A PROOF

A.1 PROOF OF EQUATION 22

Let us rewrite $\nabla_{\mathcal{D}_t} \log p_t(\mathcal{D}_t|\mathcal{R}_{\text{atm}}, \mathcal{G})$ and $\nabla_{\mathcal{D}_t} \log p_t(\mathcal{D}_t|\mathcal{R}_{\text{atm}}, \mathcal{R}_{\text{aux}}, \mathcal{G})$ using Baye's rule:

$$\begin{aligned}
\nabla_{\mathcal{D}_t} \log p_{\mathcal{G}}(\mathcal{D}|\mathcal{R}_{\text{atm}}) &= \nabla_{\mathcal{D}_t} \log p_{\mathcal{G}}(\mathcal{R}_{\text{atm}}|\mathcal{D}_t) + \nabla_{\mathcal{D}_t} \log p_{\mathcal{G}}(\mathcal{D}_t) \\
\nabla_{\mathcal{D}_t} \log p_{\mathcal{G}}(\mathcal{D}_t|\{\mathcal{R}_{\text{atm}}, \mathcal{R}_{\text{aux}}\}) &= \nabla_{\mathcal{D}_t} \log p_{\mathcal{G}}(\{\mathcal{R}_{\text{atm}}, \mathcal{R}_{\text{aux}}\}|\mathcal{D}_t) + \nabla_{\mathcal{D}_t} \log p_{\mathcal{G}}(\mathcal{D}_t) \\
&= \nabla_{\mathcal{D}_t} \log p_{\mathcal{G}}(\mathcal{R}_{\text{atm}}|\mathcal{D}_t) \\
&+ \nabla_{\mathcal{D}_t} \log p_{\mathcal{G}}(\mathcal{R}_{\text{aux}}|\{\mathcal{R}_{\text{atm}}, \mathcal{D}_t\}) \\
&+ \nabla_{\mathcal{D}_t} \log p_{\mathcal{G}}(\mathcal{D}_t),
\end{aligned} \tag{25}$$

and we complete the proof.

A.2 PROOF OF PROPOSITION 1.

Proposition 1. *Our training target $p(\mathcal{C}|\mathcal{R}_{atm}, \mathcal{G})$ is SE(3)-equivariant, i.e.,$p(\mathcal{C}|\mathcal{R}_{atm}, \mathcal{G}) = p(T_g(\mathcal{C})|T_g(\mathcal{R}_{atm}), \mathcal{G})$, then for all diffusion time t, the time-dependent score function is SE(3)-equivariant:*

$$\begin{aligned}
\nabla_{\mathcal{C}} \log p_t(\mathcal{C}|\mathcal{R}_{atm}, \mathcal{G}) &= \nabla_{\mathcal{C}} \log p_t(T(\mathcal{C})|T(\mathcal{R}_{atm}), \mathcal{G}) \\
&= S(\nabla_{\mathcal{C}} \log p_t(S(\mathcal{C})|S(\mathcal{R}_{atm}), \mathcal{G}))
\end{aligned} \tag{26}$$

for translation T and rotation S.

Proof. In VP-SDE, the perturbation kernel can be written as:

$$p_{t|0}(\mathcal{C}(t) \mid \mathcal{C}(0)) = \mathcal{N}\left(\mathcal{C}(t); \mathcal{C}(0)e^{-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s}, \mathbf{I} - \mathbf{I}e^{-\int_0^t \beta(s)\mathrm{d}s}\right), \tag{27}$$

which is SE(3) equivariant. We can link the perturbation kernel under translation and rotation:

$$\begin{aligned}
p_t(\mathcal{C}|\mathcal{R}_{\text{atm}}, \mathcal{G}) &= \int p_0(\mathcal{C}'|\mathcal{R}_{\text{atm}}, \mathcal{G}) p_{t|0}(\mathcal{C}|\mathcal{C}') d\mathcal{C}' \\
&= \int p_0(T_g(\mathcal{C}')|T_g(\mathcal{R}_{\text{atm}}), \mathcal{G}) p_{t|0}(T_g(\mathcal{C})|T_g(\mathcal{C}')) dT_g(\mathcal{C}') \\
&= p_t(T_g(\mathcal{C})|T_g(\mathcal{R}_{\text{atm}}), \mathcal{G}).
\end{aligned} \tag{28}$$

For T being translational transformation, we have:

$$\begin{aligned}
\nabla_{\mathcal{C}} \log p_t(\mathcal{C}|\mathcal{R}_{\text{atm}}, \mathcal{G}) &= \nabla_{\mathcal{C}} \log p_t(T(\mathcal{C})|T(\mathcal{R}_{\text{atm}}), \mathcal{G}) \\
&= \frac{\partial T(\mathcal{C})}{\partial \mathcal{C}} \nabla_{T(\mathcal{C})} \log p_t(T(\mathcal{C})|T(\mathcal{R}_{\text{atm}}), \mathcal{G}) \\
&= \nabla_{T(\mathcal{C})} \log p_t(T(\mathcal{C})|T(\mathcal{R}_{\text{atm}}), \mathcal{G}).
\end{aligned} \tag{29}$$

Similarly, for S being rotational transformation, we have

$$\begin{aligned}
\nabla_{\mathcal{C}} \log p_t(\mathcal{C}|\mathcal{R}_{\text{atm}}, \mathcal{G}) &= \nabla_{\mathcal{C}} \log p_t(S(\mathcal{C})|S(\mathcal{R}_{\text{atm}}), \mathcal{G}) \\
&= \frac{\partial S(\mathcal{C})}{\partial \mathcal{C}} \nabla_{S(\mathcal{C})} \log p_t(S(\mathcal{C})|S(\mathcal{R}_{\text{atm}}), \mathcal{G}) \\
&= S(\nabla_{S(\mathcal{C})} \log p_t(S(\mathcal{C})|S(\mathcal{R}_{\text{atm}}), \mathcal{G})),
\end{aligned} \tag{30}$$

and we complete the proof. □

B DETAILS OF DENOISING DIFFUSION PROBABILISTIC MODELS AND SCORE-BASED DIFFUSION MODEL

The forward diffusion process with T iterations of a DDPM model is defined as a fixed posterior distribution $p(\mathbf{x}_{1:T}|\mathbf{x}_0)$. Given a list of fixed variance schedule $\beta_1, ..., \beta_T$, we can define a Markov

chain process:

$$p(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} p(\mathbf{x_t}|\mathbf{x_{t-1}})$$
$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t I). \tag{31}$$

We have the following property:

Property 1. *The marginal distribution of the forward diffusion process $p(\mathbf{x}_t|\mathbf{x}_0)$ can be written as:*

$$p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)I). \tag{32}$$

This can be obtained by the following proof:

Proof. Using $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ from equation 31, we can obtain:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\mathbf{z}_t \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-1} + \sqrt{\alpha_t\beta_{t-1}}\mathbf{z}_{t-1} + \sqrt{\beta_t}\mathbf{z}_t \\
&= ... \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{\alpha_t\alpha_{t-1}...\alpha_2\beta_1}\mathbf{z}_1 + ... + \sqrt{\alpha_t\beta_{t-1}}\mathbf{z}_{t-1} + \sqrt{\beta_t}\mathbf{z}_t.
\end{aligned} \tag{33}$$

We can see that $p(\mathbf{x}_t|\mathbf{x}_0)$ can be written as a Gaussian with mean $\sqrt{\bar{\alpha}_t}\mathbf{x}_0$ and variance $(\alpha_t\alpha_{t-1}...\alpha_2\beta_1 + ... + \alpha_t\beta_{t-1} + \beta_t)I = (1-\bar{\alpha}_t)I$. □

This property allows us to write the forward diffusion process in the form of equation 5. As $T \to \infty$, the discretized equation 5 converges to the SDE form defined in equation 4.

Lemma 1. *(Tweedie's formula) Let μ be sampled from a prior probability distribution $G(\mu)$ and $z \sim \mathcal{N}(\mu, \sigma^2)$, the posterior expectation of μ given z is as:*

$$\mathbb{E}[\mu \mid z] = z + \sigma^2 \nabla_z \log p(z). \tag{34}$$

From Tweedie's formula, we can obtain the following property:

Property 2. *For DDPM with the marginal distribution $p(\mathbf{x}_t|\mathbf{x}_0)$ of the forward diffusion process computed in equation 32, $p(\mathbf{x}_0|\mathbf{x}_t)$ has a posterior mean at:*

$$\mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t] = \frac{1}{\sqrt{\bar{\alpha}(t)}}\left(\mathbf{x}_t + (1-\bar{\alpha}(t))\nabla_{\mathbf{x}_t}\log p_t(\mathbf{x}_t)\right). \tag{35}$$

C ALGORITHMS

C.1 TRAINING AND SAMPLING ALGORITHM OF BACKDIFF

We provide the training procedure in Algorithm 1 and the manifold constraint sampling procedure in Algorithm 2.

C.2 CG ATOMS CHOICE STRATEGIES

We elaborate on the CG atoms' choice strategies for the self-supervised training framework, as described in Sec. 4.2. The random strategy is shown in Algorithm 3 and the semi-random strategy is shown in Algorithm 4. In this work, we choose a semi-random strategy throughout the training, with the training ratio defined in Table 4. The training ratio value is obtained by roughly estimating the usage of each atom type in popular CG models. We notice that incorporating a larger percentage of other atom types not listed, while enhancing the generalization across different CG protocols, will require longer training time. Except for the training ratio of C_α, users can adjust the other values as needed.

Algorithm 1 Training of Backdiff

Input: proteins $[\mathcal{G}_0, ..., \mathcal{G}_N]$, each with ensembles $[\mathcal{C}_0, ..., \mathcal{C}_{K_\mathcal{G}}]$, learning rate a, CG choice strategy $\mathcal{T}$, sequence of noise levels $[\alpha_1, ..., \alpha_T]$
Output: trained score model $\mathbf{s}_\theta$
for $i = 1$ to N_{iter} **do**
  for $\mathcal{G} \sim [\mathcal{G}_0, ..., \mathcal{G}_N]$ **do**
    uniformly sample $t \sim [1, ..., T]$ and $\mathcal{C} \sim [\mathcal{C}_0, ..., \mathcal{C}_{K_\mathcal{G}}]$
    Separate $\mathcal{C}$ into CG atoms $\mathcal{R}_{\text{atm}}$ and omit atoms (backmapping targets) $\mathcal{C}_{\text{omit}}$ by the CG choice strategy $\mathcal{T}$ with the observation mask $\mathcal{M}$
    Calculate the displacement $\mathcal{D}$ of each omitted atom from its residue's C_α's position
    $\mathbf{z} \sim \mathcal{N}(0, I)$
    Calculate noisy displacement $\mathcal{D}_t = \sqrt{\alpha_t}\mathcal{D} + (1 - \alpha_t)\mathbf{z}$
    Obtain noisy configuration $\mathcal{C}_t$ from $\mathcal{D}_t$
    predict $\hat{\mathbf{s}} = \mathbf{s}_{\theta,\mathcal{G}}(\mathcal{C}_t, \mathcal{M}, t)$
    update $\theta \leftarrow \theta - a\nabla_\theta \left\| \hat{\mathbf{s}} - \nabla_{\mathcal{D}_t} \log p_{t|0}(\mathcal{D}_t \mid \mathbf{0}) \right\|^2$
  end for
end for

Algorithm 2 BackDiff sampling with manifold constraint

Input: protein molecular graph $\mathcal{G}$, CG mask $\mathcal{M}$, diffusion steps T, CG atoms $\mathcal{R}_{\text{atm}}$, CG auxiliary variables $\mathcal{R}_{\text{aux}}$, auxiliary CG mapping function $\xi_{\textbf{aux}}$, $\{\zeta_i\}_{i=1}^T, \{\tilde{\sigma}_i\}_{i=1}^T$, sequence of noise levels $[\alpha_1, ..., \alpha_T]$
Output: predicted conformers $\mathcal{C}$
$\mathcal{D}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $i = T - 1$ to 0 **do**
  Obtain noisy configuration $\mathcal{C}_i$ from $\mathcal{D}_i$ and $\mathcal{R}_{\text{atm}}$
  $\hat{\mathbf{s}} \leftarrow \mathbf{s}_\theta(\mathcal{C}_i, \mathcal{M}, t)$
  $\hat{\mathcal{D}}_0 \leftarrow \frac{1}{\sqrt{\bar{\boldsymbol{\alpha}}_i}} \left(\mathcal{D}_i + (1 - \bar{\alpha}_i)\hat{\mathbf{s}} \right)$
  $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\mathcal{D}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\mathcal{D}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\mathcal{D}}_0 + \tilde{\sigma}_i \boldsymbol{z}$
  $\mathcal{D}_{i-1} \leftarrow \mathcal{D}'_{i-1} - \zeta_i \nabla_{\mathcal{D}_i} \left\| \mathcal{R}_{\text{aux}} - \xi_{\textbf{aux}} \left(\hat{\mathcal{D}}_0, \mathcal{R}_{\text{atm}} \right) \right\|_2^2$
end for
Obtain $\mathcal{C}$ from $\hat{\mathcal{D}}_0$ and $\mathcal{R}_{\text{atm}}$

	C_α	N	C	O	C_β	Other
r	1	0.6	0.6	0.4	0.4	0.05

Table 4: The training ratio of each atom type. Atoms with the same atom types will have the same training ratio.

D CG MAPPING PROTOCOLS

In this section, we briefly introduce the three CG methods used for backmapping experiments in this paper. These CG models are designed from a mixing of knowledge-based and physics-based potentials and have been successfully applied in studying ab initio protein structure prediction, protein folding and binding, and extended to even larger systems like protein-DNA interactions. The CG mapping protocol of each method will vary from systems. In this paper, we take the general form of each protocol, summarized in Table 5. Among the three chosen CG methods, MARTINI has the highest CG resolutions: roughly four sidechain heavy atoms represented by one CG atom and two heavy atoms on the ring-like structure represented by one CG atom.

Algorithm 3 CG atoms choice: random strategy

Input: a training sample with N heavy atoms: $\mathcal{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_N]$
Output: CG atoms $\mathcal{R}_{\text{atm}}$, omitted atoms $\mathcal{C}_{\text{omit}}$, CG mask $\mathcal{M} = [m_1, m_2, \cdots, m_N]$
CG atom ratio $r \sim \text{Uniform}(0, 1)$
for $i = 1$ to N **do**:
    if atom i is a C_α **then**
        $\mathcal{C}_i \in \mathcal{R}_{\text{atm}}$
        $m_i = 0$
    else
        $p_i \sim \text{Uniform}(0, 1)$
        if $p_i > r$ **then**
            $\mathcal{C}_i \in \mathcal{C}_{\text{omit}}$
            $m_i = 1$
        else
            $\mathcal{C}_i \in \mathcal{R}_{\text{atm}}$
            $m_i = 0$
        end if
    end if
end for

Algorithm 4 CG atoms choice: semi-random strategy

Input: a training sample with N heavy atoms: $\mathcal{C} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_N]$, a pre-defined training ratio $r = [r_1, r_2, ..., r_N]$
Output: CG atoms $\mathcal{R}_{\text{atm}}$, omitted atoms $\mathcal{C}_{\text{omit}}$, CG mask $\mathcal{M} = [m_1, m_2, \cdots, m_N]$
CG atom ratio $r \sim \text{Uniform}(0, 1)$
for $i = 1$ to N **do**:
    $p_i \sim \text{Uniform}(0, 1)$
    if $p_i > r_i$ **then**
        $\mathcal{C}_i \in \mathcal{C}_{\text{omit}}$
        $m_i = 1$
    else
        $\mathcal{C}_i \in \mathcal{R}_{\text{atm}}$
        $m_i = 0$
    end if
end for

	R_{atm}	R_{aux}
MARTINI	C_α	Up to Four side chain COM beads
Rosetta	C_α, C, N, O	side chain COM
UNRES	C_α, N	side chain COM

Table 5: The CG mapping protocol of three CG methods used in this paper.

E MODIFIED TORSIONAL DIFFUSION

In this section, we briefly introduce Torsional Diffusion. Torsional Diffusion is a diffusion framework operating on the space of torsion angles. Torsion angles describe the rotation of bonds within a molecule. It lies in $[0, 2\pi)$, and a set of m torsion angles define a hypertorous space $\mathbb{T}^m$. The theory behind score-based diffusion holds for compact Riemannian manifolds, with subtle modifications. For $\boldsymbol{\tau} \in M$, where $\boldsymbol{\tau}$ represents the torsion angles and M is Riemannian manifold, the prior distribution $p_T(\mathbf{x})$ is a uniform distribution over M. We choose VE-SDE as our forward diffusion, with $\mathbf{f}(\boldsymbol{\tau}, t) = 0, g(t) = \sqrt{\frac{d}{dt}\sigma^2(t)}$, where $\sigma(t)$ represents the noise scale. We use an exponential diffusion $\sigma(t) = \sigma_{\min}^{1-t}\sigma_{\max}^t$, with $\sigma_{\min} = 0.01\pi, \sigma_{\max} = \pi, t \in (0, 1)$. As shown in equation 3, training a denoising score matching model requires sampling from the perturbation kernel $p(\boldsymbol{\tau}(t)|\boldsymbol{\tau}(0))$. We consider the perturbation kernel on $\mathbb{T}^m$ with wrapped normal distribution:

$$p(\boldsymbol{\tau}(t)|\boldsymbol{\tau}(0)) \propto \sum_{\mathbf{d}\in\mathbb{Z}^m} \exp\left(-\frac{\|\boldsymbol{\tau}(0) - \boldsymbol{\tau}(t) + 2\pi\mathbf{d}\|^2}{2\sigma^2(t)}\right), \tag{36}$$

and the other terms in the loss function equation 3 remain unchanged.

The sampling process of Torsional Diffusion is also similar to normal diffusion models with little changes: we draw samples from a uniform distribution as prior on torus space, and then discretize and solve the reverse diffusion via a geodesic random walk. We implement the model as a Torsional Diffusion conditioned on CG variables. The sampling procedure of the modified Torsional Diffusion is shown in the pseudo-code in Algorithm. 5.

Algorithm 5 Modified Torsional Diffusion sampling

Input: protein molecular graph $\mathcal{G}$, diffusion steps T, CG atoms $\mathcal{R}_{\text{atm}}$, auxiliary variables $\mathcal{R}_{\text{aux}}$ (including bond lengths l and bond angles ω)
Output: predicted conformers $\mathcal{C}$
$\boldsymbol{\tau}_T \sim U(0, 2\pi)^m$
for $i = T - 1$ to 0 **do**
  let $t = i/T, g(t) = \sigma_{\min}^{1-t}\sigma_{\max}^t\sqrt{2\ln(\sigma_{\max}/\sigma_{\min})}$
  Obtain noisy configuration $\mathcal{C}_i$ from $\boldsymbol{\tau}_i$, $\mathcal{R}_{\text{atm}}, l, \omega$
  $\hat{\mathbf{s}} \leftarrow \mathbf{s}_{\theta,\mathcal{G}}(\mathcal{C}_i, t)$
  $\boldsymbol{z} \sim$ wrapped normal with $\sigma^2 = 1/T$
  $\boldsymbol{\tau}'_{i-1} = \boldsymbol{\tau}_i + \left(g^2(t)/N\right)\hat{\mathbf{s}}$
  $\boldsymbol{\tau}_{i-1} = \boldsymbol{\tau}'_{i-1} + g(t)\boldsymbol{z}$
end for
Obtain $\mathcal{C}$ from $\boldsymbol{\tau}'_0$,$\mathcal{R}_{\text{atm}}, l, \omega$

F EVALUATION METRICS

Root Mean Squared Distance (RMSD) Root Mean Square Deviation (RMSD) is a commonly used measure in structural biology to quantify the difference between two protein structures. It's particularly useful for comparing the similarity of protein three-dimensional structures. The RMSD is calculated by taking the square root of the average of the square of the distances between the

atoms of two superimposed proteins:

$$\text{RMSD} = \min_{T_g \in \text{SE}(3)} \sqrt{\frac{1}{N}\sum_{i=1}^{N} \left\| T_g(\mathbf{r}_i) - \mathbf{r}_i^{\text{ref}} \right\|^2} \tag{37}$$

, where N is the number of atoms in the protein, and $\mathbf{r}_i$ and $\mathbf{r}_i^{\text{ref}}$ are positions of the $i-$th equivalent atoms of two structures being compared. A lower RMSD of a generated configuration indicates more similarity to the original all-atom configuration.

Generative diversity score (DIV) RMSD can be a confusing metric when evaluating the diversity of the generated samples. The main reason lies in that a high RMSD can simultaneously indicate high diversity and low accuracy. As suggested by Jones et al. (2023), the average pairwise RMSDs between (1) generated samples and the original reference and (2) between all generated samples should be approximately equal. Following this idea, a generative diversity score DIV is defined as:

$$\begin{aligned} \text{RMSD}_{\text{ref}} &= \frac{1}{N}\sum_{i}^{N} \text{RMSD}\left(\mathcal{C}_i^{\text{gen}}, \mathcal{C}^{\text{ref}}\right) \\ \text{RMSD}_{\text{gen}} &= \frac{2}{N(N-1)}\sum_{i}^{N}\sum_{j}^{(i-1)} \text{RMSD}\left(\mathcal{C}_i^{\text{gen}}, \mathcal{C}_j^{\text{gen}}\right) \\ \text{DIV} &= 1 - \frac{\text{RMSD}_{\text{gen}}}{\text{RMSD}_{\text{ref}}}, \end{aligned} \tag{38}$$

where N is the number of generated samples conditioned on a single CG configuration. DIV approximately lies on the interval $[0, 1]$. A deterministic backmapping (all generated samples are the same) will have $\text{DIV} = 1$, indicating no diversity. On the contrary, $\text{DIV} \approx 0$ is achieved when $\text{RMSD}_{\text{ref}} \approx \text{RMSD}_{\text{gen}}$, which indicates $\mathcal{C}^{\text{ref}}$ and $\mathcal{C}_i^{\text{gen}}$ shares a similiar distribution. In this case, the backmapping algorithm generates diverse all-atom configurations following a correct probability distribution. Overall this metric can indicate diversity well and avoid giving high diversity scores (low DIV) to models that generate totally off configurations.

Steric clash ratio A steric clash in protein structures refers to a situation where atoms are positioned too close to each other, leading to overlapping electron clouds. This results in an energetically unfavorable state, as it violates the principles of van der Waals radii and can destabilize the protein structure. Following GenZProt (Yang & Gómez-Bombarelli (2023)), we report the ratio of steric clash occurrence in all atom-atom pairs within 5.0 Å, where the steric clash is defined as an atom-atom pair with a distance smaller than 1.2 Å.

G EXPERIMENT DETAILS

G.1 MODEL ARCHITECTURE

Graph Neural Network (GNN) has been widely applied in molecular conformation prediction problems. In this paper, we adopt the equivariant GNN, and more specifically, e3nn library as our GNN architecture to parameterize the conditional score function $\mathbf{s}_\theta$. Following Batzner et al. (2022), we denote each node a with node representations $V_{acm}^{k,l,p}$, where k represents the message-passing layer number, l represents the rotation order, $p \in [-1, 1]$ represents the parity, with $p = 1$ representing even parity (invariant under parity), and $p = -1$ representing odd parity (equivariant under parity).

In this study, we denote the choice of CG atoms with an observation mask $\mathcal{M} = \{n_1, ..., n_N\} \in \{0, 1\}^N$, with $n_a = 0$ representing the a-th atom is a CG atom and $n_a = 1$ representing the a-th atom is an omitted atom. We then have each protein configuration data input expressed as $\{\mathcal{D}, \mathcal{R}_{\text{atm}}, \mathcal{M}, \mathcal{G}\}$. Each node in the graph $\mathcal{G}$ is represented as $v_a = \{n_a, t_a\}$, where n_a is a learnable atom type embedding fixed for a given atomic number and t_a is a learnable amino acid type embedding fixed for a given amino acid. Each edge in the graph is represented as $e_{ab} = \{v_a + v_b, s_{ab}, \mu(d_{ab}), t_{\text{GRF}}\}$, where s_{ab} is a learnable bond type embedding for a given bond type, $\mu(d_{ab})$ is the radial basis representation of distance between node a and node b, and $t_{\text{GRF}} = \{\sin 2\pi\omega t, \cos 2\pi\omega t\}$ represents the diffusion time information with Gaussian random features. Given the protein configuration data input $\{\mathcal{D}, \mathcal{R}_{\text{atm}}, \mathcal{M}, \mathcal{G}\}$ and the diffusion time t, we first

embed node and edge attributes into higher dimensional feature spaces using feedforward networks:

$$\begin{aligned} V_a^{0,0,1} &= \text{MLP}(v_a) \quad \forall v_a \in \mathcal{V}, \\ \mathbf{h}_{e_{ab}} &= \text{MLP}(e_{ab}) \quad \forall e_{ab} \in \mathcal{E}. \end{aligned} \tag{39}$$

The message-passing layers are based on E(3) equivariant convolution from Batzner et al. (2022), Jing et al. (2022). At each layer, messages passing between two paired nodes are constructed using tensor products of nodes' irreducible representation with the spherical harmonic of edge vectors. The messages are weighted by a learnable function that takes in the scalar representations ($l = 0$) of two nodes and edges. Finally, the tensor product is computed via contract with the Clebsch-Gordan coefficients. At the message-passing layer k, for the node a, its rotation order l_0, and output dimension c', the message-passing layer is expressed as:

$$V_{ac'm_o}^{(k,l_o,p_o)} = \sum_{l_f,l_i,p_i} \sum_{m_f,m_i} C_{(l_i,m_i)(l_f,m_f)}^{(l_o,m_o)} \frac{1}{|\mathcal{N}_a|} \sum_{b \in \mathcal{N}_a} \sum_{c} \psi_{abc}^{(k,l_o,l_f,l_i,p_i)} Y_{m_f}^{(l_f)}\left(\hat{r}_{ab}\right) V_{bcm_i}^{(k-1,l_i,p_i)}, \tag{40}$$

where the tensor product between the input feature of rotation order l_i and spherical harmonics of order l_f generates irreducible representations of output orders $|l_i - l_f| \leq l_o \leq |l_i + l_f|$, $(-1)^{l_f} p_i = p_o$, C represents the Clebsch-Gordan coefficients, $\mathcal{N}_a = \{b \mid \forall e_{ab} \in \mathcal{E}\}$ represents the neighboring nodes of node a, Y represents the spherical harmonics, and

$$\psi_{abc}^{(k,l_o,l_f,l_i,p_i)} = \Psi_c^{(k,l_o,l_f,l_i,p_i)} \left(\mathbf{h}_{e_{ab}} \left\| V_a^{(k-1,0,1)} \right\| V_b^{(k-1,0,1)}\right) \tag{41}$$

is the weight function using feedforward networks that take in the scalar representations of two nodes and the edge embeddings. In this paper, the rotational order of nodes (l_0, l_i) and spherical harmonics (l_f) are below 3.

After L layers of message-passing, the node feature becomes $V_a = (V^{(L,0,p)} \in \mathbb{R}^c, V^{(L,1,p)} \in \mathbb{R}^{3c}, V^{(L,2,p)} \in \mathbb{R}^{5c})$. We parameterize the time-dependent score function $\mathbf{s}_\theta(\mathcal{D}(t), t|\mathcal{R}_{\text{atm}})$ with rotational and parity equivariant feature $V_a^{(L,1,-1)}$:

$$\mathbf{s}_\theta = [V_a^{(L,1,-1)} : n_a = 1]. \tag{42}$$

G.2 Hyperparameters

In this section, we introduce the details of our experiments. The score function $\mathbf{s}_\theta$ is parameterized by the equivariant GNN presented in Sec. G.1. The atom type embedding n_a has an embedding size of 4 and the amino acid type embedding t_a has an embedding size of 8. The bond type embedding s_{ab}, which denotes if an edge represents a bonded or nonbonded interaction, has an embedding size of 2. In the initial embedding step, node and edge features are embedded into a latent dimension of 32. 8 message-passing layers as in equation 40 are used. The final 3-dimensional rotational and parity equivariant output features of each omitted atom are concatenated as the final predicted score. For the hyperparameters of the VP-SDE, we choose $\beta_1 = 1.0 \times 10^{-7}, \beta_T = 1.0 \times 10^{-3}$, with a sigmoid β scheduler and diffusion step numbers $T = 10000$. BackDiff is trained on a single NVIDIA-A10 GPU until convergence, with a training time of around 24 hours and ADAM as the optimizer, with 64 batch size.

G.3 Choice of the correction weight

An important hyperparameter in the manifold constraint sampling is the correction term weight ζ. We should expect that a too-low weight will lead to inconsistency with the conditions and an overly-high weight will make the sampling path noisy. Following Chung et al. (2022), we set $\zeta_i = \zeta_i' / \left\| \mathcal{R}_{\text{aux}} - \xi_{\textbf{aux}}\left(\hat{\mathcal{D}}_0, \mathcal{R}_{\text{atm}}\right)\right\|$, with $\zeta_i' = 0.5$ yielding the optimized sampling quality. An ablation study on the influence of correction weight is summarized in Table 6. From the table, we can see that the proposed correction weights produce the best result. Although a higher correction weight can offer stronger manifold constraints, leading to a smaller bond length and bond angle error, it over-deviates the sampling path and thus generates samples at low probability space.

	ζ_i'	PED00011	PED00055	PED00151
Bond length MAE (Å)	0.5	$< \mathbf{0.001}$	$< \mathbf{0.001}$	$< \mathbf{0.001}$
	0.01	$0.003 (< 0.001)$	$0.007 (0.002)$	$0.004 (0.001)$
	500	$< \mathbf{0.001}$	$< \mathbf{0.001}$	$< \mathbf{0.001}$
Bond angle MAE	0.5	$0.167 (0.095)$	$0.106 (0.088)$	$0.124 (0.097)$
	0.01	$0.293 (0.164)$	$0.176 (0.150)$	$0.194 (0.123)$
	500	$\mathbf{0.099 (0.003)}$	$\mathbf{0.078 (0.004)}$	$\mathbf{0.065 (0.002)}$
SCR (%)	0.5	$\mathbf{0.918 (0.609)}$	$\mathbf{0.786 (0.335)}$	$\mathbf{0.820 (0.316)}$
	0.01	$2.485 (0.743)$	$2.201 (0.469)$	$2.093 (0.554)$
	500	$1.966 (0.451)$	$1.835 (0.644)$	$1.752 (0.340)$

Table 6: Ablation study on different correction weights.

H ADDITIONAL EXPERIMENTAL RESULTS

We present the multi-protein backmapping results for Rosetta CG model in Table 7 and MARTINI CG model in Table 8. Note that the CG-transferable BackDiff model is not retrained for the two new experiments. The results further demonstrate BackDiff's enhanced accuracy and transferability. Notably, in the experiments with the MARTINI CG model, which features a higher dimensionality of CG auxiliary variables, BackDiff achieves superior backmapping results compared to its performance with the other two CG models (UNRES and Rosetta). On the other hand, baseline methods like GenZProt and Torsional Diffusion deliver similar or less impressive results with the MARTINI CG model than with UNRES and Rosetta. This indicates that BackDiff can harness the benefits of CG models with a richer set of auxiliary variables, a capability not apparent in the other methods. Additionally, we evaluate the sidechain torsion angle distribution of ground truth and sampled configurations from different methods. As shown in Figure 3, 4 and 5, BackDiff aligns closer to the ground truth distributions, even though torsion angles aren't its primary training objective.

	Method	PED00011	PED00055	PED00151
RMSD_{min}(Å)	**BackDiff (fixed)**	$\mathbf{0.616 (0.201)}$	$1.587 (0.359)$	$\mathbf{1.287 (0.163)}$
	BackDiff (trans)	$0.751 (0.222)$	$\mathbf{1.344 (0.275)}$	$1.410 (0.197)$
	GenZProt	$2.245 (0.430)$	$2.568 (0.496)$	$2.661 (0.325)$
	TD	$1.599 (0.357)$	$2.003 (0.376)$	$1.458 (0.256)$
SCR (%)	**BackDiff (fixed)**	$\mathbf{0.611 (0.456)}$	$\mathbf{0.784 (0.529)}$	$\mathbf{0.463 (0.268)}$
	BackDiff (trans)	$0.923 (0.647)$	$0.792 (0.475)$	$0.820 (0.316)$
	GenZProt	$2.215 (1.237)$	$2.192 (0.673)$	$1.545 (0.602)$
	TD	$1.034 (0.499)$	$1.205 (0.471)$	$0.772 (0.315)$
$\text{SCMSE}_{\text{min}}$ (Å^2)	**BackDiff (fixed)**	$\mathbf{0.068 (0.010)}$	$\mathbf{0.097 (0.020)}$	$0.119 (0.015)$
	BackDiff (trans)	$0.075 (0.023)$	$0.104 (0.021)$	$\mathbf{0.111 (0.044)}$
	GenZProt	$1.787 (0.289)$	$1.704 (0.368)$	$1.633 (0.301)$
	TD	$1.108 (0.309)$	$0.946 (0.247)$	$1.513 (0.350)$
DIV	BackDiff (fixed)	$0.139 (0.056)$	$0.261 (0.115)$	$0.200 (0.079)$
	BackDiff (trans)	$\mathbf{0.084 (0.060)}$	$\mathbf{0.155 (0.067)}$	$\mathbf{0.108 (0.058)}$
	GenZProt	$0.625 (0.117)$	$0.637 (0.132)$	$0.604 (0.136)$
	TD	$0.184 (0.061)$	$0.271 (0.091)$	$0.205 (0.081)$

Table 7: Results on multi-protein experiments backmapping from Rosetta CG model.

	Method	PED00011	PED00055	PED00151
RMSD_{min}(Å)	**BackDiff (fixed)**	**0.415(0.156)**	1.012(0.208)	**0.827(0.141)**
	BackDiff (trans)	0.517(0.182)	**0.827(0.174)**	0.957(0.196)
	GenZProt	2.993(0.526)	3.015(0.728)	2.982(0.552)
	TD	1.969(0.527)	2.493(0.643)	1.738(0.216)
SCR (%)	**BackDiff (fixed)**	**0.314(0.232)**	**0.629(0.512)**	**0.227(0.135)**
	BackDiff (trans)	0.536(0.478)	0.701(0.435)	0.520(0.393)
	GenZProt	2.759(0.988)	3.000(0.672)	1.894(0.433)
	TD	1.103(0.570)	1.741(0.513)	1.450(0.513)
$\text{SCMSE}_{\text{min}}$ (Å^2)	**BackDiff (fixed)**	0.035(0.008)	**0.030(0.005)**	**0.028(0.005)**
	BackDiff (trans)	**0.030(0.006)**	0.034(0.007)	0.040(0.015)
	GenZProt	2.307(0.378)	2.145(0.488)	2.389(0.404)
	TD	1.302(0.284)	1.436(0.527)	1.784(0.496)
DIV	BackDiff (fixed)	0.205(0.050)	0.325(0.087)	0.198(0.074)
	BackDiff (trans)	**0.147(0.072)**	**0.169(0.078)**	**0.152(0.063)**
	GenZProt	0.674(0.130)	0.691(0.115)	0.640(0.128)
	TD	0.282(0.056)	0.326(0.075)	0.233(0.059)

Table 8: Results on multi-protein experiments backmapping from MARTINI CG model.

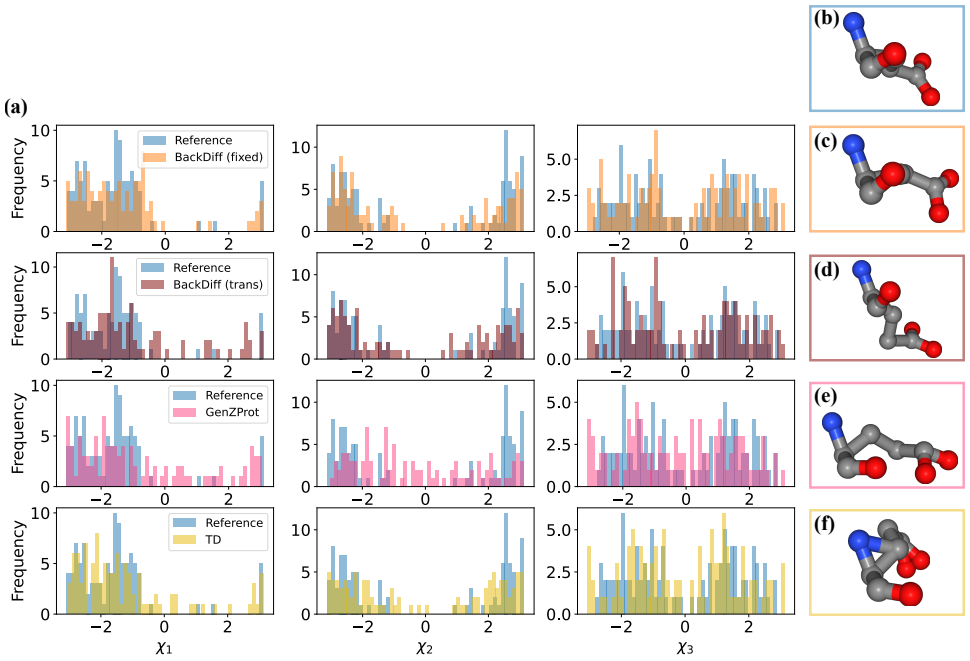

Figure 3: Multi-protein experiments backmapping from the UNRES CG model showing results on residue 7 of PED00011, a Glutamine (GLU) amino acid residue: (a) Histogram of sidechain torsion angles of ground truth and samples generated from four models, (b)-(f): the sidechain configurations visualization from (b) reference (c) fixed CG BackDiff (d) transferable CG BackDiff (e) GenZProt (f) Torsional Diffusion.

I LIMITATIONS OF BACKDIFF

As shown in Sec. 5, BackDiff significantly improves the protein backmapping accuracy. However, BackDiff has a number of limitations.

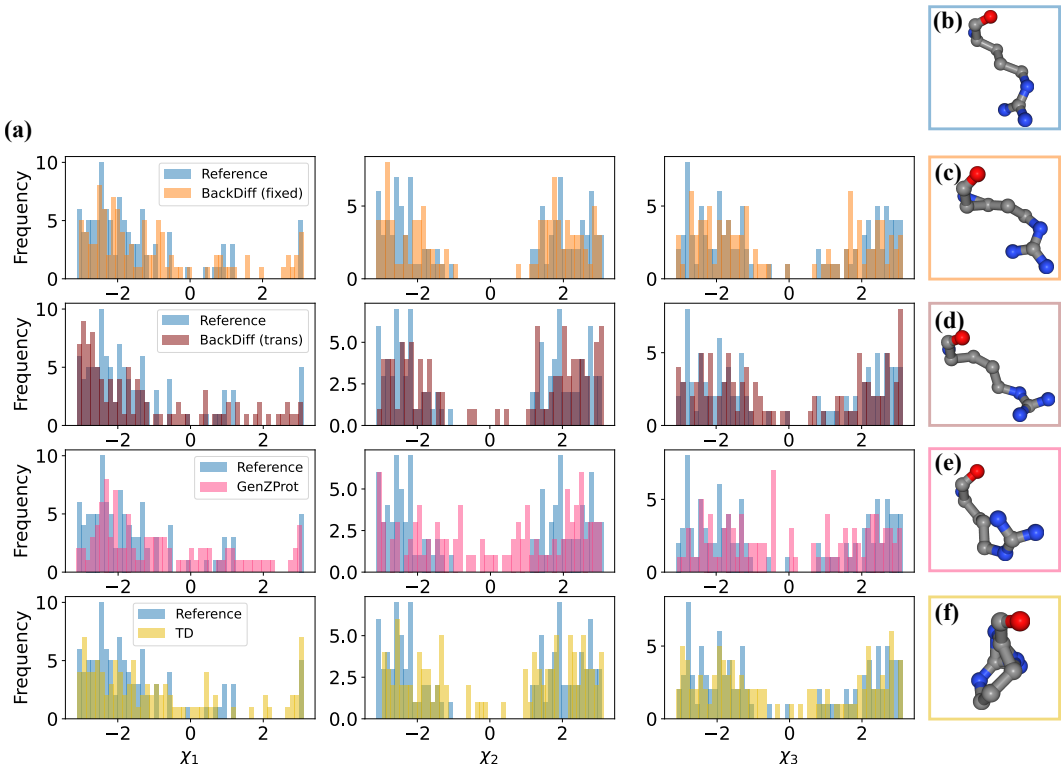

Figure 4: Multi-protein experiments backmapping from the UNRES CG model showing results on residue 8 of PED00011, an Arginine (ARG) amino acid residue.

Bond lengths and bond angles A primary drawback of BackDiff, in comparison to internal-coordinate-based generative models, is its susceptibility to producing unrealistic bond lengths and angles, even with manifold constraint sampling. This inaccuracy is notably prominent in bond angles possibly because of their nonlinear mappings from Cartesian coordinates. On the other hand, internal-coordinate-based models inherently avoid such issues by constructing geometries from predefined, reasonable bond lengths and angles. Future work will focus on refining these nonlinear manifold constraints to reduce errors in bond angles and other nonlinear CG auxiliary variables.

Sampling efficiency A notable limitation of diffusion models is their slower sampling efficiency. Compared to other generative models like Variational Autoencoders (VAE) and Normalizing Flows (NF), which often achieve generation in a single step, diffusion models require hundreds to thousands of reverse diffusion steps for effective sampling. This demand is even more pronounced for manifold constraint sampling, where fewer diffusion steps might not sufficiently constrain the conditions. In BackDiff, generating 100 samples per frame requires an average of 293 seconds, whereas for GenZProt (a VAE-based method) takes an average of 0.009 seconds. Improving the sampling efficiency of both diffusion models and manifold constraint sampling presents a compelling direction for future research.

Training data quality An optimal training dataset for BackDiff would encompass data from tens of thousands of proteins, all simulated under a unified force field. Such a dataset would ensure comprehensive coverage of the protein space and minimize inconsistencies in data quality. In contrast, our current dataset comprises a mere 92 proteins, sourced from varied simulations and sampling methodologies. Such diversity in data origins may compromise the model's broader applicability across protein spaces. Moving forward, our goal is to integrate a more expansive and consistent set of high-quality protein simulation data, enhancing the robustness and performance of BackDiff.

Chirality of proteins Proteins are made up of amino acids, most of which are chiral. This means they exist in two forms (enantiomers) that are mirror images of each other but cannot be superimposed. In nature, almost all amino acids in proteins are in the L-form (left-handed). This chirality is

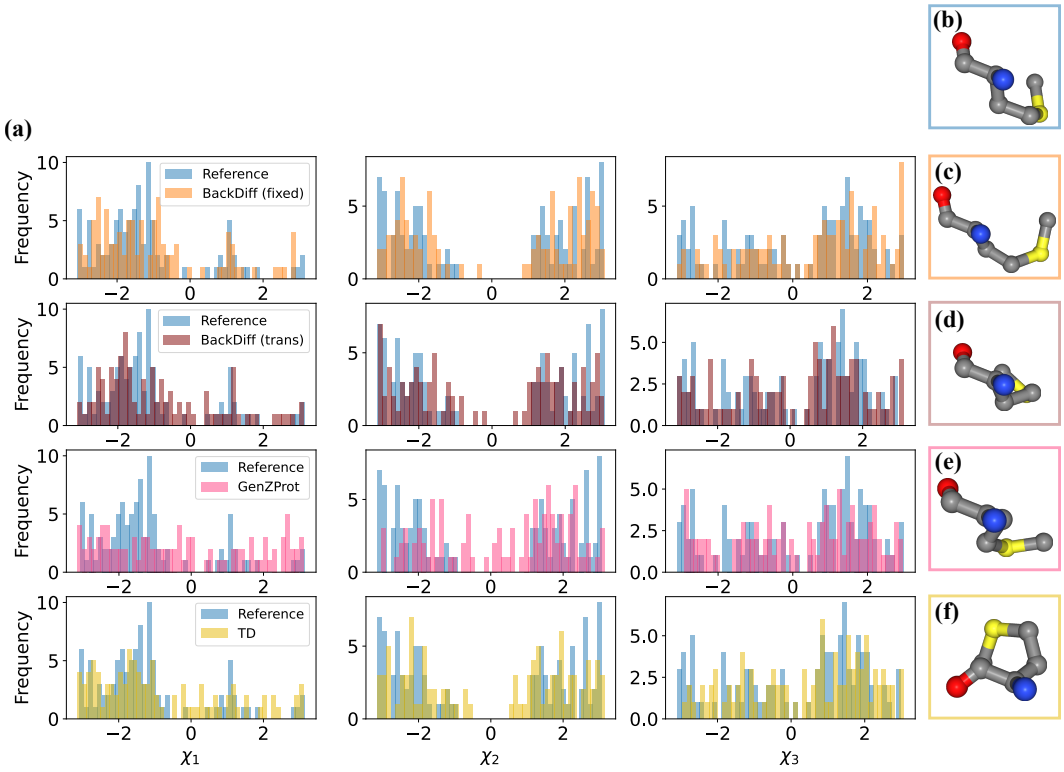

Figure 5: Multi-protein experiments backmapping from the UNRES CG model showing results on residue 27 of PED0001, a Methionine (MET) amino acid residue.

crucial for the structure and function of proteins. Performing a parity transformation on the protein will change left-handed coordinate systems into right-handed ones. Our model does not take care of the chirality and simply assumes parity equivariant: $p(\mathcal{C}|\mathcal{R}_{\text{atm}}, \mathcal{G}) = p(-\mathcal{C}| - \mathcal{R}_{\text{atm}}, \mathcal{G})$. This can be a point for improvement.