Contrasting Sequence with Structure: Pre-training Graph Representations with PLMs

Louis Robinson InstaDeep
Timothy Atkinson InstaDeep
Liviu Copoiu InstaDeep
Patrick Bordes InstaDeep
Thomas Pierrot* InstaDeep
Thomas D. Barrett* InstaDeep

{l.robinson,t.atkinson,l.copoiu,p.bordes,t.pierrot,t.barrett}@instadeep.com

Abstract

Understanding protein function is vital for drug discovery, disease diagnosis, and protein engineering. While Protein Language Models (PLMs) pre-trained on vast protein sequence datasets have achieved remarkable success, equivalent Protein Structure Models (PSMs) remain underrepresented. We attribute this to the relative lack of high-confidence structural data and suitable pre-training objectives. In this context, we introduce BioCLIP, a contrastive learning framework that pre-trains PSMs by leveraging PLMs, generating meaningful per-residue and per-chain structural representations. When evaluated on tasks such as protein-protein interaction, Gene Ontology annotation, and Enzyme Commission number prediction, BioCLIP-trained PSMs consistently outperform models trained from scratch and further enhance performance when merged with sequence embeddings. Notably, BioCLIP approaches, or exceeds, specialized methods across all benchmarks using its singular pre-trained design. Our work addresses the challenges of obtaining quality structural data and designing self-supervised objectives, setting the stage for more comprehensive models of protein function. Source code is publicly available[2].

1 Introduction

The study of proteins, central to cellular function, impacts fields such as medicine, biotechnology, and computational biology. While the amino acid sequence of a protein carries vital information, its 3D structure often holds the key to understanding its function and putative interactions. Machine learning, especially Protein Language Models (PLMs), has recently revolutionized protein modelling. PLMs pre-trained on extensive sequence data have demonstrated the ability to capture intrinsic relationships in amino acid sequences [1] in rich representations that can be leveraged for various downstream applications [2] - mirroring the trends observed in other domains such as natural language processing [3, 4] and computer vision [5]. However, despite the success of machine learning in protein structure prediction, epitomized by AlphaFold [6], general pre-trained protein structure models (PSMs) have not yet found the same ubiquitous application as their sequence-based counterparts. We attribute this to two main challenges; (i) data scarcity and (ii) objective complexity.

High-quality protein structure data is hard to come by and often expensive. Methods such as X-ray crystallography and cryo-electron microscopy, though very insightful, are not without limitations

*Equal supervision.
[2]https://github.com/instadeepai/bioclip

Machine Learning for Structural Biology Workshop, NeurIPS 2023.

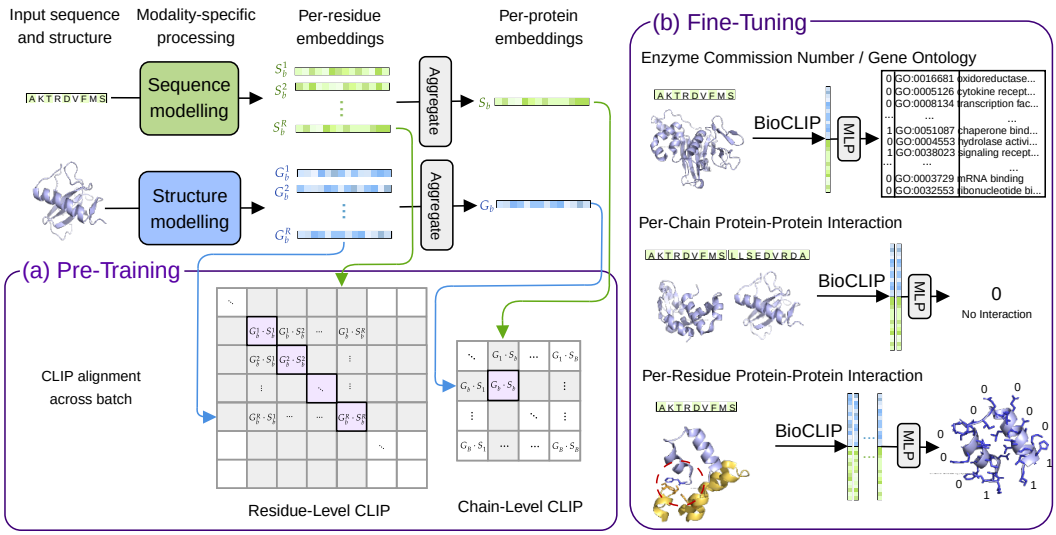

Figure 1: Pre-training, fine-tuning and downstream task illustrations. The modules in light blue are tuned, the modules in green are fixed.

[7]. Whilst recent tools such as AlphaFold and ESMFold have enabled the generation of massive protein structure datasets, they only predict the 3D coordinates, and can still be imperfect especially for multi-state proteins or proteins with shallow MSAs [8, 9]. However, objective formulation remains a significant hurdle. While masked sequence prediction has proven highly effective for pre-training PLMs, defining self-supervised objectives for structure data is far more challenging due to the continuous and multi-dimensional nature of protein structures.

Motivated by this, we introduce BioCLIP, a self-supervised contrastive learning framework for building latent representations of protein structures and sequences. The core intuition behind BioCLIP is to leverage the quality of pre-trained PLM embeddings trained on abundant sequence data to facilitate the training of a PSM. The model employs a loss function inspired by Contrastive Language–Image Pretraining (CLIP) [10], incorporating both per-residue and per-chain embeddings to create a comprehensive representation of protein structures.

We validate BioCLIP's efficacy through tests on protein-protein interaction prediction, GO-term annotation, and Enzyme Commission number prediction. Our findings underscore three points: (1) BioCLIP's pre-trained Graph Neural Network (GNN) surpasses conventional training methods, (2) structural embeddings enhance sequence embeddings and usually boost performance when combined, and (3) BioCLIP approaches or outperforms specialized methods.

2 BioCLIP

BioCLIP is visualised in Figure 1. The core idea is to pass as input both a sequence and a structure representation of a given protein through a PLM and a PSM, respectively, to obtain per-residue and per-protein level embeddings. For a given sequence embedding S_a and structure embedding G_b, the scaled cosine distance metric can be used to measure the similarity of the embeddings,

$$d\left(S_a, G_b\right) = \frac{S_a \cdot G_b}{||S_a|| \cdot ||G_b||}. \tag{1}$$

For a batch of sequence embeddings $S = \{S_1 \dots S_N\}$ and structure embeddings $G = \{G_1 \dots G_N\}$ where S_i corresponds to G_i, a CLIP-style loss can be employed,

$$L_S = -\frac{1}{N} \sum_{a=1}^{N} \log \frac{e^{d(S_a, G_a)}}{\sum_{b=1}^{N} e^{d(S_a, G_b)}}, \tag{2}$$

such that by minimising the contrastive loss term for a given batch of proteins, the model is trained to produce aligned embeddings for paired sequence-structure inputs, which are far away from all other embeddings produced by the model. An equivalent method is used at the level of per-residue embeddings across all proteins in the batch. These learned embeddings can then be used for a variety of down-stream tasks; for example, for a given protein with sequence and structure representation, the outputs of the PSM and the PLM can be concatenated and passed through an MLP to predict whether that protein has a given Enzyme Commission number.

In principle, while a separate BioCLIP model can be trained end-to-end at the level of both proteins and their component amino acids, these tasks are inherently related. Each structure or sequence is by definition a composition of its component residue nodes and amino acid types, respectively. There is a vast literature demonstrating that training a single backbone model for multiple, related, tasks typically yields better generalisation performance [11]. It is therefore beneficial to treat both sequence and amino-acid representations as heads of the same underlying sequence encoder, and similarly treat structure and residue representations as heads of the same underlying structure encoder. Then by minimising the sum of contrastive losses, we simultaneously minimise both the protein-level loss and the residue-level loss and benefit from the commonalities in the two problems.

We leverage the available massive pre-trained PLMs which already provide robust representations of the protein sequences, e.g. [12, 13, 14, 15, 16]. By appropriately choosing a pre-trained PLM, we are able to fix the sequence and amino-acid representations and then derive robust structural representations from limited sequence-structure pairs, reflecting the availability of substantially larger datasets of protein sequences. In our experiments, we use an instance of ESM [17] which is a state-of-the-art BERT-style PLM. The PSM used is a type of SE(3)-invariant graph neural network based on prior work [18]. By design, all node and edge features passed to the network are SE(3)-invariant, and any message passing applied on top of them inherently maintains this. Within the message passing mechanism, we use a type of graph attention network [19] with multi-head dot-product attention [20].

BioCLIP is pre-trained on a dataset of approximately 500,000 sequence-structure pairs obtained from the RCSB PDB databank [21]. Once the model has been pre-trained it can then be used for a variety of downstream tasks, such as those described in Section 3. Full details of the implementation used here are given in Appendix B.

3 Experiments

3.1 Tasks

To investigate whether BioCLIP is capable of learning meaningful structural representations that offer novel benefits on top of those already available from the underlying PLM, we empirically evaluate their performance when used as a basis for three downstream tasks. These tasks are visualised in Figure 1. Full details of the downstream tasks and their configurations are provided in Appendix C.

- **Function Prediction:** A binary protein classification task, based on datasets used in [22], where the goal is to predict enzyme-commission numbers and three gene ontology (GO) tasks: biological-process (BP), molecular-function (MF) and cellular component (CC).
- **Protein-Protein Interaction:** A binary classification task where, given two proteins, the objective is to predict whether or not they interact. We study the Human and S.cerevisiae tasks that are introduced in [23].
- **Per-Residue Protein-Protein Interaction:** A binary classification task where, given two biological sub-units within two distinct biological molecules, the objective is to predict whether or not they interact. This task is taken from [24].

3.2 Models

Across all tasks we utilise the same BioCLIP pre-trained GNN. During fine-tuning we take the structure representations obtained from the GNN *before* the application of a final MLP, such that initially, the structure representation differs from the sequence representation by a non-linear transformation. These structure embeddings are then concatenated with sequence embeddings from ESM and passed into an (initially untrained) predictor MLP model. The parameters of the GNN the final three layers of the ESM model and the final MLP are all fine-tuned for each specific task.

Ablations To ablate BioCLIP's component parts and identify their contributions, we consider four variants of the method in fine-tuning:

- **GNN (random)**: The same GNN architecture as used in our BioCLIP pre-trained, but initialised with random weights. This variant is used as a control to measure the effectiveness of pre-training of the GNN. All parameters in the GNN and the final MLP are fine-tuned.
- **GNN (pretrained)**: The GNN architecture, pre-trained to align to the ESM model. This variant is used to measure the effectiveness of pre-training the structure model in isolation. Again, all parameters in the GNN and final MLP are fine-tuned.
- **ESM**: The PLM model used in pre-training, with its final three layers fine-tuned. This variant is used as a control to measure the effectiveness of introducing the BioCLIP-aligned GNN for downstream tasks. All parameters in the last three layers of the ESM model and the final MLP are fine-tuned. Note that we did experiment with different methods of fine-tuning but found that it made little difference to the final performance.

Task-specific Baselines For each of the downstream tasks, we further identify a recent method that represents the state-of-the-art, or close, for that task:

- **DeepFRI [22] (GO-term tasks)**: This model represents the protein structure as contact maps, and employs a frozen protein language model to provide input node features for a three-layer graph convolutional network [25], which is pre-trained on 10 million Pfam protein sequences and uses an LSTM architecture with 512 hidden units. A sum operation is used to pool the per-residue representations into a single protein representation, which is finally passed through an MLP. The authors expand the training dataset by using homology models from SWISS-MODEL which they show boosts performance significantly, the dataset uses 30k non-redundant experimental PDB structures and 220k non-redundant homology models from SWISS-MODEL.
- **Jha et. al. [23] (protein-protein tasks)**: The authors use a GCN and a GAT to predict interactions between proteins. They use two pre-trained protein language models: SeqVec (LSTM) and ProtBert (Transformer) to obtain feature vectors for each residue. The SeqVec embedder produces a sequence representation by summing three representations: a 1-character convolution (CharNN), and bi-directional LSTM layers. The second PLM, ProtBert, is a BERT model is trained on the BFD-100 dataset [26], which has 2.1 million protein sequences. Finally, the per protein representation is obtained by averaging over residue activations. The PPI datasets used are from two organisms: Human and S. cerevisiae. The Pan's human dataset [27] is modified to remove duplicates and apply some filtering, see the original paper for details [23].
- **PeSTo [24] (per-residue protein interaction)**: The authors employ a deep GAT network on the atoms of a protein structure. Latent atom representations are obtained through 32 SO-3 invariant message-passing layers with increasing neighborhood sizes. Finally, cross attention is used from the atoms in each residue, to the corresponding residue.

3.3 Results

The results of the downstream evaluation experiments are summarised in Figure 2, with corresponding exact numerical values found in Appendix D. For each experimental configuration, we present results after 1, 2 and 3 epochs and also the final epoch from each fine-tuning experiment. From our experiments, we draw three important conclusions:

1. **BioCLIP's pre-trained structure representations are informative :** Observing the results for GNN (random) and GNN (pretrained), we find that for all 7 tasks and at every epoch, the pre-trained GNN provides better performance. This result is particularly stark in the Protein-Protein Interaction tasks, where the pre-trained GNN significantly outperforms the randomly initialised GNN in early epochs. From these observations, we conclude that pre-training a GNN via the BioCLIP method can result in meaningful representations of protein structure that can aid in downstream tasks.
2. **Aligned pre-trained structure and sequence representations are additive:** We find that in 25/28 cases, the results for fine-tuning of the full BioCLIP system outperforms the ESM model alone, and in 23/28 cases the pre-trained GNN alone. We argue that, although the GNN and ESM models are aligned, they provide different inductive biases which are mutually beneficial

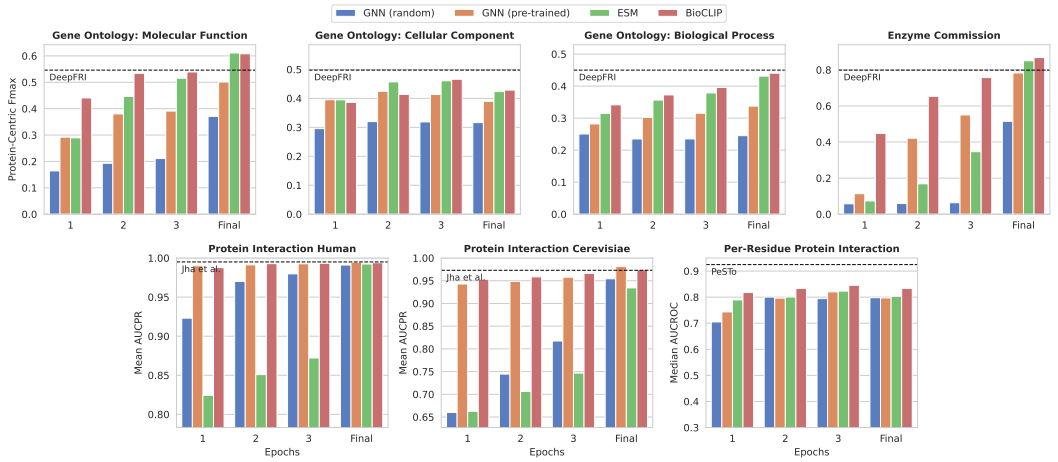

Figure 2: Performance metrics across 7 downstream tasks for: Random GNN, pre-trained GNN with the contrastive BioCLIP objective, ESM 150M with the last three layers tuned, and the BioCLIP model which combines ESM and the pre-trained GNN. For each task, we also provide a recent, task-specific, benchmark represented as a dashed line.

for fine-tuning. We note that in the cases where we did not observe a benefit in combining the representations, the drop in performance was very marginal.

3. **BioCLIP is competitive with state-of-the-art methods:** In 4 of the 7 problems considered, we find that BioCLIP is able to out-perform or match a recent, state-of-the-art, method designed specifically for that task. In the GO: Molecular Function and Enzyme Commission problems, fine-tuning BioCLIP surpasses the results achieved by DeepFRI. Further, on the GO: Cellular Component and GO: Biological Process tasks, BioCLIP's performance reaches within 7% of the performance achieved by DeepFRI. For the two PPI tasks, we find that BioCLIP is able to match the performance of Jha et. al. Additionally, we find that BioCLIP is able to reach very close to the performance of Jha et. al. within only a few epochs. We also note that while BioCLIP struggled to reach the performance of PeSTo on per-residue protein interaction, it did outperform all of the methods that PeSTo was compared to in its original publication [24][3].

4 Conclusion & Future Work

This work introduces BioCLIP, a contrastive learning framework for learning structure and sequence representations of proteins, which we have demonstrated in a practical setting by aligning a GNN to a pre-trained PLM. We have carried out numerous empirical evaluations on a variety of downstream tasks for protein function prediction. We show that the representations derived from BioClip are meaningful, complementary to existing sequence embeddings and can be used to obtain competitive performance in comparison to task-specific methods. Overall, we believe that BioCLIP addresses the problem of limited structural data for pre-training in a systematic way, and provides a general template for models of protein function based on their full sequential and structural representations.

There are a number of areas for future work further developing the ideas presented here. Firstly, recent work [28] has demonstrated that a CLIP-style model can be effectively trained using a sigmoid loss, rather than a softmax loss, which may pave the way for training of BioCLIP with substantially larger batch sizes. An empirical investigation in this direction may yield improved performance on downstream tasks. We also note that the PSM used in this work was substantially smaller than the PLM, which may limit the richness of the structural embedding obtained. This reflects the broader problem within the geometric deep learning community of over smoothing in deeper networks [29]. Investigation into alternative models of structure, such as the EvoFormer module used in AlphaFold2 [6] and may therefore allow for larger, richer representations of structure.

[3]We performed a preliminary experiment pre-training a small version of the PeSTo architecture and fine-tuned it on the per-residue PPI task. As we found that we achieved similar performance to [24], we hypothesise that the difference in performance can be attributed to the PSM architecture and use of the all-atom representation.

References

[1] Dan Ofer, Nadav Brandes, and Michal Linial. The language of proteins: Nlp, machine learning & protein sequences. *Computational and Structural Biotechnology Journal*, 19:1750–1758, 2021.

[2] Tristan Bepler and Bonnie Berger. Learning the protein language: Evolution, structure, and function. *Cell systems*, 12(6):654–669, 2021.

[3] OpenAI. Gpt-4 technical report, 2023.

[4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

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

[6] 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.

[7] Radostin Danev, Haruaki Yanagisawa, and Masahide Kikkawa. Cryo-electron microscopy methodology: current aspects and future directions. *Trends in biochemical sciences*, 44(10):837–848, 2019.

[8] Anastassis Perrakis and Titia K Sixma. Ai revolutions in biology: The joys and perils of alphafold. *EMBO reports*, 22(11):e54046, 2021.

[9] Felix Wong, Aarti Krishnan, Erica J Zheng, Hannes Stärk, Abigail L Manson, Ashlee M Earl, Tommi Jaakkola, and James J Collins. Benchmarking alphafold-enabled molecular docking predictions for antibiotic discovery. *Molecular Systems Biology*, 18(9):e11081, 2022.

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[11] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[12] Yanbin Wang, Zhu-Hong You, Shan Yang, Xiao Li, Tong-Hai Jiang, and Xi Zhou. A high efficient biological language model for predicting protein–protein interactions. *Cells*, 8(2):122, 2019.

[13] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

[14] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[15] Robert Verkuil, Ori Kabeli, Yilun Du, Basile IM Wicky, Lukas F Milles, Justas Dauparas, David Baker, Sergey Ovchinnikov, Tom Sercu, and Alexander Rives. Language models generalize beyond natural proteins. *bioRxiv*, pages 2022–12, 2022.

[16] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Hassan Sirelkhatim, Guillaume Richard, et al. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. *bioRxiv*, pages 2023–01, 2023.

[17] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019.

[18] Octavian-Eugen Ganea, Xinyuan Huang, Charlotte Bunne, Yatao Bian, Regina Barzilay, Tommi S. Jaakkola, and Andreas Krause. Independent se(3)-equivariant models for end-to-end rigid protein docking. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

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

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

[21] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000.

[22] 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):3168, 2021.

[23] Kanchan Jha, Sriparna Saha, and Hiteshi Singh. Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, 12(1):1–12, 2022.

[24] Lucien F. Krapp, Luciano A. Abriata, Fabio Cortés Rodriguez, and Matteo Dal Peraro. Pesto: parameter-free geometric deep learning for accurate prediction of protein binding interfaces. *Nature Communications*, 14(1):2175, Apr 2023.

[25] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[26] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature communications*, 9(1):2542, 2018.

[27] Xiao-Yong Pan, Ya-Nan Zhang, and Hong-Bin Shen. Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features. *J Proteome Res*, 9(10):4992–5001, October 2010.

[28] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023.

[29] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3438–3445, 2020.

[30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[31] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[32] Henriette Capel, Robin Weiler, Maurits Dijkstra, Reinier Vleugels, Peter Bloem, and K Anton Feenstra. Proteinglue multi-task benchmark suite for self-supervised protein modeling. *Scientific Reports*, 12(1):16047, 2022.

[33] Richard Michael, Jacob Kæstel-Hansen, Peter Mørch Groth, Simon Bartels, Jesper Salomon, Pengfei Tian, Nikos S Hatzakis, and Wouter Boomsma. Assessing the performance of protein regression models. *bioRxiv*, pages 2023–06, 2023.

[34] Karim Beguir, Marcin J Skwark, Yunguan Fu, Thomas Pierrot, Nicolas Lopez Carranza, Alexandre Laterre, Ibtissem Kadri, Abir Korched, Anna U Lowegard, Bonny Gaby Lui, et al. Early computational detection of potential high-risk sars-cov-2 variants. *Computers in biology and medicine*, 155:106618, 2023.

[35] Fang Wu, Lirong Wu, Dragomir Radev, Jinbo Xu, and Stan Z Li. Integration of pre-trained protein language models into geometric deep learning networks. *Communications Biology*, 6(1):876, 2023.

[36] 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.

[37] David Lee, Oliver Redfern, and Christine Orengo. Predicting protein function from sequence and structure. *Nature reviews molecular cell biology*, 8(12):995–1005, 2007.

[38] Kristof T Schütt, Huziel E Sauceda, P-J Kindermans, Alexandre Tkatchenko, and K-R Müller. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24), 2018.

[39] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pages 9377–9388. PMLR, 2021.

[40] 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):2453, 2022.

[41] Jianwen Chen, Shuangjia Zheng, Huiying Zhao, and Yuedong Yang. Structure-aware protein solubility prediction from sequence through graph convolutional network and predicted contact map. *Journal of cheminformatics*, 13(1):1–10, 2021.

[42] Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023.

[43] 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*, 50(D1):D439–D444, 2022.

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

[45] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33:5812–5823, 2020.

[46] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 594–604, 2022.

[47] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference 2022*, pages 1070–1079, 2022.

[48] 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.

[49] Yuyang Wang, Jianren Wang, Zhonglin Cao, and Amir Barati Farimani. Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.

[50] Can Chen, Jingbo Zhou, Fan Wang, Xue Liu, and Dejing Dou. Structure-aware protein self-supervised learning. *Bioinformatics*, 39(4):btad189, 2023.

[51] Chenguang Zhao, Tong Liu, and Zheng Wang. PANDA2: protein function prediction using graph neural networks. *NAR Genomics and Bioinformatics*, 4(1):lqac004, 02 2022.

[52] Amelia Villegas-Morcillo, Stavros Makrodimitris, Roeland CHJ van Ham, Angel M Gomez, Victoria Sanchez, and Marcel JT Reinders. Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics*, 37(2):162–170, 2021.

[53] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics*, 20(1):1–17, 2019.

[54] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[55] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

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

[57] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[58] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[59] Thomas D Barrett, Amelia Villegas-Morcillo, Louis Robinson, Benoit Gaujac, David Admete, Elia Saquand, Karim Beguir, and Arthur Flajolet. So manyfolds, so little time: Efficient protein structure prediction with plms and msas. *bioRxiv*, pages 2022–10, 2022.

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

[61] P. Gainza, F. Sverrisson, F. Monti, E. à, D. Boscaini, M. M. Bronstein, and B. E. Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nat Methods*, 17(2):184–192, Feb 2020.

[62] 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*, pages 15272–15281, 2021.

Appendix

A Related Work

Protein Language Models Recent advances in scaleable transformer architectures [20] have facilitated an explosion in pre-trained language models, e.g. [30, 31, 4]. Correspondingly, a number of pre-trained *protein language models* have emerged, employing the same principles of auto-regressive or masked-prediction while targeting the vast available protein sequence data. For example, the ESM family of BERT-like models are trained to do masked prediction [17] which can be used for a variety of downstream tasks [32, 33, 34], ProtGPT2 is an autoregressive model capable of generating novel, realistic protein sequences [14]. While these PLMs in many ways represent a state-of-the-art, the inclusion of additional structural information may improve performance in practice [35].

Protein Structure Models Protein structure is a key modality in modelling of protein function, with a variety of research interest in both predicting protein structure [6, 36] and leveraging predicted or experimentally obtained structural information for protein modelling [37]. With the view that protein structure, represented as a set of residue coordinates in 3D space, can directly be mapped to a graph structure, there is a clear affinity with the subject of graph neural networks. In particular, a number of GNN architectures have emerged in the past decade which specifically target invariance and equivariance with respect to the 3D coordinate system [38, 39, 40], some of which have shown promise in protein tasks such as prediction of solubility [41], function [22] and binding affinity [24]. However these *protein structure models* (PSMs) face bespoke challenges, such as the well-studied 'over-smoothing' problem for GNNs [29] that limits the size and depth of these models. Additionally, there is a relative lack of available structural data; consider that at the time of writing, the Protein Data Bank [21] has hundreds of thousands of experimentally obtained structures, in comparison to the millions of available protein sequences [42], although we note that this issue can increasingly be mitigated using predicted structures e.g. [43, 44].

Pre-trained Structure Models Recently, a variety of techniques for contrastive learning on graph representations have been proposed. Typically, these incorporate some form of structural augmentation [45], structure masking [46] or network perturbation [47] to create neighborhoods of structure representations for contrastive learning. Of particular relevance to this work, GearNet [48] and MolCLR [49] propose graph-augmentation approaches to contrastive learning of protein and molecule structures, respectively. [50] propose a mask-prediction method where a GNN is trained to reconstruct pairwise distances and angles between residues. However, as these techniques focus on contrastive learning between graph structures, they are, in isolation, unable to leverage the vast available protein sequence data to improve their representations.

Multi-modal Protein Embeddings As both sequence and structure are considered key modalities for modelling protein function, a number of works naturally consider combined representations for downstream tasks. For example, a number of works aims for a 'best of both worlds' by incorporating amino acid embeddings, obtained from a pre-trained PLM, as node features to a PSM that is then trained to predict protein function [51, 23]. However, these approaches do not incorporate unsupervised learning into their PSM components. The approach taken in this work can be motivated by [52], which presents experimental results showing that representations of sequences obtained via unsupervised learning, specifically with the ELMo model [53], are more effective in downstream tasks than hand-crafted representations. Further motivating the unsupervised learning of complimentary structural representations, [53] find that although the embeddings are obtained from sequence alone, they do not benefit from including hand-crafted structural representations. In contrast to these directions, a number of works provide avenues for multi-modal representations through large-scale contrastive learning, particularly in the case of image-text modalities [10, 54]. These methods provide a systematic way to build multi-modal representations of data, which we leverage here, alongside the ubiquitous success of PLMs, to achieve effective pre-training of data-scarce PSMs.

B BioCLIP Implementation

B.1 Loss

BioCLIP is inspired by the Contrastive Language-Image Pre-training (CLIP) method. Consider a batch of n paired protein sequences and structures. The goal of BioCLIP is to learn a latent representation of each sequence, S_i, and structure G_j so that the scaled cosine similarity,

$$d\left(S_i, G_j\right) = \frac{S_i \cdot G_j}{||S_i|| \cdot ||G_j||} \tag{3}$$

is maximised for paired sequences and structures, $i = j$, and minimised otherwise, $i \neq j$. This can be achieved by minimising the symmetric cross-entropy loss, with a term L_S considering the discriminative power of the cosine similarity across all structures in the batch for each sequence, and similarly a term L_G considering the discriminative power of the cosine similarity across all sequences in the batch for each structure,

$$L_S = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{e^{\tau d(S_i, G_i)}}{\sum_{j=1}^{n} e^{\tau d(S_i, G_j)}}, \tag{4}$$

$$L_G = -\frac{1}{n}\sum_{i=1}^{n} \log \frac{e^{\tau d(S_i, G_i)}}{\sum_{j=1}^{n} e^{\tau d(S_j, G_i)}}, \tag{5}$$

where in both cases, $\tau \in \mathbb{R}^+$ is a learned 'temperature' controlling the sharpness of the softmax operator. Then the total 'protein-level' loss to be minimised is,

$$L_P = \frac{L_S + L_G}{2}. \tag{6}$$

Additionally, we are interested in learning aligned latent representations at a more granular level, as many downstream tasks consider per-residue function and interaction [22, 24]. Consider that each protein sequence is an ordered sequence of m amino acids, $a_{i,1} \ldots a_{i,m}$, and, for each amino acid $a_{i,j}$ in that sequence, there exists a corresponding node $v_{i,j}$ within the structure's set of nodes $V_i = \{v_{i,1} \ldots v_{i,m}\}$. Then, as with the aligned protein-level representations S_i, G_i, we also desire aligned residue-level representations S_i^j, G_i^j that, while conditioned on their global contexts S_i, G_i are themselves latent representations of their corresponding amino acids and nodes. Then the symmetric cross-entropy loss is defined equivalently across the entire batch of residues,

$$L_A = -\frac{1}{n}\sum_{i=1}^{n} \frac{1}{m_i}\sum_{j=1}^{m_i} \log \frac{e^{\tau d\left(S_i^j, G_i^j\right)}}{\sum_{k=1}^{n}\sum_{l=1}^{m_k} e^{\tau d\left(S_i^j, G_k^l\right)}}, \tag{7}$$

$$L_V = -\frac{1}{n}\sum_{i=1}^{n} \frac{1}{m_i}\sum_{j=1}^{m_i} \log \frac{e^{\tau d\left(S_i^j, G_i^j\right)}}{\sum_{k=1}^{n}\sum_{l=1}^{m_k} e^{\tau d\left(S_k^l, G_i^j\right)}}, \tag{8}$$

and the total 'residue-level' loss to be minimised is,

$$L_R = \frac{L_A + L_V}{2}. \tag{9}$$

As we train a single model for both protein-level and residue-level contrastive learning, the overall loss term is simply,

$$L = L_P + L_R. \tag{10}$$

We did not find that it was necessary to introduce different weightings for the loss terms.

B.2 Protein Language Model

We consider the `esm2_t30_150M_UR50D` instance of Evolutionary Scale Modeling (ESM) [15], which is a BERT-style [30] transformer architecture consisting of 30 layers and 150 million total parameters. We use the publicly available weights for this model, which are pre-trained to perform masked amino-acid prediction on 65 million unique protein sequences taken from the UniRef protein sequence database [55].

Each amino acid is represented as a token in an ordered protein sequence which is passed through a encoder-only transformer architecture. After processing by the ESM, the latent representation S_i^j of each amino acid token $a_{i,j}$ is obtained from the final layer. The representation of the entire protein sequence S_i is obtained by averaging over the amino acids' embeddings,

$$S_i = \frac{1}{m_i} \sum_{j=1}^{m_i} S_j^i \tag{11}$$

B.3 Protein Structure Model

SE(3)-invariance We employ a SE(3)-invariant graph neural network based on prior work [18]. Each protein p is represented as a graph $G = (V, E, f_V, f_E)$ where V is the set of nodes, E is the set of edges, $f_V \in \mathbb{R}^{|V| \times N}$ is the initial node features in dimension N and $f^E \in \mathbb{R}^{|E| \times M}$ is the initial edge features in dimension M. The set of edges E is constructed by taking each node's k-nearest neighbours in euclidean space, and each edge $e \in E$ has an associated source node $s(e) : E \to V$ and target node $t(e) : E \to V$. The neighborhood $\mathcal{N}_v \subseteq E$ of a node v is therefore defined,

$$\forall v \in V, \mathcal{N}_v = \{e \in E \mid s(e) = v\} \tag{12}$$

The initial node features $f_V^1 \dots f_V^{|V|}$ and edge features $f_E^1 \dots f_E^{|E|}$ contain information about the local geometry, and are SE(3)-invariant. After this, any conventional message passing system can be employed without breaking this invariance. Every node/residue v_i has a 3D coordinate z_i – chosen as the coordinate of the α-carbon atom – and a feature vector f_V^i. There are several possibilities to define the feature vector f_V^i (these can be combined via concatenation). In our experiments, we choose the following:

- A one-hot encoding of the type of amino acid residue. This one-hot encoding serves as input of an embedding layer that is learned during the training phase.
- A sinusoidal encoding of sequence position. As in [56], the order of nodes is fixed to the corresponding sequence order. For the i-th residue, the positional encoding is $(\phi(i, 1)...\phi(i, D))$ where D is the embedding size.

Edge features are defined following the same method as [18]. For each residue v_i, a local coordinate system is formed by (a) the unit vector t_i pointing from the α-carbon atom to the nitrogen atom, (b) the unit vector u_i pointing from the α-carbon to the carbon atom of the carboxyl (-CO-) and (c) the normal of the plane defined by t_i and u_i: $n_i = \frac{u_i \times t_i}{\|u_i \times t_i\|}$. Finally, setting: $q_i = n_i \times u_i$, the edge features are then defined as the concatenation of the following:

- relative position edge features: $p_{j \to i} = \left(n_i^T u_i^T q_i^T\right)(x_j - x_i)$
- relative orientation edge features: $q_{j \to i} = \left(n_i^T u_i^T q_i^T\right) n_j$, $k_{j \to i} = \left(n_i^T u_i^T q_i^T\right) u_j$, $t_{j \to i} = \left(n_i^T u_i^T q_i^T\right) v_j$
- distance-based edge features, defined as radial basis functions: $f_{j \to i, r} = e^{-\frac{\|x_j - x_i\|^2}{2\sigma_r^2}}, r = 1, 2...R$ where $R = 15$ and $\sigma_r = 1.5$

Message Passing We use a type of graph attention network [19] described as follows. Consider hidden node features at layer l, $x_l^1 \dots x_l^{|V|}$ where for $l = 0$, these correspond to the initial SE(3)-invariant node features e.g. $x_0^v = f_V^v, \forall v \in V$. Then each hidden node representation undergoes layer normalisation with affine parameters [57],

$$\forall v \in V, y_l^v = \texttt{layer\_norm}\left(x_{l-1}^v\right), \tag{13}$$

For node v's neighborhood $\mathcal{N}_v$, we construct a representation of each outgoing edge e by concatenating its original features with the normalised hidden features of the edge's target node $t(e)$,

$$\forall e \in E, x_l^e = f_E^e \parallel y_l^{t(e)}, \tag{14}$$

where $a \parallel b$ denotes concatenation of vectors a and b. Multi-head dot-product attention [20] with H heads is applied over the neighborhood, where the output of the hth head is computed,

$$n_{l,h}^v = \sum_{e \in \mathcal{N}_v} \frac{\exp\left(W_{l,h}^q y_l^v \cdot W_{l,h}^k x_l^e\right)}{\sum_{e' \in \mathcal{N}_v} \exp\left(W_{l,h}^q y_l^v \cdot W_{l,h}^k x_l^{e'}\right)} W_{l,h}^x x_l^e, \tag{15}$$

and $W_{l,h}^q, W_{l,h}^k, W_{l,h}^x$ are the learned query, key and value projections, respectively. Finally, a residual connection [58] is used to ensure effective gradient propagation,

$$\forall v \in V, x_l^v = \phi_l(n_l^v) + x_{l-1}^v \tag{16}$$

where ϕ is a simple MLP with a single hidden layer and layer normalisation applied at its input, and n_l^v is obtained by concatenating the outputs of the H heads, $n_{l,1}^v \ldots n_{l,H}^v$.

Latent Representation After L steps of message passing, 2-layer MLP ϕ_v is used to map each node/residue into the PLM embedding dimension to obtain the per-residue representation,

$$\forall v \in V, G^v = \phi_v\left(x_L^v\right) \tag{17}$$

To obtain the representation of the structure, we employ cross attention mechanism to reduce the nodes' representations into a single vector. A single learned vector takes the role of the query $W_{l,h}^q y_l^v$ in equation 15, giving a reduced representation of the structure after L layers of message passing, G_L. Finally, an MLP ϕ_S with two hidden layers, is used to map the reduced representation to the PLM embedding dimension to obtain the per-structure representation,

$$G = \phi_S(G_L) \tag{18}$$

B.4 Pre-training

Data The BioCLIP GNN is pre-trained by processing all mmcif files in the Protein Data Bank (PDB) [21] with a cutoff before 2020-05-14, resolution < 9Å, and no single amino acid accounting for more than 80% of the sequence [59]. This results in around 490 k structures.

Dataloader We found we could get a boost in performance by sampling the batch based on sequence similarity. Specifically, we hypothesised that the majority of protein chains will be completely different and very easy to satisfy the pre-training loss. To force the batch to have more similar sequences we clustered our pre-training data at 50% and minimum sequence coverage of 0.8, then we sample clusters for the batch proportional to the square root of the cluster size, and take four samples uniformly at random within each cluster.

Hyperparameters The BioCLIP GNN is trained for 150 thousand steps with a batch size of 128 protein sequences. A loss is computed across representations S_i, G_j for all 128 proteins in the batch. We consider all residues in the batch, and sample 2048 of them uniformly at random, without replacement, for the purposes of approximating the loss for contrasting amino acid-node representations $a_{i,j}, v_{k,l}$. We use the Adam optimiser [60] with a learning rate of 10^{-4} and default values for β_1, β_2. The GNN model has 3 layers of message passing with a hidden node representation dimension of 512. Each GAT layer is followed by an MLP which has input and output of size 512

and one hidden layer of size 2048. The per-chain and per-residue structure representation has an MLP which input size 512, two hidden layers of size 1024 and output size 640 to match ESM. All MLPs have the ReLU activation function.

B.5 Fine-tuning

The initialisation of the full BioCLIP model for fine-tuning requires loading the pre-trained GNN parameters but discarding the final MLP, concatenating with the ESM embeddings then adding a randomly initialised MLP. This MLP has two hidden layers of size $h + c$ and an output size of c which is the number of categories for that task, h is the hidden size of the GNN, $h = 512$. We use the per-residue or per-chain representation depending on the task. For each task the batch size is 32 and we use the adamw optimiser with learning rate and weight decay equal to 10^{-4} (weight decay is 0 for the per-residue PPI task). We train GO/EC for 20 epochs, PPI for 30 epochs and per-residue PPI for 5 epochs.

C Downstream Tasks

Function Prediction To evaluate BioCLIP in the context of protein function prediction, we consider four multiple binary classification tasks: enzyme-commission numbers (EC) and three gene ontology (GO) tasks: biological-process (BP), molecular-function (MF) and cellular component (CC). All datasets are the same as in [22] and, in the case of comparison to DeepFRI, we recompute all methods' performances on identical test sets. The datasets are divided as follows: train/validation/test datasets are made up of 27581 / 3061 / 2991 examples for GO tasks and 15035 / 1665 / 1840 for the EC task. The number of terms in each task is as follows: BP 1943, MF 489, CC 320 and EC 538. Class imbalance is quite severe, the median positive class percentage across terms is BP 0.122%, CC 0.116% MF 0.105% and EC 0.057%. As each task is itself a collection of binary classification tasks, the loss function used is binary cross entropy, averaged across all terms. We evaluate performance with two criteria: (1) for a given term, how well does the model classify the proteins which have that term; (2) for a given protein, how good is the model at classifying which terms are positive. The reason we opted for two different metrics is due to the convoluted design of the GO nomenclature, where one protein sequence can have multiple GO terms. With the above mentioned metrics we will asses (1) which GO terms are easier to predict and (2) the degree to which we can comprehensively annotate test sequences.

Protein-Protein Interaction We evaluate the learned BioCLIP representations on two benchmarks of protein-protein interaction introduced in [23]. The benchmarks Human and S.cerevisiae are composed of pairs of proteins annotated with a 0/1 label indicating whether proteins interact. The dataset contains positive protein-protein interaction pairs from the human protein reference database (HPRD) and the Database of Interacting Proteins (DIP) for humans and S. cerevisiae, respectively. After preprocessing steps, such as eliminating duplicates and removing proteins with fewer than 50 amino acids, the dataset has 37K and 17K interacting pairs for humans and S. cerevisiae. Negative instances, representing non-interacting protein pairs, are generated based on subcellular localization differences, with totals of 36,323 and 48,594 pairs for each organism, and homologous pairs are filtered using the CD-HIT tool at a 40% sequence identity cutoff. In total we have 22K in human and 9K for cerevisiae after filtering lengths below 1024. The datasets are relatively balanced: for the human targets there is 73% in the positive class, for cerevisiae there is 50%. The loss function in both tasks is cross entropy. We use the same metric as our baseline for this task [23] to be able to directly compare to BioCLIP. AUCPR is a reasonable metric for this task as it is robust to the relatively significant class in-balance that is present.

Per-Residue Protein-Protein Interaction A formulation of a per-residue protein interaction task is provided in [24]. The task is as follows: given a static biological molecule extracted from a PDB file for instance a protein, DNA or RNA molecule, predict for all biological sub-units (residue, nucleotide, ligand, lipid or ion) the probability of interacting with another type of sub-unit which is not part of the same molecule. The data is constructed by parsing an assembly in a PDB file and, using a representative coordinate for all sub-units considered; over all sub-unit pairs belonging to different molecules, store the pairs within a distance cutoff, in this case 5Å. We use the data processing script provided with the original paper, which processes the following unique biological sub-units: 20

amino acids, 8 nucleic acids, 16 common ions, 31 ligands, and 4 lipids which amounts to 79 unique types of molecules. These are subsequently grouped into five groups: amino-acids, nucleotides, ions, ligands and lipids. As in [24] we evaluate on the MaSIF-site dataset [61, 62]. The authors use a model which can take a protein as input and for each residue produce five predicted probabilities for interaction with all five sub-unit types. Note that a given residue can be in contact with multiple other sub-unit types, or none at all. The loss function for this task is binary cross-entropy which has a positive class weighting as in the original paper. For comparison reasons, we use the same metric as PeSTo [24] which is median ROCAUC. Of the 27M residues in the training dataset 14% of them are in contact with another residue (nucleotides is 0.64%, ion 1.77%, ligand 3.40%, lipid 0.06%). Since the residue-residue interactions are relatively balanced, we believe it is acceptable to use ROCAUC; which also has the advantage of always knowing what a random guess is, over AUCPR.

D Results Table

Exact numerical values corresponding to bar charts shown in Figure 2.

Task	ESM	GNN (random)	GNN (pre-trained)	BioCLIP	SOTA	Epochs
MF	0.289	0.164	0.292	0.440		1
	0.446	0.193	0.380	0.533		2
	0.515	0.210	0.390	0.539		3
	0.611	0.370	0.500	**0.607**	0.546	Final
CC	0.394	0.295	0.395	0.386		1
	0.456	0.319	0.424	0.413		2
	0.460	0.318	0.413	0.466		3
	0.423	0.316	0.389	0.428	**0.497**	Final
BP	0.314	0.250	0.281	0.341		1
	0.356	0.234	0.302	0.372		2
	0.378	0.235	0.315	0.395		3
	0.431	0.245	0.337	0.440	**0.449**	Final
EC	0.073	0.057	0.113	0.447		1
	0.168	0.059	0.421	0.653		2
	0.346	0.063	0.550	0.757		3
	0.850	0.514	0.782	**0.868**	0.798	Final
PPI-Human	0.824	0.923	0.990	0.987		1
	0.850	0.970	0.991	0.992		2
	0.872	0.979	0.992	0.993		3
	0.992	0.991	0.994	0.993	**0.995**	Final
PPI-Cerevisiae	0.662	0.659	0.942	0.953		1
	0.706	0.744	0.948	0.958		2
	0.746	0.817	0.957	0.966		3
	0.934	0.954	**0.981**	0.974	0.973	Final
Res-PPI	0.789	0.704	0.743	0.818		1
	0.800	0.800	0.796	0.833		2
	0.823	0.794	0.820	0.845		3
	0.803	0.797	0.796	0.833	**0.924**	Final

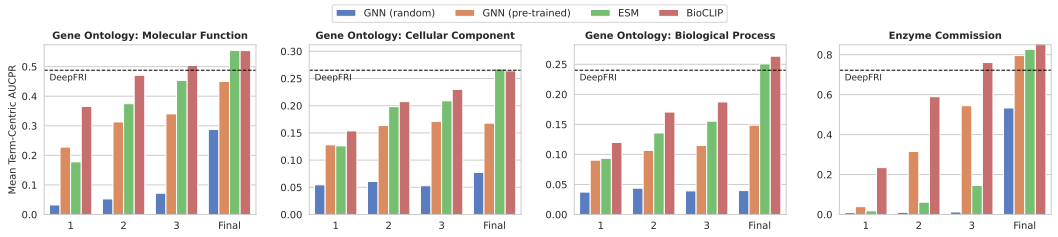

Figure 3: Performance with the alternative metric mean term-centric Fmax across the GO/EC downstream tasks for: Random GNN, pre-trained GNN with the contrastive BioCLIP objective, ESM 150M with the last three layers tuned, and the BioCLIP model which combines ESM and the pre-trained GNN. For each task, we also provide a recent, task-specific, benchmark represented as a dashed line.

E Results Table

Exact numerical values corresponding to bar charts shown in Figure 3.

Task	ESM	GNN (random)	GNN (pre-trained)	BioCLIP	SOTA	Epochs
MF	0.177	0.032	0.227	0.365		1
	0.374	0.052	0.312	0.470		2
	0.453	0.071	0.340	0.503		3
	0.554	0.287	0.449	**0.554**	0.487	Final
CC	0.126	0.054	0.128	0.153		1
	0.198	0.060	0.163	0.207		2
	0.208	0.052	0.171	0.230		3
	0.267	0.077	0.167	0.264	0.265	Final
BP	0.093	0.037	0.090	0.119		1
	0.135	0.043	0.106	0.170		2
	0.155	0.039	0.114	0.187		3
	0.250	0.039	0.148	**0.263**	0.240	Final
EC	0.018	0.009	0.039	0.234		1
	0.061	0.010	0.315	0.589		2
	0.145	0.012	0.544	0.760		3
	0.826	0.533	0.795	**0.854**	0.722	Final