSR prediction, we utilize the global and mean Spearman's ρ to

Table 1: Residue-level results on EC and Atom3D tasks. We use three colors to denote the first, second, third best performance among all models.

	Method	EC		PSR		MSP		Mean Rank
		F_{max}	AUPR	Global ρ	Mean ρ	AUROC	AUPRC	
	GearNet-Edge	0.810	0.835	0.739	0.374	0.586	0.192	-
w/ pre-training	Denoising Score Matching	0.842	0.862	0.823	0.472	0.559	0.183	8.0
	Residue Type Prediction	0.843	0.870	**0.839**	**0.497**	0.642	**0.229**	**4.3**
	Distance Prediction	0.839	0.863	0.780	0.437	0.588	0.168	9.3
	Angle Prediction	0.853	0.880	0.794	0.449	0.632	0.191	6.3
	Dihedral Prediction	**0.859**	**0.881**	0.735	0.366	0.617	0.210	6.3
	Multiview Contrast	**0.874**	**0.892**	0.736	0.362	**0.656**	0.220	4.7
	SiamDiff	**0.864**	**0.882**	**0.829**	**0.506**	**0.695**	**0.331**	**1.5**
	SiamDiff (*w/o* seq.&struct. diff.)	0.802	0.787	0.726	0.338	0.626	0.200	9.7
	SiamDiff (*w/o* seq. diff.)	0.858	0.876	0.789	0.474	0.638	0.202	5.2
	SiamDiff (*w/o* struct. diff.)	0.856	0.873	0.818	**0.481**	**0.660**	**0.246**	**4.0**
	SiamDiff (*w/o* MI max.)	0.855	0.875	**0.826**	0.462	0.616	0.180	6.7

Table 2: Atom-level results on Atom3D tasks. Accuracy is abbreviated as Acc. We use three colors to denote the first, second, third best performance among all models.

	Method	PSR		MSP		RES	Mean Rank
		Global ρ	Mean ρ	AUROC	AUPRC	Acc.	
	GVP	0.809	0.486	0.652	0.228	0.550	-
w/ pre-training	Denoising Score Matching	0.849	0.535	0.625	0.209	0.558	7.2
	Residue Type Prediction	0.845	0.527	0.664	0.254	0.558	5.6
	Distance Prediction	0.825	0.513	0.621	0.250	0.504	8.8
	Angle Prediction	**0.872**	**0.545**	0.659	0.244	**0.581**	**3.2**
	Dihedral Prediction	**0.852**	0.538	**0.692**	**0.260**	**0.585**	**2.4**
	Multiview Contrast	0.848	0.518	0.640	0.187	0.335	9.0
	SiamDiff	**0.854**	**0.548**	**0.694**	**0.263**	**0.563**	**1.6**
	SiamDiff (*w/o* seq.&struct. diff.)	0.796	0.487	0.640	0.216	0.516	9.2
	SiamDiff (*w/o* seq. diff.)	0.850	0.530	**0.670**	**0.255**	0.472	5.2
	SiamDiff (*w/o* struct. diff.)	0.823	0.519	0.642	0.222	0.560	7.0
	SiamDiff (*w/o* MI max.)	0.850	**0.540**	0.631	0.217	0.556	6.2

assess the ranking performance. The MSP task measures the binary classification performance with AUROC and AUPRC, and the micro-averaged accuracy serves as the evaluation metric of RES.

5.2 Experimental Results

Tabs. 1 and 2 report the results of GearNet-Edge and GVP on residue- and atom-level graphs, respectively. We do not include the results of GVP on EC and GearNet-Edge on RES tasks due to their poor performance. We analyze and discuss these results below.

Overall, SiamDiff gains consistently superior or competitive performance on all benchmarks across residue- and atom-level structures. On all residue- and atom-level benchmark tasks, the performance of SiamDiff is among the top three, which is not attained by previous protein structure pre-training methods. Although dihedral prediction performs well on atom-level downstream tasks, its performance on the residue-level PSR and MSP tasks is not satisfactory. The extensive experimental results demonstrate the strength of SiamDiff on boosting diverse types of structure-based protein understanding tasks, including function prediction (EC), protein model quality assessment (PSR), mutation effect prediction (MSP) and structural role understanding (RES).

On EC, SiamDiff outperforms all baselines except Multiview Contrast. Note that this function prediction task aims to predict their catalyzed biochemical reactions, which are mainly determined by the structures of active/binding sites on protein surfaces. The superior performance of Multiview Contrast can be attributed to the design of its correlated views, which aims to capture the statistical relevance of these protein substructures that reflect protein functions and thus aligns representations of proteins with similar functional sites. In contrast, SiamDiff maximizes MI between siamese trajectories for modeling atom- and residue-level interactions underlying protein structural changes.

On PSR, SiamDiff is the best in terms of mean ρ and ranks the second in terms of global ρ. Among all baselines, only Residue Type Prediction can achieve consistently good performance on both structure levels. It is worth noticing that both SiamDiff and residue type prediction can acquire the compatibility of protein sequence and structure. Specifically, Residue Type Prediction models the conditional distribution of sequences given structures, while SiamDiff models the joint distribution of sequences and structures. Such cross-modality modeling mechanisms benefit their performance on this task where the validity of generated structures is assessed given a protein sequence.

On MSP, SiamDiff achieves the best performance. This task classifies a set of similar mutant structures into two groups according to their ability to stabilize protein-protein interactions. Such classification is mostly determined by the local structure change around the mutation site, which can be well modeled by the structure diffusion process in SiamDiff. Essentially, structure diffusion guides the model to recover the original/stable structure from the perturbed/destabilized one, which is highly correlated to the objective of MSP. This can also explain why only applying the structure diffusion objective (SiamDiff *w/o* seq. diff.) still achieves the third best performance at the atom level.

On RES, SiamDiff ranks the third place, lagging behind Angle and Dihedral Prediction. In this task, the types of target residues are predicted given coordinates of its three backbone atoms and atoms from surrounding residues. It is known that dihedral angles on the protein backbone can determine the distribution of plausible side chain structures (Shapovalov & Dunbrack Jr, 2011) and further imply residue types. Therefore, by capturing this information explicitly or implicitly, Dihedral Prediction and Angle Prediction achieve very good results. By comparison, though SiamDiff is not specifically designed for dihedral modeling, it can still achieve competitive performance on this task.

5.3 Ablation Study

For analysis of different components in our method, we report the ablation study results in the last blocks of Tabs. 1 and 2.

Effect of multimodal diffusion. We study three degenerated settings of multimodal diffusion, *i.e.*, *w/o* sequence and structure diffusion, *w/o* sequence diffusion and *w/o* structure diffusion. For the setting without sequence and structure diffusion, we simply perform contrastive learning between two correlated starting states from a protein. Compared to the original method, all these three settings lead to performance decay on all tasks. These results prove the necessity of both structure and sequence diffusion for SiamDiff. These two diffusion schemes jointly capture the process of approaching the protein native state (*i.e.*, the folded structure and the sequence) from scratch.

Effect of MI maximization. We evaluate our method under the setting without MI maximization. Specifically, we only use the original protein to derive the multimodal diffusion trajectory, and multimodal denoising is performed on this single trajectory. This setting is also clearly inferior to the original SiamDiff, since it cannot capture shared information between correlated views. Such a modeling capability is important on many tasks, *e.g.*, general protein function prediction on EC (F_{max} score before and after removing this component: 0.864 *v.s.* 0.855), and it is thus an essential component of SiamDiff. More ablation studies are provided in App. C.1.

6 Conclusions and Future Work

In this work, we propose the Siamese Diffusion Trajectory Prediction (SiamDiff) for protein structure pre-training. SiamDiff deems the whole diffusion trajectory of a protein as a view and seeks to maximize the mutual information between correlated trajectories. In particular, we consider the multimodal diffusion process in which both the protein folded structure and the protein sequence are gradually corrupted towards chaos, so as to capture the joint distribution of structures and sequences. SiamDiff uses a principled theoretical framework to perform multimodal denoising across views. Extensive experiments on diverse types of tasks and on both residue- and atom-level structures verify the consistently superior or competitive performance of SiamDiff against previous baselines.

In future works, we will enhance SiamDiff with a dedicated multimodal encoder that can better model the protein structure and sequence in a joint fashion, and we will also explore how to incorporate the large-scale protein sequence corpus into the pre-training process.

REPRODUCIBILITY STATEMENT

For the sake of reproducibility, in App. D, we provide detailed data processing schemes, graph construction schemes, model configurations, pre-training configurations and fine-tuning configurations. All source code of this work will be released to public upon acceptance.

REFERENCES

Korbinian Abstreiter, Stefan Bauer, Bernhard Schölkopf, and Arash Mehrjou. Diffusion-based representation learning. *arXiv preprint arXiv:2105.14257*, 2021.

Kazuyuki Akasaka. Probing conformational fluctuation of proteins by pressure perturbation. *Chemical Reviews*, 106(5):1814–1835, 2006.

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

Christian B Anfinsen. The formation and stabilization of protein structure. *Biochemical Journal*, 128 (4):737, 1972.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

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.

Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):1–11, 2022.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pp. 531–540. PMLR, 2018.

B Chandra and Rajesh Kumar Sharma. Adaptive noise schedule for denoising autoencoder. In *International conference on neural information processing*, pp. 535–542. Springer, 2014.

Can Chen, Jingbo Zhou, Fan Wang, Xue Liu, and Dejing Dou. Structure-aware protein self-supervised learning. *arXiv preprint arXiv:2204.04213*, 2022.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020b.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 2021.

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

Ken A Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *science*, 338(6110): 1042–1046, 2012.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time. iv. *Communications on Pure and Applied Mathematics*, 36(2):183–212, 1983.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Wang Yu, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/TPAMI.2021.3095381.

Bent Fuglede and Flemming Topsoe. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, pp. 31. IEEE, 2004.

Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

Krzysztof J Geras and Charles Sutton. Scheduled denoising autoencoders. *arXiv preprint arXiv:1406.3269*, 2014.

Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12 (1):1–14, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Yuzhi Guo, Jiaxiang Wu, Hehuan Ma, and Junzhou Huang. Self-supervised pre-training for protein embeddings using tertiary structures. In *AAAI*, 2022.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2), 2012.

Michael J Harms and Joseph W Thornton. Analyzing protein structure and function using ancestral gene reconstruction. *Current opinion in structural biology*, 20(3):360–366, 2010.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pp. 4116–4126. PMLR, 2020.

Pedro Hermosilla and Timo Ropinski. Contrastive representation learning for 3d protein structures. *arXiv preprint arXiv:2205.15675*, 2022.

Pedro Hermosilla, Marco Schäfer, Matěj Lang, Gloria Fackelmann, Pere Pau Vázquez, Barbora Kozlíková, Michael Krone, Tobias Ritschel, and Timo Ropinski. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. *International Conference on Learning Representations*, 2021.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

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

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

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

Bowen Jing, Stephan Eismann, Pratham N. Soni, and Ron O. Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=1YLJDvSx6J4.

Bowen Jing, Stephan Eismann, Pratham N Soni, and Ron O Dror. Equivariant graph neural networks for 3d macromolecular structure. *arXiv preprint arXiv:2106.03843*, 2021b.

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.

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. In *International conference on machine learning*, pp. 5361–5370. PMLR, 2020.

Andriy Kryshtafovych, Torsten Schwede, Maya Topf, Krzysztof Fidelis, and John Moult. Critical assessment of methods of protein structure prediction (casp)—round xiii. *Proteins: Structure, Function, and Bioinformatics*, 87(12):1011–1020, 2019.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022.

Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*, 2021.

Shengchao Liu, Hongyu Guo, and Jian Tang. Molecular geometry pretraining with se (3)-invariant denoising distance matching. *arXiv preprint arXiv:2206.13602*, 2022.

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*, 2022.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171, 2021.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

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

Maxim V Shapovalov and Roland L Dunbrack Jr. A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions. *Structure*, 19(6): 844–858, 2011.

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

George N Somero. Proteins and temperature. *Annual review of physiology*, 57(1):43–68, 1995.

Vignesh Ram Somnath, Charlotte Bunne, and Andreas Krause. Multi-scale representation learning on proteins. *Advances in Neural Information Processing Systems*, 34, 2021.

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.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428, 2021.

Freyr Sverrisson, Jean Feydy, Bruno E Correia, and Michael M Bronstein. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15272–15281, 2021.

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.

Raphael JL Townshend, Martin Vögele, Patricia Suriana, Alexander Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann, et al. Atom3d: Tasks on molecules in three dimensions. *arXiv preprint arXiv:2012.04035*, 2020.

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.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 2021.

Limei Wang, Haoran Liu, Yi Liu, Jerry Kurtin, and Shuiwang Ji. Learning protein representations via complete 3d graph networks. *arXiv preprint arXiv:2207.12600*, 2022a.

Zichen Wang, Steven A Combs, Ryan Brand, Miguel Romero Calvo, Panpan Xu, George Price, Nataliya Golovach, Emmanuel O Salawu, Colby J Wise, Sri Priya Ponnapalli, et al. Lm-gvp: an extensible sequence and structure informed deep learning framework for protein property prediction. *Scientific reports*, 12(1):1–12, 2022b.

Oren F Webb, Tommy J Phelps, Paul R Bienkowski, Philip M Digrazia, David C White, and Gary S Sayler. Enzyme nomenclature. 1992.

Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018.

Fang Wu, Qiang Zhang, Dragomir Radev, Yuyang Wang, Xurui Jin, Yinghui Jiang, Zhangming Niu, and Stan Z Li. Pre-training of deep protein models with molecular dynamics simulations for drug binding. *arXiv preprint arXiv:2204.08663*, 2022a.

Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *arXiv preprint arXiv:2209.00865*, 2022b.

Minghao Xu, Hang Wang, Bingbing Ni, Hongyu Guo, and Jian Tang. Self-supervised graph-level representation learning with local and global structure. In *International Conference on Machine Learning*, pp. 11548–11558. PMLR, 2021.

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.

Kevin K Yang, Niccolò Zanichelli, and Hugh Yeh. Masked inverse folding with sequence transfer for protein representation learning. *bioRxiv*, 2022.

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, et al. Torchdrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

A MORE RELATED WORKS

Mutual Information (MI) Estimation and Maximization. MI can measure both the linear and non-linear dependency between random variables. Some previous works (Belghazi et al., 2018; Hjelm et al., 2018) try to use neural networks to estimate the lower bound of MI, including Donsker-Varadhan representation (Donsker & Varadhan, 1983), Jensen-Shannon divergence (Fuglede & Topsoe, 2004) and Noise-Contrastive Estimation (NCE) (Gutmann & Hyvärinen, 2010; 2012). The optimization with InfoNCE loss (Oord et al., 2018) maximizes a lower bound of MI and is broadly shown to be a superior representation learning strategy (Chen et al., 2020b; Hassani & Khasahmadi, 2020; Xu et al., 2021; Liu et al., 2021; Zhang et al., 2022). In this work, we adopt the MI lower bound proposed by Liu et al. (2022) with two conditional log-likelihoods, and we formulate the learning objective by mutually denoising the multimodal diffusion processes of two correlated proteins.

B PROOFS

B.1 PROOF OF PROPOSITION 1

Proof. First, the mutual information between representations of two trajectories is defined as:

$$I(\boldsymbol{P}_1^{0:T};\boldsymbol{P}_2^{0:T}) = \mathbb{E}_{\boldsymbol{\mathcal{P}}_1^{0:T},\boldsymbol{\mathcal{P}}_2^{0:T}\sim p(\boldsymbol{P}_1^{0:T},\boldsymbol{P}_2^{0:T})}\left[\log \frac{p(\boldsymbol{\mathcal{P}}_1^{0:T},\boldsymbol{\mathcal{P}}_2^{0:T})}{p(\boldsymbol{\mathcal{P}}_1^{0:T})p(\boldsymbol{\mathcal{P}}_2^{0:T})}\right], \tag{12}$$

where the joint distribution is defined as $p(\boldsymbol{P}_1^{0:T},\boldsymbol{P}_2^{0:T}) = p(\boldsymbol{P}_1^0,\boldsymbol{P}_2^0)q(\boldsymbol{P}_1^{1:T}|\boldsymbol{P}_1^0)q(\boldsymbol{P}_2^{1:T}|\boldsymbol{P}_2^0)$. Next, we can derive a lower bound with this definition:

$$\begin{aligned}
&I(\boldsymbol{P}_1^{0:T};\boldsymbol{P}_2^{0:T})\\
=&\mathbb{E}\left[\log \frac{p(\boldsymbol{\mathcal{P}}_1^{0:T},\boldsymbol{\mathcal{P}}_2^{0:T})}{p(\boldsymbol{\mathcal{P}}_1^0)q(\boldsymbol{\mathcal{P}}_1^{1:T}|\boldsymbol{\mathcal{P}}_1^0)p(\boldsymbol{\mathcal{P}}_2^0)q(\boldsymbol{\mathcal{P}}_2^{1:T}|\boldsymbol{\mathcal{P}}_2^0)}\right]\\
\geq&\mathbb{E}\left[\log \frac{p(\boldsymbol{\mathcal{P}}_1^{0:T},\boldsymbol{\mathcal{P}}_2^{0:T})}{\sqrt{p(\boldsymbol{\mathcal{P}}_1^0)p(\boldsymbol{\mathcal{P}}_2^0)}q(\boldsymbol{\mathcal{P}}_1^{1:T}|\boldsymbol{\mathcal{P}}_1^0)q(\boldsymbol{\mathcal{P}}_2^{1:T}|\boldsymbol{\mathcal{P}}_2^0)}\right]\\
=&\frac{1}{2}\mathbb{E}\left[\log \frac{p(\boldsymbol{\mathcal{P}}_1^{0:T},\boldsymbol{\mathcal{P}}_2^{0:T})^2}{p(\boldsymbol{\mathcal{P}}_1^0)p(\boldsymbol{\mathcal{P}}_2^0)q(\boldsymbol{\mathcal{P}}_1^{1:T}|\boldsymbol{\mathcal{P}}_1^0)^2q(\boldsymbol{\mathcal{P}}_2^{1:T}|\boldsymbol{\mathcal{P}}_2^0)^2}\right]\\
=&\frac{1}{2}\mathbb{E}\left[\log \frac{p(\boldsymbol{\mathcal{P}}_1^{0:T},\boldsymbol{\mathcal{P}}_2^{0:T})}{p(\boldsymbol{\mathcal{P}}_2^0)q(\boldsymbol{\mathcal{P}}_1^{1:T}|\boldsymbol{\mathcal{P}}_1^0)q(\boldsymbol{\mathcal{P}}_2^{1:T}|\boldsymbol{\mathcal{P}}_2^0)} + \log \frac{p(\boldsymbol{\mathcal{P}}_1^{0:T},\boldsymbol{\mathcal{P}}_2^{0:T})}{p(\boldsymbol{\mathcal{P}}_1^0)q(\boldsymbol{\mathcal{P}}_1^{1:T}|\boldsymbol{\mathcal{P}}_1^0)q(\boldsymbol{\mathcal{P}}_2^{1:T}|\boldsymbol{\mathcal{P}}_2^0)}\right]\\
=&\frac{1}{2}\mathbb{E}\left[\log \frac{p(\boldsymbol{\mathcal{P}}_1^{0:T}|\boldsymbol{\mathcal{P}}_2^{0:T})}{q(\boldsymbol{\mathcal{P}}_1^{1:T}|\boldsymbol{\mathcal{P}}_1^0)} + \log \frac{p(\boldsymbol{\mathcal{P}}_2^{0:T}|\boldsymbol{\mathcal{P}}_1^{0:T})}{q(\boldsymbol{\mathcal{P}}_2^{1:T}|\boldsymbol{\mathcal{P}}_2^0)}\right].
\end{aligned}$$

However, since the distribution of representations are intractable to sample for optimization, we instead sample the trajectories $\mathcal{P}_1^{0:T}$ and $\mathcal{P}_2^{0:T}$ from our defined diffusion process, *i.e.*, $p(\mathcal{P}_1^{0:T},\mathcal{P}_2^{0:T}) = p(\mathcal{P}_1^0,\mathcal{P}_2^0)q(\mathcal{P}_1^{1:T}|\mathcal{P}_1^0)q(\mathcal{P}_2^{1:T}|\mathcal{P}_2^0)$. Besides, instead of predicting representations, we use the representations from one trajectory to recover the other trajectory, which reflects more information than its representation. With these approximations, the lower bound above can be further written as:

$$\frac{1}{2}\mathbb{E}\left[\log \frac{p(\boldsymbol{\mathcal{P}}_1^{0:T}|\boldsymbol{\mathcal{P}}_2^{0:T})}{q(\boldsymbol{\mathcal{P}}_1^{1:T}|\boldsymbol{\mathcal{P}}_1^0)} + \log \frac{p(\boldsymbol{\mathcal{P}}_2^{0:T}|\boldsymbol{\mathcal{P}}_1^{0:T})}{q(\boldsymbol{\mathcal{P}}_2^{1:T}|\boldsymbol{\mathcal{P}}_2^0)}\right] \approx \frac{1}{2}\mathbb{E}_{\mathcal{P}_1^{0:T},\mathcal{P}_2^{0:T}}\left[\log \frac{p(\mathcal{P}_1^{0:T}|\boldsymbol{\mathcal{P}}_2^{0:T})}{q(\mathcal{P}_1^{1:T}|\mathcal{P}_1^0)} + \log \frac{p(\mathcal{P}_2^{0:T}|\boldsymbol{\mathcal{P}}_1^{0:T})}{q(\mathcal{P}_2^{1:T}|\mathcal{P}_2^0)}\right]$$

We now show the first term on the right hand side can be written as the loss defined in (6). The derivation is very similar with the proof of Proposition 3 in Xu et al. (2022). We include it here for

completeness:

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{P}_1^{0:T},\mathcal{P}_2^{0:T}}\left[\log\frac{p(\mathcal{P}_1^{0:T}|\boldsymbol{\mathcal{P}}_2^{0:T})}{q(\mathcal{P}_1^{1:T}|\mathcal{P}_1^0)}\right]\\
=&\mathbb{E}_{\mathcal{P}_1^{0:T},\mathcal{P}_2^{0:T}}\left[\sum_{t=1}^{T}\log\frac{p(\mathcal{P}_1^{t-1}|\mathcal{P}_1^t,\boldsymbol{\mathcal{P}}_2^{0:T})}{q(\mathcal{P}_1^t|\mathcal{P}_1^{t-1})}\right]\\
=&\mathbb{E}_{\mathcal{P}_1^{0:T},\mathcal{P}_2^{0:T}}\left[\log\frac{(\mathcal{P}_1^0|\mathcal{P}_1^1,\boldsymbol{\mathcal{P}}_2^{0:T})}{q(\mathcal{P}_1^1|\mathcal{P}_1^0)}+\sum_{t=2}^{T}\log\left(\frac{p(\mathcal{P}_1^{t-1}|\mathcal{P}_1^t,\boldsymbol{\mathcal{P}}_2^{0:T})}{q(\mathcal{P}_1^{t-1}|\mathcal{P}_1^t,\mathcal{P}_1^0)}\cdot\frac{q(\mathcal{P}_1^{t-1}|\mathcal{P}_1^0)}{q(\mathcal{P}_1^t|\mathcal{P}_1^0)}\right)\right]\\
=&\mathbb{E}_{\mathcal{P}_1^{0:T},\mathcal{P}_2^{0:T}}\left[-\log q(\mathcal{P}_1^T|\mathcal{P}_1^0)+\log p(\mathcal{P}_1^0|\mathcal{P}_1^1,\boldsymbol{\mathcal{P}}_2^{0:T})+\sum_{t=2}^{T}\log\frac{p(\mathcal{P}_1^{t-1}|\mathcal{P}_1^t,\boldsymbol{\mathcal{P}}_2^{0:T})}{q(\mathcal{P}_1^{t-1}|\mathcal{P}_1^t,\mathcal{P}_1^0)}\right]\\
=&-\mathbb{E}_{\mathcal{P}_1^{0:T},\mathcal{P}_2^{0:T}}\left[\sum\nolimits_{t=1}^{T}D_{\mathrm{KL}}\left(q(\mathcal{P}_1^{t-1}|\mathcal{P}_1^t,\mathcal{P}_1^0)||p(\mathcal{P}_1^{t-1}|\mathcal{P}_1^t,\boldsymbol{\mathcal{P}}_2^{0:T})\right)\right]+C^{(2\to1)}\\
=&-\mathcal{L}^{(2\to1)}+C^{(2\to1)},
\end{aligned}
$$

where we merge the term $p(\mathcal{P}_1^0|\mathcal{P}_1^1,\boldsymbol{\mathcal{P}}_2^{0:T})$ into the sum of KL divergences for brevity and use $C^{(2\to1)}$ to denote the constant independent of our encoder. Note that the counterpart can be derived in the same way. Adding these two terms together finishes the proof of Proposition 1. □

B.2 Proof of Pre-Training Loss Decomposition

We restate the proposition of pre-training loss decomposition rigorously as below.

Proposition 2 *Given the assumptions 1) the separation of the diffusion process on protein structures and sequences*

$$q(\mathcal{P}_a^t|\mathcal{P}_a^{t-1})=q(\mathcal{R}_a^t|\mathcal{R}_a^{t-1})\cdot q(\mathcal{S}_a^t|\mathcal{S}_a^{t-1}), \tag{13}$$

and 2) the conditional independence of the generation process

$$p_\theta(\mathcal{P}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^t)=p_\theta(\mathcal{R}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^t)\cdot p_\theta(\mathcal{S}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^t), \tag{14}$$

it can be proved that

$$\mathcal{L}^{(b\to a)}=\mathcal{L}^{(\mathcal{R},b\to a)}+\mathcal{L}^{(\mathcal{S},b\to a)}, \tag{15}$$

where the three loss terms are defined as

$$
\begin{aligned}
\mathcal{L}^{(b\to a)}:=&\mathbb{E}\left[\sum\nolimits_{t=1}^{T}D_{KL}\left(q(\mathcal{P}_a^{t-1}|\mathcal{P}_a^t,\mathcal{P}_a^0)||p_\theta(\mathcal{P}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^t)\right)\right],\\
\mathcal{L}^{(\mathcal{R},b\to a)}:=&\mathbb{E}\left[\sum\nolimits_{t=1}^{T}D_{KL}\left(q(\mathcal{R}_a^{t-1}|\mathcal{R}_a^t,\mathcal{R}_a^0)||p_\theta(\mathcal{R}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^t)\right)\right],\\
\mathcal{L}^{(\mathcal{S},b\to a)}:=&\mathbb{E}\left[\sum\nolimits_{t=1}^{T}D_{KL}\left(q(\mathcal{S}_a^{t-1}|\mathcal{S}_a^t,\mathcal{S}_a^0)||p_\theta(\mathcal{S}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^t)\right)\right],
\end{aligned}
$$

with $b\to a$ referring to the term from trajectory $\mathcal{P}_b^{0:T}$ to $\mathcal{P}_a^{0:T}$.

Proof. Let $\mathcal{L}_t^{(\cdot)}$ to denote the t-th KL divergence term in $\mathcal{L}^{(\cdot)}$. Then, we have

$$
\begin{aligned}
&\mathcal{L}_t^{(b\to a)}=D_{\mathrm{KL}}\left(q(\mathcal{P}_a^{t-1}|\mathcal{P}_a^t,\mathcal{P}_a^0)||p_\theta(\mathcal{P}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^{0:T})\right)\\
&=D_{\mathrm{KL}}\left(\left[q(\mathcal{R}_a^{t-1}|\mathcal{R}_a^t,\mathcal{R}_a^0)q(\mathcal{S}_a^{t-1}|\mathcal{S}_a^t,\mathcal{S}_a^0)\right]||\left[p_\theta(\mathcal{R}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^{0:T})p_\theta(\mathcal{S}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^{0:T})\right]\right)\\
&=D_{\mathrm{KL}}\left(q(\mathcal{R}_a^{t-1}|\mathcal{R}_a^t,\mathcal{R}_a^0)||p_\theta(\mathcal{R}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^{0:T})\right)+D_{\mathrm{KL}}\left(q(\mathcal{S}_a^{t-1}|\mathcal{S}_a^t,\mathcal{S}_a^0)||p_\theta(\mathcal{S}_a^{t-1}|\mathcal{P}_a^t,\boldsymbol{\mathcal{P}}_b^{0:T})\right)\\
&=\mathcal{L}_t^{(\mathcal{R},b\to a)}+\mathcal{L}_t^{(\mathcal{S},b\to a)},
\end{aligned}
$$

where we use the assumptions (13) and (14) in the second equality. The third equality is due to the additive property of the KL divergence for independent distributions. Adding T KL divergence terms together will lead to (15). □

B.3 PROOF OF SIMPLIFIED STRUCTURE LOSS

For completeness, we show how to derive the simplified structure loss. The proof is directly adapted from (Xu et al., 2022).

Proposition 3 *Given the definition of the forward process*

$$q(\mathcal{R}_a^t|\mathcal{R}_a^{t-1}) = \mathcal{N}(\mathcal{R}_a^t; \sqrt{1-\beta_t}\mathcal{R}_a^{t-1}, \beta_t I), \tag{16}$$

and the reverse process

$$p_\theta(\mathcal{R}_a^{t-1}|\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t) = \mathcal{N}(\mathcal{R}^{t-1}; \mu_\theta(\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t, t), \sigma_t^2 I), \tag{17}$$

$$\mu_\theta(\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathcal{R}_a^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t, t)\right), \tag{18}$$

the structure loss function

$$\mathcal{L}^{(\mathcal{R},b\to a)} := \mathbb{E}\left[\sum_{t=1}^T D_{KL}\left(q(\mathcal{R}_a^{t-1}|\mathcal{R}_a^t, \mathcal{R}_a^0)||p_\theta(\mathcal{R}_a^{t-1}|\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t)\right)\right], \tag{19}$$

can be simplified to

$$\mathcal{L}^{(\mathcal{R},b\to a)} = \sum_{t=1}^T \gamma_t \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\|\epsilon - \epsilon_\theta(\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t, t)\|_2^2\right], \tag{20}$$

where $\gamma_t = \frac{\beta_t}{2\alpha_t(1-\bar{\alpha}_{t-1})}$ *with* $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ *and* $b \to a$ *is either* $2 \to 1$ *or* $1 \to 2$.

Proof. First, we prove $q(\mathcal{R}_a^t|\mathcal{R}_a^0) = \mathcal{N}(\mathcal{R}_a^t; \sqrt{\bar{\alpha}_t}\mathcal{R}_a^0, (1-\bar{\alpha}_t)I)$. Let ϵ_i be the standard Gaussian random variable at time step i. Then, we have

$$\begin{aligned}
\mathcal{R}_a^t &= \sqrt{\alpha_t}\mathcal{R}_a^{t-1} + \sqrt{\beta_t}\epsilon_t \\
&= \sqrt{\alpha_{t-1}\alpha_t}\mathcal{R}_a^{t-2} + \sqrt{\alpha_{t-1}\beta_{t-1}}\epsilon_{t-1} + \sqrt{\beta_t}\epsilon_t \\
&= \cdots \\
&= \sqrt{\bar{\alpha}_t}\mathcal{R}_a^0 + \sqrt{\alpha_t\alpha_{t-1}...\alpha_2\beta_1}\epsilon_1 + \cdots + \sqrt{\alpha_{t-1}\beta_{t-1}}\epsilon_{t-1} + \sqrt{\beta_t}\epsilon_t,
\end{aligned}$$

which suggests that the mean of $\mathcal{R}_a^t$ is $\sqrt{\bar{\alpha}_t}\mathcal{R}_a^0$ and the variance matrix is $(\alpha_t\alpha_{t-1}...\alpha_2\beta_1 + \cdots + \alpha_{t-1}\beta_{t-1} + \beta_t)I = (1-\bar{\alpha})I$.

Next, we derive the posterior distribution as:

$$\begin{aligned}
q(\mathcal{R}_a^{t-1}|\mathcal{R}_a^t, \mathcal{R}_a^0) &= \frac{q(\mathcal{R}_a^t|\mathcal{R}_a^{t-1})q(\mathcal{R}_a^{t-1}|\mathcal{R}_a^0)}{q(\mathcal{R}_a^t|\mathcal{R}_a^0)} \\
&= \frac{\mathcal{N}(\mathcal{R}_a^t; \sqrt{\alpha_t}\mathcal{R}_a^{t-1}, \beta_t I)\cdot\mathcal{N}(\mathcal{R}_a^{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathcal{R}_a^0, (1-\bar{\alpha}_{t-1})I)}{\mathcal{N}(\mathcal{R}_a^t; \sqrt{\bar{\alpha}_t}\mathcal{R}_a^0, (1-\bar{\alpha}_t)I)} \\
&= \mathcal{N}(\mathcal{R}^{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathcal{R}_a^0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathcal{R}_a^t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t I).
\end{aligned}$$

Let $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$, then the t-th KL divergence term can be written as:

$$\begin{aligned}
&D_{\mathrm{KL}}\left(q(\mathcal{R}_a^{t-1}|\mathcal{R}_a^t, \mathcal{R}_a^0)||p_\theta(\mathcal{R}_a^{t-1}|\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t)\right) \\
=&\frac{1}{2\tilde{\beta}_t}\left\|\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathcal{R}_a^0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathcal{R}_a^t - \frac{1}{\sqrt{\alpha_t}}\left(\mathcal{R}_a^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t, t)\right)\right\|^2 \\
=&\frac{1}{2\tilde{\beta}_t}\mathbb{E}_\epsilon\left\|\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\cdot\frac{\mathcal{R}_a^t - \sqrt{1-\bar{\alpha}_t}\epsilon}{\sqrt{\bar{\alpha}_t}} + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathcal{R}_a^t - \frac{1}{\sqrt{\alpha_t}}\left(\mathcal{R}_a^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t, t)\right)\right\|^2 \\
=&\frac{1}{2\tilde{\beta}_t}\cdot\frac{\beta_t^2}{\alpha_t(1-\bar{\alpha}_t)}\mathbb{E}_\epsilon\left\|\epsilon - \epsilon_\theta(\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t, t)\right\| \\
=&\gamma_t\mathbb{E}_\epsilon\left[\|\epsilon - \epsilon_\theta(\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t, t)\|_2^2\right],
\end{aligned}$$

which completes the proof. □

B.4 PROOF OF SIMPLIFIED SEQUENCE LOSS

Now we show the equivalence of optimizing sequence loss $\mathcal{L}^{(\mathcal{S},b\to a)}$ and the masked residue type prediction problem on $\mathcal{S}_a^0$.

Proposition 4 *Given the definition of reverse process on protein sequences*

$$p_\theta(\mathcal{S}_a^{t-1}|\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t) \propto \textstyle\sum_{\tilde{\mathcal{S}}_a^0} q(\mathcal{S}_a^{t-1}|\mathcal{S}_a^t, \tilde{\mathcal{S}}_a^0) \cdot \tilde{p}_\theta(\tilde{\mathcal{S}}_a^0|\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t), \tag{21}$$

the sequence loss $\mathcal{L}^{(\mathcal{S},b\to a)}$ reaches zero when $\tilde{p}_\theta(\tilde{\mathcal{S}}_a^0|\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t)$ puts all mass on the ground truth $\mathcal{S}_a^0$.

Proof. The loss function can be written as:

$$\begin{aligned}\mathcal{L}^{(\mathcal{S},b\to a)} &:= \mathbb{E}\left[\textstyle\sum_{t=1}^T D_{\mathrm{KL}}\left(q(\mathcal{S}_a^{t-1}|\mathcal{S}_a^t, \mathcal{S}_a^0)||p_\theta(\mathcal{S}_a^{t-1}|\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t)\right)\right] \\ &= \mathbb{E}\left[\sum_{t=1}^T D_{\mathrm{KL}}\left(q(\mathcal{S}_a^{t-1}|\mathcal{S}_a^t, \mathcal{S}_a^0)\middle|\middle|\frac{\sum_{\tilde{\mathcal{S}}_a^0} q(\mathcal{S}_a^{t-1}|\mathcal{S}_a^t, \tilde{\mathcal{S}}_a^0) \cdot \tilde{p}_\theta(\tilde{\mathcal{S}}_a^0|\mathcal{P}_1^t, \boldsymbol{\mathcal{P}}_2^t)}{Z}\right)\right],\end{aligned}$$

where Z is the normalization constant. Hence, when $\tilde{p}_\theta(\tilde{\mathcal{S}}_a^0|\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t)$ puts all mass on the ground truth $\mathcal{S}_a^0$, the distribution $p_\theta(\mathcal{S}_a^{t-1}|\mathcal{P}_a^t, \boldsymbol{\mathcal{P}}_b^t)$ will be identical with $q(\mathcal{S}_a^{t-1}|\mathcal{S}_a^t, \mathcal{S}_a^0)$, which makes the KL divergence become zero. □

C MORE EXPERIMENTAL RESULTS

C.1 ABLATION STUDY

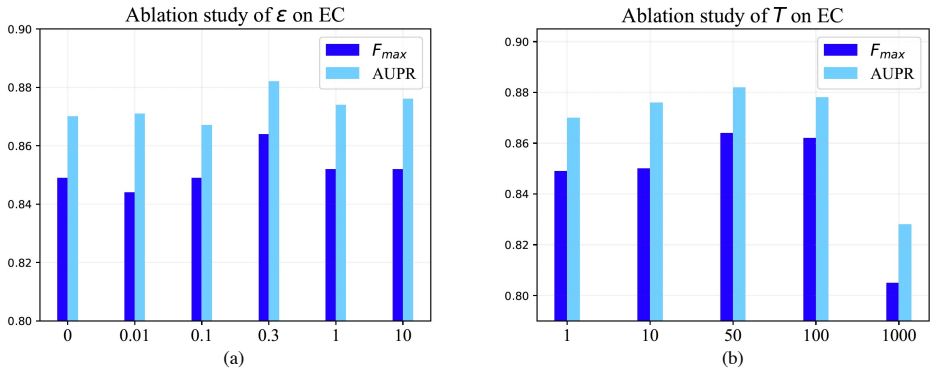

Figure 2: Ablation study of (a) structure perturbation scale ϵ and (b) time step number T on EC.

Effect of structure perturbation scale ϵ. In Fig. 2(a), we present the performance of SiamDiff on EC under different structure perturbation scales. We can observe the superiority of $\epsilon = 0.3$ over other settings on both evaluation metrics. This result illustrates that moderately large structure perturbations favor SiamDiff's effectiveness. Under such condition, the correlated views can distinguish from each other while share sufficient common information, leading to the moderate difficulty of the SiamDiff pre-training task.

Effect of time step number T. Fig. 2(b) shows SiamDiff's performance on EC by using different time step numbers. It can be observed that too few time steps (*i.e.*, 1 or 10 time steps) and too many time steps (*i.e.*, 1000 steps) both lead to the inferior performance. In these situations, the task of each denoising step is either too hard (with too few time steps) or too simple (with too many time steps). Therefore, it is suggested to use moderately many time steps (*e.g.*, 50 or 100 time steps) in the SiamDiff pre-training method.

D EXPERIMENTAL DETAILS

In this section, we introduce the details of our experiments. All these methods are developed based on PyTorch and TorchDrug (Zhu et al., 2022).

Graph construction. For atom graphs, we connect atoms with Euclidean distance lower than a distance threshold. For PSR and MSP tasks, we remove all hydrogen atoms following Jing et al. (2021b). For residue graphs, we discard all non-alpha-carbon atoms and add three different types of directed edges: sequential edges, radius edges and K-nearest neighbor edges. For sequential edges, two atoms are connected if their sequential distance is below a threshold and these edges are divided into different types according to these distances. For two kinds of spatial edges, we connect atoms according to Euclidean distance and k-nearest neighbors. We further apply a long range interaction filter that removes edges with low sequential distances.

Backbone models. We directly borrow the best hyperparameters reported in the original paper of GearNet-Edge (Zhang et al., 2022). We adopt the same configuration of relational graph construction, *i.e.*, the sequential distance threshold $d_{\text{seq}} = 3$, the radius $d_{\text{radius}} = 10.0$Å, the number of neighbors $k = 10$ and the long range interaction cutoff $d_{\text{long}} = 5$. We use one-hot features of residue types as node features and concatenate (1) one-hot features of end nodes, (2) one-hot features of edge types, (3) sequential distance, (4) spatial distance as edge features. Then we use 6 message passing layers with 512 hidden dimensions and ReLU as the activation function. For edge message passing, the edge types on the line graph are determined by the discretized angles. The hidden representations in each layer of GearNet will be concatenated for the final prediction. Since only alpha carbon atoms are kept in the graph, their representations are used for both atom and residue representations.

For GVP, the original design only includes atom types as node features, which makes pre-training tasks with residue type prediction very difficult to learn. To address this issue, we slightly modify its architecture to add add the embedding of atom and corresponding residue types as atom features. Then, the default configurations in Jing et al. (2021b) are adopted. We construct an atom graph for each protein by drawing edges between atoms closer than 4.5Å. Each edge is featured with a 16-dimensional Gaussian RBF encoding of its Euclidean distance. We use five GVP layers and hidden representations with 16 vector and 100 scalar channels and use ReLU and identity for scalar and vector activation functions, respectively. The dropout rate is set as 0.1. The final atom representations are followed by a mean pooling layer to obtain residue and protein representations.

Pre-training methods. For all baselines in (Zhang et al., 2022), we adopt the original configurations. For Multiview Contrast, we use subsequence cropping that randomly extracts protein subsequences with no more than 50 residues and space cropping that takes all residues within a 15Å Euclidean ball with a random center residue. Then, either an identity function or a random edge masking function with mask rate equal to 0.15 is applied for constructing views. The temperature τ in the InfoNCE loss function is set as 0.07. We set the number of sampled items in each protein as 256 for Distance Prediction and as 512 for Angle and Dihedral Prediction. The mask rate for Residue Type Prediction is set as 0.15. When masking a residue on atom graphs, we discard all non-backbone atoms and set the residue features as zero. Since the backbone models and tasks in our paper are quite different with those in Guo et al. (2022), we re-implement the method on our codebase. We consider 50 different noise levels log-linearly ranging from 0.01 to 10.0.

For our method, we set the variance of structure perturbation noises ϵ as 0.3 when constructing the other view. For structure diffusion, we use a sigmoid schedule for variances β_t with the lowest variance $\beta_1 = 1e-3$ and the highest variance $\beta_T = 0.1$. For sequence diffusion, we simply set the cumulative transition probability to [MASK] over time steps as a linear interpolation between minimum mask rate 0.15 and maximum mask rate 1.0. The number of diffusion steps is set as 1000.

All other optimization configurations for these pre-training methods are reported in Tab. 3. All methods are pre-trained on four Tesla A100 GPUs and Tab. 3 reports the batch sizes on each GPU.

Fine-tuning on downstream tasks. For all models on all downstream tasks, we apply the a three-layer MLP head for prediction, the hidden dimension of which is set to the dimension of model outputs. The batch sizes for each model are chosen according the memory limit. For all residue-level tasks, we fine-tune the GearNet-Edge on 4 GPUs (A100 for EC/GO and V100 for PSR/MSP) with the batch size on a single one as 2. For all atom-level tasks except RES, we set the batch size as 2 and fine-tune the model on a single V100 GPU. For RES, we fine-tune the pre-trained GVP on 4 A100 GPUs with the batch size on a single one as 32.

Table 3: Optimization configurations for pre-training methods. Here max length denotes the maximum number of residues kept in each protein and lr stands for learning rate.

Method	**Max length**		**Batch size**		**Optimizer**	**lr**
	residue	atom	residue	atom		
Residue Type Prediction	100	100	96	64	Adam	1e-3
Distance Prediction	100	100	128	64	Adam	1e-3
Angle Prediction	100	100	96	64	Adam	1e-3
Dihedral Prediction	100	100	96	64	Adam	1e-3
Multiview Contrast	-	-	96	64	Adam	1e-3
Denoising Score Matching	200	200	12	12	Adam	1e-4
SiamDiff	150	100	32	32	Adam	1e-4