PROGAE: A GEOMETRIC GENERATIVE MODEL FOR DISENTANGLING PROTEIN CONFORMATIONAL SPACE

N. Joseph Tatro
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180
tatron@rpi.edu

Payel Das, Pin-Yu Chen, & Vijil Chenthamarakshan
IBM Research
Yorktown, NY 10598

Rongjie Lai
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180

ABSTRACT

Understanding the protein conformational landscape is critical, as protein function in processes such as ligand binding is intimately connected with structural variations. This work focuses on learning a generative neural network on a simulated ensemble of protein structures obtained from molecular simulation to characterize the distinct structural fluctuations of a protein bound to various drug molecules. Specifically, we use a geometric autoencoder to learn separate latent space encodings of the intrinsic and extrinsic geometries of the system. Our proposed Protein Geometric AutoEncoder (ProGAE) model is trained on the length of the alpha-carbon pseudobonds and the orientation of the backbone bonds of the protein. Using ProGAE latent embeddings, we reconstruct and generate the conformational ensemble of a protein at or near the experimental resolution, while gaining better interpretability and controllability of the learned latent space. Results show that our geometric learning-based method enjoys both accuracy and efficiency for generating complex structural variations, charting the path toward scalable and improved approaches for analyzing and enhancing molecular simulations.

1 INTRODUCTION

The complex and time-consuming calculations in molecular simulations have been significantly impacted by the application of machine learning in recent years. For a comprehensive review of such work, see (Noé et al., 2020a;b). There has been interest in modeling the conformational space of proteins using deep generative models, e.g. (Bhowmik et al., 2018; Ramaswamy et al., 2020). In this work, we learn the protein conformational space from a set of protein simulations by using geometric deep learning. We also investigate how the geometry of a protein can assist learning and improve interpretability of the latent conformational space. Our main contributions summarized:

- Inspired by recent unsupervised geometric disentanglement learning works (Tatro et al., 2020; Yang et al., 2020), we propose a novel geometric autoencoder named ProGAE that directly learns from 3D protein structures via separately encoding intrinsic and extrinsic geometries.
- We find that the intrinsic geometric latent space, even with a small variation, is important for reducing geometric errors in reconstructed proteins.
- Analysis shows the learned extrinsic geometric latent space can be used for drug classification and property prediction, where the drug is bound to the given protein.

Related Work Several recent papers use AE-based approaches for either analyzing and/or generating structures from the latent space (Bhowmik et al., 2018; Guo et al., 2020; Ramaswamy et al.,

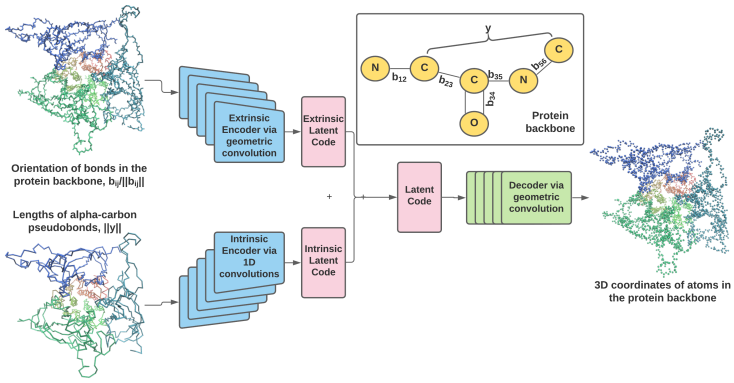


Figure 1: Architecture of our network, ProGAE, that generates protein conformations via separate encoding of data related to coarse intrinsic and extrinsic geometries. These geometries are captured via the orientation of the backbone bonds (extrinsic) and length of $C_\alpha - C_\alpha$ pseudobonds (intrinsic).

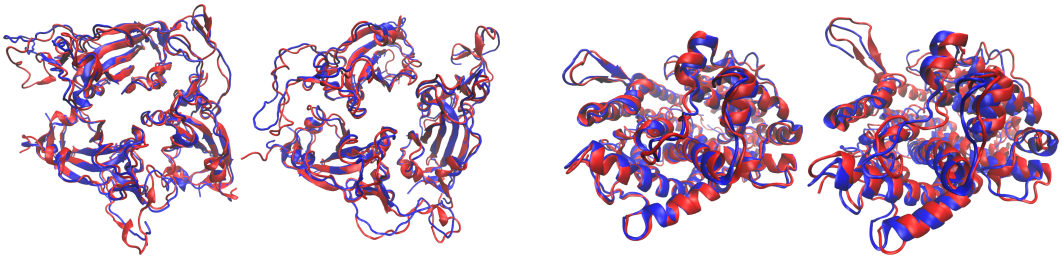


Figure 2: ProGAE reconstructions of S protein (left) and hACE2 data (right). Blue and red structures correspond to the reconstructed and ground truth structures, respectively.

2020; Varolgüneş et al., 2020) . Bhowmik et al. (2018) and Guo et al. (2020) aim at learning from and generating protein contact maps, while ProGAE directly deals with 3D structures. Ramaswamy et al. (2020) trains a 1D CNN autoencoder on backbone coordinates and uses a loss objective comprised of geometric MSE error and physics-based error. We run ProGAE on the same MurD protein simulations studied in Ramaswamy et al. (2020) and compare the reconstruction quality with respect to the value reported in that study as well as to the experimental resolution.

To our knowledge, our work is the first to propose an autoencoder for the unsupervised modeling of the geometric disentanglement of protein conformational space captured in molecular simulations. This representation provides better interpretability of the latent space, in terms of the physicochemical and geometric attributes and results in more geometrically accurate protein conformations.

2 ProGAE for Protein Conformational Space

Geometric Features as Network Input ProGAE separately encodes intrinsic and extrinsic geometry with the goal of achieving better latent space interpretability. Mathematically, we can consider a manifold (i.e. curve) independent of its embedding in Euclidean space. Properties that do not

Table 1: The leading canonical correlation between the intrinsic and extrinsic ProGAE latent spaces, and the performance of a linear model trained on the latent spaces for bound drug classification.

Drug classification	**S protein**	**hACE2**
1st canonical corr.	0.08 ± 0.00	0.07 ± 0.01
Trained on intrinsic	$2.0 \pm 0.1\%$	1.4 ± 0.0 %
Trained on extrinsic	$99.6 \pm 0.3\%$	99.6 ± 0.2 %

Table 2: Reconstruction, bond length, and RMSD (from ground truth) error on the test sets using ProGAE. The RMSD is within the resolution of the associated PDB files.

	Errors	**S protein**	**hACE2**	Benchmark **MurD**
Exp. Res. (Å)		2.68/2.80	2.20/3.00	2.40/1.77/1.84
Test	Reconstruction	1.39 ± 0.01 E0	6.27 ± 0.03 E-1	2.02 ± 0.32 E-1
	Bond length (Å)	3.88 ± 0.02 E-1	1.63 ± 0.02 E-1	1.73 ± 0.06 E-1
	RMSD (Å)	2.54 ± 0.56	1.24 ± 0.23	1.79 ± 0.36

depend on an embedding are known as intrinsic geometric properties, with others referred to as extrinsic. For an in-depth review of geometry, we refer to (Do Carmo, 2016).

As we will learn the conformational space of a given protein, the protein primary structure is implicit. Then we view the protein at the level of its backbone, as it is sufficient for reconstructing it. Of importance in the backbone are the C_α atoms. A coarse-level description of the backbone is the C_α atoms connected linearly in terms of the protein sequence, known as the trace of the protein.

We model the backbone by the graph, $\mathcal{G}_b = (\mathbb{V}_b, \boldsymbol{E}_b)$, and the backbone trace by the graph, $\mathcal{G}_t = (\mathbb{V}_t, \boldsymbol{E}_t)$. Then our introduced intrinsic and extrinsic signals, $Int : \boldsymbol{E}_t \to \mathbb{R}$ and $Ext : \boldsymbol{E}_b \to \mathbb{R}^3$ are defined, $Int(E_{ij}) = \|E_{ij}\|_2, \quad E_{ij} \in \boldsymbol{E}_t$ and $Ext(E_{ij}) = sgn(j-i)\frac{E_{ij}}{\|E_{ij}\|}, \quad E_{ij} \in \boldsymbol{E}_b$. These correspond to the lengths of $C_\alpha - C_\alpha$ pseudobonds and backbone bond orientations. We will see they allow us to faithfully reconstruct protein backbone. These signals are depicted in Figure 1.

Network Architecture The core idea is to create an *intrinsic* latent space, $L_I \in \mathbb{R}^{n_i}$, and an *extrinsic* latent space, $L_E \in \mathbb{R}^{n_e}$, via separately encoding the intrinsic and extrinsic signals. Consequently, our network contains two encoders, Enc_i and Enc_e. Here $Enc_i \circ Int(\boldsymbol{E}_t) \in L_I$ and $Enc_e \circ Ext(\boldsymbol{E}_b) \in L_E$. We then jointly decode these latent vectors to recover the coordinates of the backbone atoms. Thus, we define the decoder, $Dec : L_I \times L_E \to \mathbb{R}^{|\mathbb{V}_b| \times 3}$.

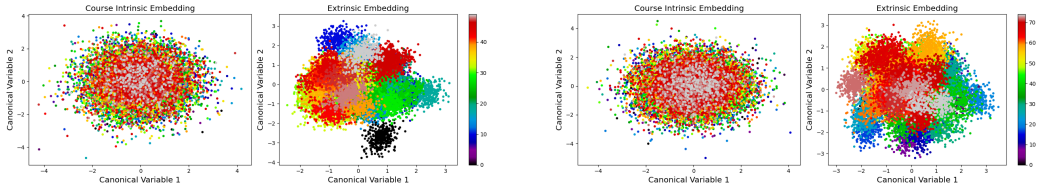
This high level structure of ProGAE is depicted in Figure 1. Specific details on layer widths and other parameters can be found in Appendix A.1. The intrinsic encoder is simple as the signal is defined on the backbone trace, which corresponds to a set of discrete curves. Then we define Enc_i to be a series of 1D convolutions operating on each curve/fragment. In contrast, the extrinsic encoder operates on the backbone, which is a graph. So the layers of graph attention networks (GATs) introduced in (Veličković et al., 2017) are a natural tool to use, albeit with some modification. Since the input signal is defined only on the edges of the graph, $\boldsymbol{E}_b$, we define a signal on the graph vertices, $\mathbb{V}_b$, as the average value of its incident edges, $f_0(v_i)$.

Then the first layer of the extrinsic encoder uses the edge-convolution operator of (Gong & Cheng, 2019) to map this edge-defined signal to a vertex-defined signal. The rest of the encoder contains successive graph attention layers with sparsity defined by a neighborhood radius. At each layer, the signal is downsampled by a factor of two based on farthest point sampling. Given L layers, this defines a sequence of graphs, $\{\mathcal{G}_{b,i}\}_{i=0}^{L}$, with increasing decimation. Along with Enc_i, each layer is followed with batch normalization and ReLU. Summarily, for $l = 2, ..., L$,

$$f_l = \sigma \circ BN \circ GAT(DS(f_{l-1}; 2)), \quad f_1(v_i) := GAT(f_0(\mathbb{V}_b), Ext(\boldsymbol{E}_b)). \tag{1}$$

The intrinsic and extrinsic latent codes, $\boldsymbol{z}_i$ and $\boldsymbol{z}_e$, are produced after global average pooling, a dense layer, and Tanh function are applied to the output of the encoders.

The latent code $\boldsymbol{z}$ is taken as the concatenation of the two latent codes, $[\boldsymbol{z}_i, \boldsymbol{z}_e]$. A dense layer maps $\boldsymbol{z}$ to the a signal defined on the most decimated backbone graph, $\mathcal{G}_{b,L}$. The structure of the decoder, Dec, mirrors Enc_e with convolutions mapping to upscaled graphs. The output of Dec is the point cloud, $\hat{\boldsymbol{P}}$, corresponding to the predicted coordinates of the backbone atoms, $\mathbb{V}_b \approx \boldsymbol{P}$.



(a) S protein (b) Human ACE2

Figure 3: The projection of the latent space embedding to the first two canonical vectors between the intrinsic and extrinsic latent spaces. Color indicates the identity of the drug that the protein is bound to in that conformation. Clustering by drug identity is apparent in the extrinsic latent space, but not the intrinsic latent space, consistent with Table 1.

Table 3: Percentage of bonds that are 10% shorter than the minimum seen in training data. The difference (Diff) between the intrinsic+extrinsic ProGAE and the extrinsic-only ProGae is reported.

Dataset		C-CA	C-N	C-O	CA-N	CA-CA
S protein	Int.+Ext. (%)	14.41	22.58	27.00	15.24	15.33
	Ext. Only (%)	19.04	27.34	27.66	18.58	20.89
	Diff of Adding Int.	-4.63	-4.76	-0.66	-3.40	-5.56
hACE2	Int.+Ext. (%)	2.45	8.19	12.07	4.62	0.51
	Ext. Only (%)	4.99	9.81	12.34	5.21	1.59
	Diff of Adding Int.	-2.54	-1.62	-0.27	-0.59	-1.08

3 Numerical Experiments and Results

For each dataset, we train three models, each from a different random seed, and report both mean and standard deviation in our results. Datasets used are atomistic simulation trajectories [1] (D.E. Shaw Research, 2020). These two main datasets are: (1) *50 independent trajectories*, each simulating the SARS-CoV-2 trimeric spike protein (S protein) in the presence of a distinct drug for 2μs. ; (2) *75 independent trajectories*, each simulating the ectodomain protein of human ACE2 (hACE2) in the presence of a distinct drug for 2μs. For comparing with existing work, we run ProGAE on MurD protein simulation data (Ramaswamy et al., 2020) publicly available [2]. Further details on the datasets and their use in training can be found in appendix A.2.

Structure Reconstruction Figure 2 displays the ability of ProGAE to accurately reconstruct protein conformations. The backbones are visible with atom-wise error in Figures 4a and 4b in the appendix. Table 2 contains performance metrics, such as RMSD (after alignment), of ProGAE. In either case, the RMSD of the reconstruction is within the experimental resolution of the associated PDB files. The average error in the length of the pseudobonds is also sub-Angstrom. The RMSD (on secondary structure elements) on the benchmark MurD test data (Ramaswamy et al., 2020) is lower or comparable to the experimental resolution and within the range of what has been reported in the original study that uses more explicit loss (bond, angle, nonbonded) terms compared to ProGAE.

We also evaluate the performance of linear interpolations in the learned latent space. Results of interpolation between conformations from different trajectories in terms of RMSD is shown in Figure 5 in the appendix. A smooth exchange in the RMSD error from both endpoints is evident.

Utility of the Extrinsic Latent Space We explore the statistical relationship between the learned intrinsic latent space and the extrinsic latent space. Canonical correlation analysis (CCA) is a natural approach to assess if a linear relationship exists (Hardoon et al., 2004). Table 1 includes the leading correlation between these spaces for each dataset. Note this correlation is very low, implying

[1] available here: `http://www.deshawresearch.com/resources_sarscov2.html`
[2] `https://collections.durham.ac.uk/files/r26w924b81m#.YBuGGi-z2Mw`

that there is a negligible linear relationship between the intrinsic and extrinsic latent spaces. This confirms a notable level of disentanglement that has been explicitly encoded in our architecture.

We investigate if the distinct drug information associated with a trajectory is encoded in the two disentangled latent spaces. Table 1 contains the performance of a linear classifier trained on the different latent spaces to classify the drug present in each frame. It is clear that the drug molecule can be almost perfectly classified in the extrinsic latent space, while such classification is random in the intrinsic latent space. Figures 3a and 3b visualize these embeddings of the test set in the latent spaces, projected to the first two canonical components. We also train a linear regression model on the extrinsic latent space to predict physico-chemical properties of the bound drug. Results comparing this regression to one on the PCA embedding are in Table 4 in the appendix.

Utility of the Intrinsic Latent Space The inclusion of the intrinsic latent space improves the geometric validity of the reconstructed protein. We trained a model that only encodes the extrinsic signal to reconstruct the protein. While it was comparable in performance regarding L_2 error, we found this extrinsic-only model resulted in a higher percentage of erroneous bonds. This is shown in Table 3. We define a erroneous bond, if the bond length deviates by more than 10% from the minimum of the ground truth distribution, as such deviations will result in steric clashes.

4 Conclusion

We introduce a novel geometric autoencoder, ProGAE, for learning meaningful disentangled representations of the protein conformational space. The autoencoder separately encodes intrinsic and extrinsic geometries to ensure better latent interpretability. The extrinsic latent space can classify structures with respect to their bound drug molecules. The intrinsic space improves the validity of bond geometry in the reconstructions. The disentangled, smooth latent space enables controllable generation in a drug-dependent manner. These results suggest that the proposed framework can serve as a step towards bridging geometric deep learning with molecular simulations.

Acknowledgments

J. Tatro's work was supported by the IBM-RPI AIRC program. R. Lai's work is supported in part by NSF CAREER Award (DMS—1752934)

References

Debsindhu Bhowmik, Shang Gao, Michael T Young, and Arvind Ramanathan. Deep clustering of protein folding simulations. *BMC bioinformatics*, 19(18):47–58, 2018.

Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. Limp: Learning latent shape representations with metric preservation priors. *arXiv preprint arXiv:2003.12283*, 2020.

D.E. Shaw Research. Molecular dynamics simulations related to sars-cov-2. `http://www.deshawresearch.com/resources_sarscov2.html`, 2020. Accessed: 2020-09-30.

Manfredo P Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Liyu Gong and Qiang Cheng. Exploiting edge features for graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9211–9219, 2019.

Xiaojie Guo, Sivani Tadepalli, Liang Zhao, and Amarda Shehu. Generating tertiary protein structures via an interpretative variational autoencoder. *arXiv preprint arXiv:2004.07119*, 2020.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Frank Noé, Gianni De Fabritiis, and Cecilia Clementi. Machine learning for protein folding and dynamics. *Current Opinion in Structural Biology*, 60:77–84, 2020a.

Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual review of physical chemistry*, 71:361–390, 2020b.

Venkata K. Ramaswamy, Chris G. Willcocks, and Matteo T. Degiacomi. Learning protein conformational space by enforcing physics with convolutions and latent interpolations, 2020.

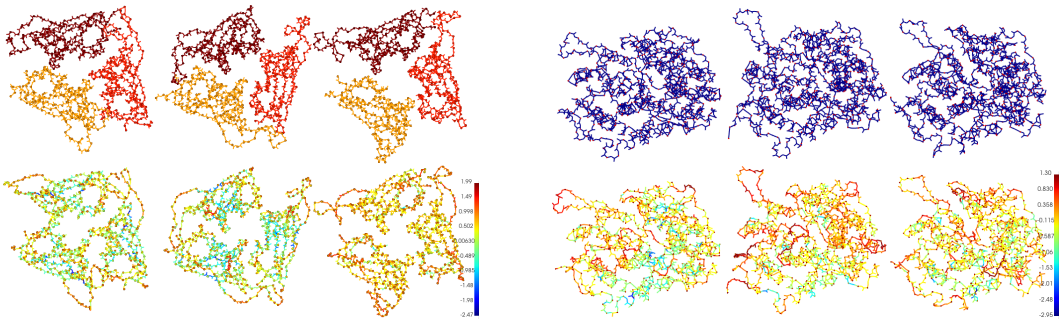
N. Joseph Tatro, Stefan C. Schonsheck, and Rongjie Lai. Unsupervised geometric disentanglement for surfaces via cfan-vae, 2020.

Yasemin Bozkurt Varolgüneş, Tristan Bereau, and Joseph F Rudzinski. Interpretable embeddings from molecular simulations using gaussian mixture variational autoencoders. *Machine Learning: Science and Technology*, 1(1):015012, 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Jie Yang, Kaichun Mo, Yu-Kun Lai, Leonidas J. Guibas, and Lin Gao. Dsm-net: Disentangled structured mesh net for controllable generation of fine geometry, 2020.

A APPENDIX



(a) 3 RBDs of S protein

(b) Ectodomain of Human ACE2

Figure 4: Reconstructions of protein frames from test data using ProGAE. The top row displays the ground truth, while the bottom row displays the corresponding structure generated by the network. Color in the top row denotes separate protein chains, while color in the bottom row indicates the log of atom-wise L_2 error. Color of the bonds indicates the average of the constituent atoms.

Table 4: Results of linear regression on the extrinsic latent space for predicting physical and chemical properties of the drugs that a protein is bound to. Error is normalized for interpretability. For comparison, performance of linear regression on the PCA embeddings of the orientation of the backbone bonds is reported. This embedding is restrained to the same dimension as the latent space.

Dataset		Molecular weight	Hydrogen bond donor count	Topological polar surface area
S protein	PCA error (σ)	0.78 ± 0.00	0.81 ± 0.01	0.79 ± 0.00
	Latent error (σ)	$\mathbf{0.55 \pm 0.04}$	$\mathbf{0.56 \pm 0.03}$	$\mathbf{0.61 \pm 0.00}$
hACE2	PCA error (σ)	0.71 ± 0.00	0.65 ± 0.00	0.73 ± 0.00
	Latent error (σ)	$\mathbf{0.55 \pm 0.01}$	$\mathbf{0.57 \pm 0.01}$	$\mathbf{0.53 \pm 0.02}$

Loss Function The first term in the loss function is a basic reconstruction loss, where $\boldsymbol{P}$ and $\hat{\boldsymbol{P}}$ are taken to be the true and predicted coordinates of the protein backbone atoms. Namely, we evaluate their difference using Smooth-L_1 loss, SL_1. This loss, $SL_1(\boldsymbol{x}, \boldsymbol{y})$, is defined, with $\delta = 2$, as

$$\sum_{i=1}^{\#\boldsymbol{x}} \min\left(\frac{\delta^2}{2}(x_i - y_i)^2, \delta|x_i - y_i| - \frac{1}{2}\right). \tag{2}$$

This loss is less sensitive to outliers (Girshick, 2015).

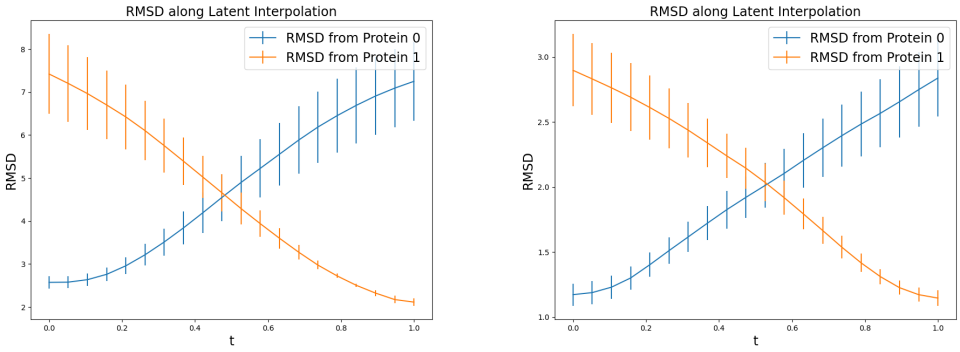
As the reconstruction loss depends on the embedding of the protein in Euclidean space, it may not best measure if intrinsic geometry is faithfully reconstructed. To address this, we consider two encoded proteins with latent codes, $[\boldsymbol{z}_{i,1}, \boldsymbol{z}_{e,1}]$ and $[\boldsymbol{z}_{i,2}, \boldsymbol{z}_{e,2}]$. Then we form a new latent variable,

$$\hat{\boldsymbol{z}}_i = (1-\beta)\boldsymbol{z}_{i,1} + \beta \boldsymbol{z}_{i,2}, \quad \hat{\boldsymbol{z}}_e = \boldsymbol{z}_{e,1}, \quad \beta \sim \mathbf{U}[0,1]. \tag{3}$$

Each of these latent variable decodes to a point cloud $\hat{\boldsymbol{P}}$. We let $Int(\hat{\boldsymbol{E}}_{t,\beta})$, $Int(\hat{\boldsymbol{E}}_{t,1})$, and $Int(\hat{\boldsymbol{E}}_{t,2})$ be the lengths of the pseudobonds of the generated proteins from the interpolated latent code and the two given latent codes. We then introduce a bond length penalty, $\mathcal{R}(\hat{\boldsymbol{P}}_1, \hat{\boldsymbol{P}}_2)$, given by,

$$\mathbb{E}_\beta || Int(\hat{\boldsymbol{E}}_{t,\beta}) - l\left(Int(\hat{\boldsymbol{E}}_{t,1}), Int(\hat{\boldsymbol{E}}_{t,2})\right) ||_1, \tag{4}$$

$$\text{where } l(\boldsymbol{x}, \boldsymbol{y}) = (1-\beta)\boldsymbol{x} + \beta \boldsymbol{y}, \quad \beta \in \mathbf{U}[0,1]. \tag{5}$$



(a) S protein

(b) Human ACE2

Figure 5: RMSD of proteins generated along the latent interpolation between two proteins from different trajectories. The RMSDs are computed with respect to the endpoint proteins, with standard error shown. We see a smooth interpolation between the RMSD errors as desired.

This penalty can be viewed as promoting faithful reconstruction of the pseudobond length between C_α atoms, as well as a smooth interpolation of these lengths along paths in L_I, that is independent of L_E. This penalty is analogous to the metric preservation regularizer introduced in (Cosmo et al., 2020) for 3D meshes. Thus, the loss function $\mathcal{L}((\hat{\boldsymbol{P}}_1, \hat{\boldsymbol{P}}_2), (\boldsymbol{P}_1, \boldsymbol{P}_2))$ for ProGAE is, $\sum_{i=1}^{2} SL_1(\hat{\boldsymbol{P}}_i, \boldsymbol{P}_i) + \lambda_R \mathcal{R}(\hat{\boldsymbol{P}}_1, \hat{\boldsymbol{P}}_2)$.

A.1 Network Hyperparameters

Here we specify the hyperparameters of the networks used in conducting our experiments. Each encoder contains 5 layers with filter sizes, $\{12, 24, 48, 96, 96\}$. The decoder structure is mirrored with filter sizes, $\{128, 128, 64, 32, 16, 3\}$. Each graph attention layer has 4 heads of attention. The dimensions of the intrinsic and extrinsic latent spaces are set to 16 and 32 respectively.

Each Enc_i convolution is taken to have a kernel size of 3 and a stride of 2, being followed with batch normalization layers and ReLU.

For training, we use ADAM with a learning rate of 1E-3 (Kingma & Ba, 2014). Learning rate decays at a rate of 0.995 per epoch. We train models with a weight decay penalty of 5E-5. The models are trained 100 epochs, which is enough to achieve convergence, with a batch size of 64. Additionally, we set $\lambda_{\mathcal{R}} =$ 5E-1 for the bond length penalty.The neighborhood radius for defining the sparsity of the graph attention layer is set to 2.5 Å in the first layer. This radius is scaled at each layer with the stride of the previous convolution.

A.2 Additional Dataset Details

The backbones of the S protein and the hACE2 protein contain 3,690 atoms and 2,386 atoms, respectively. The time resolution is 1,200 ps. We use the first 70% of frames from each trajectory to form the training set. The next 10% and the last 20% of frames form the validation and test sets. The train and test sets are intentionally kept temporally disjoint to better assess generalization.