A EQUIVARIANCE PROPERTY OF SEQUENCE AND STRUCTURE TRANSLATION

We first quickly recap the process of sequence and structure translation in each translation layer. At $(t+1)^{\text{th}}$ layer, the decoder takes protein $\mathcal{P}^t = \{(\boldsymbol{s}_i^t, \boldsymbol{x}_i^t, \boldsymbol{O}_i^t)\}_{i=1}^N$ and context features $\{\boldsymbol{m}_i^t\}, \{\boldsymbol{z}_{ij}^t\}$ as the input. It encodes sequence-structure interplay and integrates all interactions into updated context features using SeqIPA adapted from Invariant Point Attention (IPA) (Jumper et al., 2021) in a way that its roto-translation invariant property is kept. The updates of C_α positions, frame orientations, and type distributions are then predicted based on updated context features. The whole process can be summarized as follows:

$$\boldsymbol{s}_i^{t+0.5} = \mathrm{MLP}_{\mathrm{e}}(\boldsymbol{s}_i^t), \tag{12}$$

$$\boldsymbol{m}_i^{t+1}, \boldsymbol{z}_{ij}^{t+1} = \mathrm{SeqIPA}(\{\boldsymbol{m}_i^t\}, \{\boldsymbol{z}_{ij}^t\}, \{\boldsymbol{s}_i^{t+0.5}\}, \{\boldsymbol{x}_i^t\}, \{\boldsymbol{O}_i^t\}), \tag{13}$$

$$\hat{\boldsymbol{x}}_i^t = \mathrm{MLP}_{\mathrm{x}}(\boldsymbol{m}_i^{t+1}, \boldsymbol{m}_i^0), \quad \boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \Delta\boldsymbol{x}_i^t = \boldsymbol{x}_i^t + \boldsymbol{O}_i^t \hat{\boldsymbol{x}}_i^t, \tag{14}$$

$$\hat{\boldsymbol{O}}_i^t = \mathrm{convert}\left(\mathrm{MLP}_{\mathrm{o}}(\boldsymbol{m}_i^{t+1}, \boldsymbol{m}_i^0)\right), \quad \boldsymbol{O}_i^{t+1} = \boldsymbol{O}_i^t \hat{\boldsymbol{O}}_i^t, \tag{15}$$

$$\boldsymbol{s}_i^{t+1} = \mathrm{softmax}\left(\lambda \cdot \mathrm{MLP}_{\mathrm{s}}(\boldsymbol{m}_i^{t+1}, \boldsymbol{m}_i^0, \boldsymbol{s}_i^{t+0.5})\right). \tag{16}$$

To derive the equivariance property of each translation step, we use three functions $\mathcal{X}, \mathcal{O}, \mathcal{S}$ to denote the network that predicts the C_α position translation, orientation translation, and sequence translation described above, respectively. Formally, we have:

$$\Delta\boldsymbol{x}_i^t = \mathcal{X}(\mathcal{P}^t), \tag{17}$$

$$\boldsymbol{O}_i^{t+1} = \mathcal{O}(\mathcal{P}^t), \tag{18}$$

$$\boldsymbol{s}_i^{t+1} = \mathcal{S}(\mathcal{P}^t). \tag{19}$$

Note that $\mathcal{X}, \mathcal{O}, \mathcal{S}$ also take $\{\boldsymbol{m}_i^t\}$ and $\{\boldsymbol{z}_{ij}^t\}$ as input and we omit these context features for simplicty, as they remain invariant to global rigid transformations. $\mathcal{X}, \mathcal{O}, \mathcal{S}$ are not separate networks, and they share the same input and the same SeqIPA, but are equipped with different MLPs. With the above definitions, we can derive the following proposition:

Proposition 1 (Roto-Translation Equivariance)**.** *Let $\mathcal{T}_{\boldsymbol{R},\boldsymbol{r}}$ denote any SE(3) transformation (rigid transformation) operating on the protein object $\mathcal{P}^t = \{(\boldsymbol{s}_i^t, \boldsymbol{x}_i^t, \boldsymbol{O}_i^t)\}_{i=1}^N$, with a rotation matrix $\boldsymbol{R} \in \mathrm{SO}(3)$ and a translation vector $\boldsymbol{r} \in \mathbb{R}^3$. The function $\mathcal{X}, \mathcal{O}, \mathcal{S}$ satisfy the following equivariance properties:*

$$\mathcal{X} \circ \mathcal{T}_{\boldsymbol{R},\boldsymbol{r}}(\mathcal{P}^t) = \boldsymbol{R}\mathcal{X}(\mathcal{P}^t), \tag{20}$$

$$\mathcal{O} \circ \mathcal{T}_{\boldsymbol{R},\boldsymbol{r}}(\mathcal{P}^t) = \boldsymbol{R}\mathcal{O}(\mathcal{P}^t), \tag{21}$$

$$\mathcal{S} \circ \mathcal{T}_{\boldsymbol{R},\boldsymbol{r}}(\mathcal{P}^t) = \mathcal{S}(\mathcal{P}^t), \tag{22}$$

where $\mathcal{T}_{\boldsymbol{R},\boldsymbol{r}}(\mathcal{P}^t) = \{(\boldsymbol{s}_i^t, \boldsymbol{x}_i^t + \boldsymbol{r}, \boldsymbol{R}\boldsymbol{O}_i^t)\}_{i=1}^N$.

Intuitively, the proposition states that in each translation step, the updates of C_α positions and frame orientations are equivariant with respect to input protein structures, and the updates of type distributions are invariant.

Proof. We first prove that Eq. 20 holds. Notice that SeqIPA is aware of the orientations of the input structure, and the updated context features are invariant (Eq. 13). Therefore, the predicted deviation of C_α positions, i.e., $\hat{\boldsymbol{x}}_i^t$, is invariant (Eq. 14). Then, we have:

$$\mathcal{X} \circ \mathcal{T}_{\boldsymbol{R},\boldsymbol{r}}(\mathcal{P}^t) = \boldsymbol{R}\boldsymbol{O}_i^t \hat{\boldsymbol{x}}_i^t = \boldsymbol{R}\mathcal{X}(\mathcal{P}^t). \tag{23}$$

The Eq. 21 and Eq. 22 can be proved in a similar way. □

B MODEL DETAILS

B.1 PSEUDO CODE

Algorithm 1 PROTSEED

Require: Initial single features $\{\boldsymbol{m}_i\} \in \mathbb{R}^{N \times c_m}$ and pair features $\{\boldsymbol{z}_{ij}\} \in \mathbb{R}^{N \times N \times c_z}$.

$\boldsymbol{m}_i^0, \boldsymbol{z}_{ij}^0 \leftarrow \text{Linear}(\boldsymbol{m}_i), \text{Linear}(\boldsymbol{z}_{ij})$ ▷ $\boldsymbol{m}_i^0 \in \mathbb{R}^c, \boldsymbol{z}_{ij}^0 \in \mathbb{R}^c$
for $l \leftarrow 0$ to $L-1$ **do**
  $\boldsymbol{m}_i^{l+1} \leftarrow \text{MHA}(\{\boldsymbol{m}_i^l\}, \{\boldsymbol{z}_{ij}^l\})$ ▷ Eq. 1
  $\boldsymbol{z}_{ij}^{l+0.5} \leftarrow \boldsymbol{z}_{ij}^l + \text{Linear}(\boldsymbol{m}_i^{l+1} \otimes \boldsymbol{m}_j^{l+1})$ ▷ Eq. 2
  $\boldsymbol{z}_{ij}^{l+0.75} \leftarrow \boldsymbol{z}_{ij}^{l+0.5} + \text{TriangleUpdate}_1(\{\boldsymbol{z}_{ij}^{l+0.5}\})$ ▷ Eq. 4
  $\boldsymbol{z}_{ij}^{l+1} \leftarrow \boldsymbol{z}_{ij}^{l+0.75} + \text{TriangleUpdate}_2(\{\boldsymbol{z}_{ij}^{l+0.75}\})$ ▷ Eq. 5
end for
$\boldsymbol{m}_i^0, \boldsymbol{z}_{ij}^0 \leftarrow \boldsymbol{m}_i^L, \boldsymbol{z}_{ij}^L$ ▷ Initialize context features for decoder
$\mathcal{P}^0 \leftarrow \{(\boldsymbol{s}_i^0, \boldsymbol{x}_i^0, \boldsymbol{O}_i^0)\}_{i=1}^N \leftarrow \{(\frac{1}{20} \cdot \mathbf{1}, (0,0,0), \boldsymbol{I_3})\}_{i=1}^N$ ▷ Initialize protein $\mathcal{P}^0$
for $t \leftarrow 0$ to $T-1$ **do**
  $\boldsymbol{s}_i^{t+0.5} \leftarrow \text{MLP}_\text{e}(\boldsymbol{s}_i^t)$ ▷ $\boldsymbol{s}_i^{t+0.5} \in \mathbb{R}^c$
  $\boldsymbol{m}_i^{t+1}, \boldsymbol{z}_{ij}^{t+1} \leftarrow \text{SeqIPA}(\{\boldsymbol{m}_i^t\}, \{\boldsymbol{z}_{ij}^t\}, \{\boldsymbol{s}_i^{t+0.5}\}, \{\boldsymbol{x}_i^t\}, \{\boldsymbol{O}_i^t\})$ ▷ Eq. 6 and Section B.2
  $\hat{\boldsymbol{x}}_i^t \leftarrow \text{MLP}_\text{x}(\boldsymbol{m}_i^{t+1}, \boldsymbol{m}_i^0)$ ▷ Deviation of C_α positions in local frame
  $\boldsymbol{x}_i^{t+1} \leftarrow \boldsymbol{x}_i^t + \boldsymbol{O}_i^t \hat{\boldsymbol{x}}_i^t$ ▷ Deviation of C_α positions in global frame
  $\hat{\boldsymbol{O}}_i^t \leftarrow \text{convert}\left(\text{MLP}_\text{o}(\boldsymbol{m}_i^{t+1}, \boldsymbol{m}_i^0)\right)$ ▷ Convert a quaternion to a rotation matrix
  $\boldsymbol{O}_i^{t+1} \leftarrow \boldsymbol{O}_i^t \hat{\boldsymbol{O}}_i^t$
  $\boldsymbol{s}_i^{t+1} \leftarrow \text{softmax}\left(\lambda \cdot \text{MLP}_\text{s}(\boldsymbol{m}_i^{t+1}, \boldsymbol{m}_i^0, \boldsymbol{s}_i^{t+0.5})\right)$
  $\mathcal{P}^{t+1} \leftarrow \{(\boldsymbol{s}_i^{t+1}, \boldsymbol{x}_i^{t+1}, \boldsymbol{O}_i^{t+1})\}_{i=1}^N$ ▷ Eq. 9
end for
Return: The trajectory of the protein translation $\{\mathcal{P}^t\}_{t=1}^T$.

B.2 PARAMETERIZATION OF SEQIPA

SeqIPA is adapted from the Invariant Point Attention (IPA) (Jumper et al., 2021), which takes residue types as the additional input to capture the interactions between current decoded sequences, structures, and the context features. We ensure that the additional input does not affect the invariance property of the IPA to make full use of its capacity. Specifically, we propose the following two strategies to parameterize the SeqIPA.

SeqIPA-Addition. Given that $\{\boldsymbol{s}_i^{t+0.5}\}$ share the same dimensionality with $\{\boldsymbol{m}_i^t\}$, a very simple strategy is to just add embeddings of residue types onto single representations. Following the original implementation of IPA, we leave the pair features unchanged in this approach.

$$\boldsymbol{m}_i^{t+1}, \boldsymbol{z}_{ij}^{t+1} = \text{SeqIPA}(\{\boldsymbol{m}_i^t\}, \{\boldsymbol{z}_{ij}^t\}, \{\boldsymbol{s}_i^{t+0.5}\}, \{\boldsymbol{x}_i^t\}, \{\boldsymbol{O}_i^t\}) \tag{24}$$

$$= \text{IPA}(\{\boldsymbol{m}_i^t + \boldsymbol{s}_i^{t+0.5}\}, \{\boldsymbol{z}_{ij}^t\}, \{\boldsymbol{x}_i^t\}, \{\boldsymbol{O}_i^t\}). \tag{25}$$

SeqIPA-Attention. Another more complicated strategy is to construct a new set of single representations and pair representations based on the embeddings of the current residue types. Then, we adopt a lightweight encoder similar to the encoder introduced in Section 3.2 to update $\boldsymbol{m}_i^t$ and $\boldsymbol{z}_{ij}^t$, which are then fed into the vanilla IPA module. We summarize the computation flow as follows:

$$\bar{\boldsymbol{m}}_i, \bar{\boldsymbol{z}}_{ij} = \text{Linear}(\boldsymbol{s}_i^{t+0.5}), \text{Linear}(\boldsymbol{s}_i^{t+0.5} + \boldsymbol{s}_j^{t+0.5}) \tag{26}$$

$$\bar{\boldsymbol{m}}_i, \bar{\boldsymbol{z}}_{ij} = \text{Encoder}(\{\bar{\boldsymbol{m}}_i\}, \{\bar{\boldsymbol{z}}_{ij}\}), \tag{27}$$

$$\boldsymbol{m}_i^{t+0.5}, \boldsymbol{z}_{ij}^{t+0.5} = \boldsymbol{m}_i^t + \bar{\boldsymbol{m}}_i, \boldsymbol{z}_{ij}^t + \bar{\boldsymbol{z}}_{ij}, \tag{28}$$

$$\boldsymbol{m}_i^{t+1}, \boldsymbol{z}_{ij}^{t+1} = \text{SeqIPA}(\{\boldsymbol{m}_i^t\}, \{\boldsymbol{z}_{ij}^t\}, \{\boldsymbol{s}_i^{t+0.5}\}, \{\boldsymbol{x}_i^t\}, \{\boldsymbol{O}_i^t\}) \tag{29}$$

$$= \text{IPA}(\{\boldsymbol{m}_i^{t+0.5}\}, \{\boldsymbol{z}_{ij}^{t+0.5}\}, \{\boldsymbol{x}_i^t\}, \{\boldsymbol{O}_i^t\}). \tag{30}$$

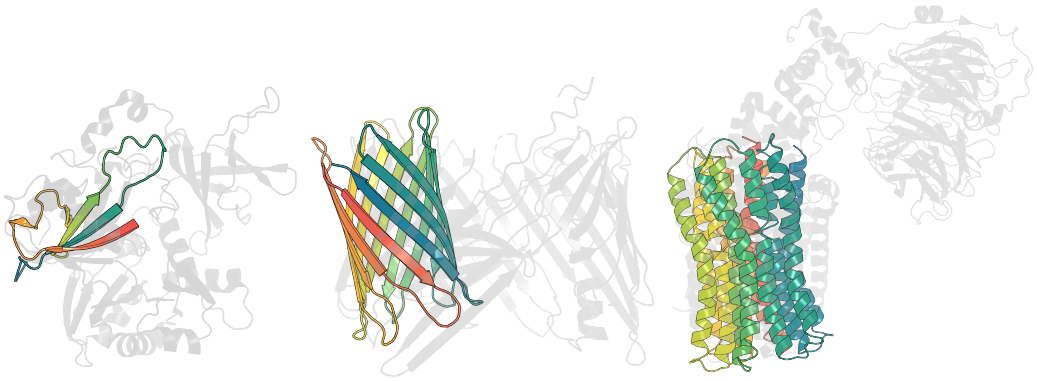

Figure 5: Superimposition of three generated proteins and their most similar proteins found in the PDB by FoldSeek. left: a novel protein with loop extended. middle: a novel $\beta-$barrel. right: a novel helical complex.

In practice, we find both strategies work well and their performance is on par with each other. To make the whole model lightweight, we adopt the first strategy across all the experiments in this work. We emphasize that the parameterization of the SeqIPA is quite flexible, as long as it can model interactions between sequences, structures, and context features, and is invariant to the global transformation of input structures.

B.3 HYPER-PARAMETERS

PROTSEED is implemented in Pytorch. The trigonometry-aware context encoder is implemented with $L = 8$ layers, and the sequence-structure decoder is implemented with $T = 8$ layers. The hidden dimension is set as 256 across all modules. For training, we use a learning rate of 0.001 with 2000 linear warmup iterations. The model is optimized with Adam optimizer on four Tesla V100 GPU cards. For inference, the temperature of the sequence distribution, i.e., λ, is set as 1. We will release the source code of this work upon acceptance.

C EXPERIMENTAL DETAILS

C.1 CASE STUDY

We conduct three case studies to evaluate PROTSEED's capability to perform *de novo* protein design, including extending the loop of existing proteins, designing novel $\beta-$barrels, and designing novel helical complexes. Specifically, we manually curate a set of secondary structure annotations and contact features, and ask the model trained in the second task (Section 4.2) to generate novel proteins based on these context features. We elaborate the way we design context features for each setting.

Extending the Loop. In this setting, we start by calculating the secondary structure annotations and the contact matrix of an existing protein. We then insert n consecutive "C" ("C" is the secondary structure annotation for the loop) letters into the original secondary structure annotations at the position where we want to extend the loop. Similarly, we insert n consecutive rows and columns filled with zero into the original contact matrix. For a new-inserted residue indexed by i, we let it to be in contact with $i-2, i-1, i, i+1, i+2$.

Designing Novel Beta Barrels. In this setting, we grab a simple pattern of $\beta-$barrel proteins and then repeat this pattern multiple times to construct the contact features. The secondary structure annotations are also calculated by repeating the annotations of the pattern multiple times.

Designing Novel Helical Complexes. Similar to the second case, in this setting, we also take a simple pattern of helical complexes and construct the contact features by repeating it multiple times. The secondary structure annotations are all set to be "H".

In Figure 5, we show the superimposition of three novel proteins designed by PROTSEED against the most similar proteins in the PDB, one for each setting, which confirms the novelty of the designed proteins.