Evaluating Representation Learning on the Protein Structure Universe

Arian R. Jamasb^{*,1}**, Alex Morehead**^{*,2}**, Chaitanya K. Joshi**^{*,1}**, Zuobai Zhang**^{*,3}**,**
Kieran Didi^{1}**, Simon V. Mathis**^{1}**, Charles Harris**^{1}**,**
Jian Tang^{3}**, Jianlin Cheng**^{2}**, Pietro Liò**^{1}**, Tom L. Blundell**^{1}
^{1}University of Cambridge ^{2}University of Missouri ^{3}Mila - Québec AI Institute

Abstract

We introduce *ProteinWorkshop*, a comprehensive benchmark suite for evaluating protein structure representation learning methods. We provide large-scale pre-training and downstream tasks comprised of both experimental and predicted structures, offering a balanced challenge to representation learning algorithms. We benchmark state-of-the-art protein-specific and generic geometric Graph Neural Networks and the extent to which they benefit from different types of pre-training. We find that: (1) Pretraining and auxiliary tasks consistently improve the performance of both rotation-invariant and equivariant geometric models; (2) More expressive equivariant models benefit from pretraining to a greater extent compared to invariant models; (3) Our open-source codebase reduces the barrier to entry for working with large protein structure datasets by providing storage-efficient dataloaders from large-scale predicted structures including AlphaFoldDB and ESM Atlas, as well as utilities for constructing new tasks from the entire PDB. *ProteinWorkshop* is available at: `https://github.com/a-r-j/ProteinWorkshop`.

1 Introduction

Modern protein structure prediction methods have led to an explosion in the availability of structural data [1, 2]. While many sequence-based functional annotations can be directly mapped to structures, this has resulted in a significantly-increasing gap between structures and meaningful *structural* annotations [3]. Recent work has focussed on developing methods to draw biological insights from large volumes of structural data by either determining representative structures that can be used to provide such annotations [4] or representing structures in a simplified and compact manner such as sequence alphabets [5] or graph embeddings [6]. These works have significantly reduced the computational resources required to process and analyse such structural representatives at scale. Nonetheless, it remains to be shown how such results can help us better understand the relationship between protein sequence, structure, and function through the use of deep learning algorithms.

Several deep learning methods have been developed for protein structures. In particular, geometric Graph Neural Networks (GNNs) have emerged as the architecture of choice for learning structural representations of biomolecules [7, 8, 9, 10, 11, 12], with methods differing in (1) the featurisation schemes and level of granularity of input structures (Cα, backbones, all-atom); as well as (2) the choice of physical symmetries and inductive biases enforced by the architecture [13]. However, there remains a need for a robust, standardised benchmark to track the progress of new and established methods with greater granularity and relevance to downstream applications.

In this work, we develop a unified and rigorous framework for evaluating protein structural encoders, providing pretraining corpora that span known foldspace and tasks that assess the ability of models

[*] Equal contribution. Correspondance to: `arj39@cam.ac.uk`

Machine Learning for Structural Biology Workshop, NeurIPS 2023.

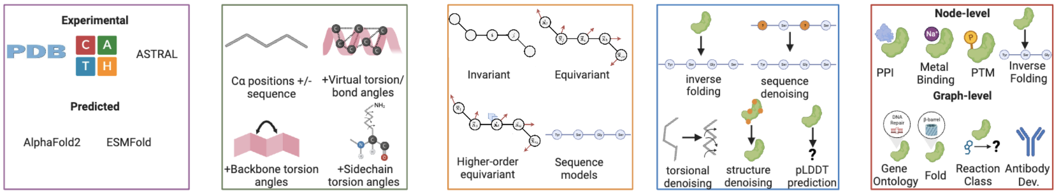

Figure 1: Overview of *ProteinWorkshop*, a comprehensive benchmark suite for evaluating pretraining and representation learning on large-scale protein structure data.

to learn informative representations at different levels of structural granularity. Previous works in protein structure representation learning have focused on learning effective *global* (i.e. graph-level) representations of protein structure, typically evaluating the methods on function or fold classification tasks [14, 12]. However, there has been comparatively little investigation into the ability of different methods to learn informative local (*node-level*) representations. Good node-level representations are important for a variety of annotation tasks, such as binding or interaction site prediction [15], as well as providing conditioning signals in structure-conditioned molecule design methods [16, 17]. Understanding the structure-function relationship at this granular level can drive progress in protein design by revealing structural motifs that underlie desirable properties, enabling them to be incorporated into designs.

Our contributions are as follows: (1) We curate numerous *structure-based* pretraining and fine-tuning datasets from the literature with a focus on tasks that can enable structural annotation of predicted structures. We compile a highly-modular benchmark, enabling the community to rapidly evaluate protein representation learning methods across tasks, models, and pretraining setups. (2) We benchmark protein structure representation learning at different levels of structural granularity (Cα, backbone, sidechain) and across several families of geometric GNNs, ranging from general-purpose [7, 18] to protein-specific architectures [11, 12]. We are the first to evaluate higher order tensor-based equivariant GNNs [19, 20] for protein representation learning. (3) We pre-train and evaluate models on, to our knowledge, the *largest non-redundant* protein structure corpus containing 2.27 million structures. (4) Our benchmarking study finds that sequence denoising-based auxiliary tasks and structure denoising-based pretraining consistently improve GNN model performance. Moreover, we surprisingly observe that sequence-based pretrained ESM2 [21] augmented with our structural featurisation matches state-of-the-art GNNs on (super)family fold and gene ontology prediction.

2 Benchmarking Geometric GNNs for Protein Structure

The overarching goal of the benchmark is to effectively cover the design space of protein structure representation learning methods. To achieve this, the benchmark is highly modular by design, enabling evaluation of different combinations of structural encoders, protein featurisation schemes, and auxiliary tasks over a wide range of both supervised and unsupervised tasks. A user manual for *ProteinWorkshop* is available in Appendix C, containing detailed listings and descriptions of the benchmark components, datasets, evaluation metrics, and hyperparameters.

Overview. To demonstrate the utility of our benchmark, we investigate how combinations of protein structure representation, architecture choice and pretraining/auxiliary tasks affect predictive performance across a range of tasks. The tasks are selected to focus on important real-world structural annotation tasks and such that we can evaluate these combinations in terms of both the local and global representational power. To this end, we select state-of-the-art protein structure encoders and generic geometric GNN architectures that span the design space of geometric GNN models with regard to both message passing body order and tensor order [13]. We evaluate several structural representations that, to varying degrees, capture the full detail of the protein structure.

Architectures. We provide a unified implementation of several rotation invariant and equivariant architectures. We benchmark 4 general purpose models: SchNet [7], EGNN [18], TFN [19], MACE [20]; and 2 protein-specific architectures: GCPNet [11], GearNet [12]. Reviewer 5uRv, foh9, kej2 We also compare geometric GNNs to the pretrained sequence-based language model ESM [21] augmented with structural featurisation. We chose the 650M pretrained ESM2 because this is the scale at which significant structure-related abilities were observed for ESM.

Featurisation Schemes. We consider five featurisation schemes, progressively increasing the amount of structural information provided to the model by incorporating sequence positional information, virtual dihedral and bond angles over the $C\alpha$ trace, backbone torsion angles, and sidechain torsion angles. Featurisation schemes are detailed in Table 4 in the Appendix.

Pretraining Dataset. For all pretraining experiments we use AlphaFoldDB [22]. This dataset provides a rich diversity of 2.27 million non-redundant protein structures and, to our knowledge, is substantially more diverse than any other previously used structure-based pretraining corpus, whilst remaining of a size that is amenable to experimentation. Models pretrained on AlphaFoldDB should, in principle, exhibit strong generalisation to the currently known (and predicted) natural protein structure universe as it would have 'seen' the same protein fold during pretraining. To facilitate working with large-scale AlphaFoldDB and ESMAtlas, we developed storage-efficient dataloaders based on FoldComp Kim et al. [23], described in Appendix C.

Pretraining and Auxiliary Tasks. In our evaluation, we focus predominantly on denoising-based pretraining and auxiliary tasks as these are comparatively less explored than contrastive or masked-attribute prediction tasks [12]. We consider five pretraining tasks: (1) structure-based denoising, (2) sequence denoising, (3) torsional denoising, (4) inverse folding and (5) pLDDT prediction. Structure and sequence denoising are also used as auxiliary tasks in our experiments. We also investigate an inverse folding pre-training task which we subsequently finetune on the CATH dataset for benchmarking inverse folding as a downstream task (see below).

Noising Schemes. For structure-based denoising we draw i.i.d. noise samples from a Gaussian distribution and scale by $\sigma = 0.1$ to corrupt the input coordinates or dihedral angles. Geometric scalar and vector-valued features are computed from the noised structure, *i.e.* $\tilde{\mathcal{G}} = (\mathcal{V}, \tilde{\mathcal{E}}, \tilde{\mathbf{X}}, \tilde{\mathbf{S}}, \tilde{\mathbf{V}})$, where $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_i + \sigma\boldsymbol{\epsilon}_i$ and $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, I_3)$. For sequence-based denoising, we use the mutation strategy and corrupt 25% of the residues in each protein. When sequence denoising is used as an auxiliary task, we weight the loss with a coefficient $\lambda = 0.1$, similar to NoisyNodes [24].

Training. As we are interested in benchmarking large-scale datasets and models, we try to consistently use six layers for all models, each with 512 hidden channels. For equivariant GNNs, we reduced the number of layers and hidden channels to fit 80GB of GPU memory on one NVIDIA A100 GPU. For downstream tasks, we set the maximum number of epochs to 150 and use the Adam optimizer with a batch size of 32 and ReduceLROnPlateau learning rate scheduler, monitoring the validation metric with patience of 5 epochs and reduction of 0.6. See Appendix C.8 for details on hyperparameter tuning for optimal learning rates and dropouts for each architecture. We train models to convergence, monitoring the validation metric and performing early stopping with a patience of 10 epochs. Pretraining is performed for 10 epochs using a linear warm-up with cosine schedule. We report standard deviations over three runs across three random seeds.

3 Results & Discussions

3.1 Auxiliary Sequence Denoising Consistently Improves Baseline Performance

In Table 1, we first set out to determine the following questions about architectural choices in conjunction with denoising auxiliary tasks and *without* pretraining:

1. **Whether invariant or equivariant models perform better?** Across 8 tasks, equivariant models such as EGNN, GCPNet and TFN attain the best performance on 5. Notably, sequence-based ESM augmented with our structural featurisation matches state-of-the-art protein-specific GNNs on (super)family and gene ontology prediction.
2. **Which input representation is the best for each respective task?** Featurising models with $C\alpha$ atoms, virtual angles, and backbone torsions provides the best performance overall on 21 out of 48 combinations of models and tasks. Letting models implicitly learn about side chain orientation by using backbone-only featurisation may prevent overfitting on crystallisation artifacts [25].
3. **Whether auxiliary denoising tasks improve model performance?** Sequence denoising is a particularly useful auxiliary task for training protein structure encoders until sufficient structural detail makes the task trivial, improving results over not using auxiliary tasks for 98 out of 210 combinations of models, task, and auxiliary tasks. Notably, structure denoising helped stabilise the training of MACE models on the GO-CC and Reaction tasks, where other runs did not converge.

Table 1: **Baseline benchmark results without pretraining.** Results for each model and featurisation pair are given as: no auxiliary task / +sequence denoising / +structure denoising. Coloured boxes mark the best auxiliary tasks per model and featurisation, underlined results denote the best featurisation choice per model, and **bold results** are the best models for each task. Greyed cells denote invalid task-setup combinations (e.g. inverse folding and sequence denoising as auxiliary task), and —— denote runs that did not converge. [*] denotes results for ESM without structural features, taken from Zhang et al. [12]. **Key takeaways:** (1) Sequence denoising as an auxiliary task consistently improves performance across models. (2) Equivariant GNNs outperform invariant GNNs, in general. (3) $C\alpha$, virtual angles, and backbone torsions provide the best featurisation. (4) Augmenting ESM2 with structural features provides compelling performance for (super)family fold classification and gene ontology prediction compared to GNNs.

Method	Features	GO-CC (↑)	Ab. Dev. (↑)	Fold (↑)			Reaction (↑)	PPI (↑)	Inverse Folding (↓)
				Fold	Family	Superfamily			
ESM	Seq.	0.488*	0.91	26.8*	97.8*	60.1*	80.87	0.953	N/A
	Seq. + $\kappa, \alpha + \phi, \psi, \omega$	**0.589**	0.91	39.89±0.00	**99.69**±0.00	**78.68**±0.00	81.79	0.954	N/A
SchNet	$C\alpha$	0.428 / 0.421 / 0.461	0.88 / 0.89 / 0.88	27.11±0.01 / 31.36±0.01 / 26.05±0.01	81.29±0.02 / 84.88±0.01 / 79.03±0.03	30.02±0.00 / 34.79±0.01 / 27.79±0.01	56.35 / 65.71 / 57.68	0.955 / 0.952 / 0.955	
	$C\alpha$ + Seq.	0.383 / 0.365 / 0.387	0.86 / 0.86 / 0.87	34.44±0.00 / 38.06±0.01 / 33.31±0.01	91.12±0.02 / 92.59±0.01 / 91.82±0.02	39.33±0.02 / 44.27±0.01 / 39.67±0.01	63.59 / 43.92 / 45.80	0.953 / 0.951 / 0.953	11.78±0.08 / 12.34±0.11
	$C\alpha + \kappa, \alpha$	0.393 / 0.336 / 0.366	0.86 / 0.84 / 0.88	33.22±0.02 / 37.93±0.00 / 34.99±0.01	91.33±0.03 / 94.06±0.01 / 93.14±0.01	39.96±0.03 / 45.61±0.01 / 39.39±0.02	59.70 / —— / 45.71	0.954 / 0.951 / 0.955	11.03±0.03 / 11.81±0.50
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega$	0.452 / 0.395 / 0.477	0.85 / 0.90 / 0.86	36.01±0.01 / 35.88±0.01 / 33.71±0.02	93.42±0.01 / 93.13±0.00 / 92.84±0.02	42.97±0.01 / 44.14±0.01 / 41.33 ±0.03	71.71 / 73.79 / 73.65	0.955 / 0.952 / 0.955	9.97±0.09 / 10.76±0.02
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega + \chi_{1-4}$	0.447 / 0.396 / 0.456	0.87 / 0.89 / 0.87	35.60±0.02 / 33.90 ±0.02 / 35.79±0.02	92.96±0.00 / 90.80±0.01 / 93.05±0.00	43.05±0.01 / 40.77±0.01 / 41.72 ±0.02	67.68 / 66.06 / 71.78	0.953 / 0.950 / 0.956	
GearNet-Edge	$C\alpha$	0.489 / 0.475 / 0.496	0.82 / 0.83 / 0.80	38.14±0.03 / 38.15 ±0.02 / 38.62±0.01	95.45±0.02 / 96.15±0.01 / 95.73±0.02	51.16±0.03 / 53.52 ±0.04 / 52.24±0.05	76.49 / —— / 76.79	0.963 / 0.957 / 0.961	
	$C\alpha$ + Seq.	0.488 / 0.496 / 0.490	0.80 / 0.84 / 0.81	—— / 38.42±0.01 / 39.45±0.02	—— / 95.35±0.02 / 96.45±0.00	—— / 50.98±0.02 / 52.35±0.01	77.14 / 78.88 / 72.42	0.957 / 0.949 / 0.954	12.79±0.17 / 12.60±0.16
	$C\alpha + \kappa, \alpha$	0.467 / 0.484 / 0.479	0.82 / 0.81 / 0.82	—— / 40.69±0.01 / 41.52±0.00	—— / 97.12±0.00 / 96.57±0.00	—— / 54.83±0.02 / 53.32±0.00	78.89 / 79.15 / 80.46	0.958 / 0.955 / 0.960	12.35±0.05 / 11.91 ±0.13
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega$	0.488 / 0.494 / 0.496	0.81 / 0.80 / 0.78	39.92±0.01 / 41.31±0.00 / 41.52 ±0.00	96.26±0.00 / 96.68±0.00 / 96.87 ±0.00	53.00±0.03 / 54.12±0.02 / 54.72±0.01	75.22 / —— / 78.18	0.959 / 0.955 / 0.956	11.61±0.12 / 11.23±0.09
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega + \chi_{1-4}$	0.480 / 0.491 / 0.484	0.81 / 0.83 / 0.83	37.59±0.02 / 39.23±0.01 / 39.11±0.00	96.09±0.01 / 95.71±0.00 / 96.10±0.00	51.10±0.01 / 50.23±0.00 / 50.86±0.00	—— / —— / 73.51	0.964 / 0.960 / 0.960	
EGNN	$C\alpha$	0.459 / 0.470 / 0.464	0.93 / 0.90 / 0.92	31.51±0.01 / 36.81±0.01 / 30.84±0.01	94.54±0.00 / 96.71±0.00 / 94.14±0.00	42.70±0.02 / 50.93±0.01 / 41.87±0.01	63.20 / —— / ——	0.966 / 0.963 / 0.964	
	$C\alpha$ + Seq.	0.445 / 0.468 / 0.444	0.87 / 0.90 / 0.88	41.63±0.01 / 44.77±0.01 / 42.68±0.00	97.87±0.00 / 98.00±0.00 / 97.89±0.00	55.47±0.00 / 59.54±0.01 / 56.80±0.03	74.79 / 78.79 / ——	0.962 / 0.961 / 0.962	10.28±0.04 / 10.53±0.01
	$C\alpha + \kappa, \alpha$	0.465 / 0.499 / 0.458	0.88 / 0.89 / 0.86	45.09±0.00 / **48.26**±0.02 / 43.65±0.00	98.09±0.00 / 98.55±0.00 / 98.03±0.00	58.15±0.01 / 62.86±0.01 / 58.31±0.01	—— / 83.44 / 81.96	0.965 / 0.964 / 0.963	9.84±0.07 / 10.07±0.04
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega$	0.464 / 0.476 / 0.451	0.88 / 0.87 / 0.88	44.70±0.01 / 46.97±0.01 / 45.49±0.01	97.76±0.00 / 97.99±0.00 / 98.00±0.00	59.66±0.01 / 62.83±0.01 / 60.38±0.02	—— / 82.80 / **85.58**	0.962 / 0.963 / 0.964	8.89±0.04 / 9.65±0.03
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega + \chi_{1-4}$	0.452 / 0.452 / 0.458	0.86 / 0.90 / 0.87	43.16±0.01 / 44.07±0.00 / 43.41±0.02	97.78±0.00 / 97.46±0.00 / 97.76±0.00	59.01±0.01 / 56.94±0.01 / 58.4±0.01	76.15 / —— / ——	0.963 / 0.962 / 0.961	
GCPNet	$C\alpha$	0.418 / 0.416 / 0.404	0.82 / 0.90 / 0.84	38.40±0.00 / 43.11±0.01 / 37.43±0.01	95.75±0.00 / 97.22±0.00 / 95.57±0.01	47.85±0.02 / 56.47±0.00 / 47.55±0.01	64.92 / 74.26 / 66.58	**0.968** / 0.960 / 0.967	
	$C\alpha$ + Seq.	0.404 / 0.417 / ——	0.84 / 0.85 / 0.84	43.89±0.01 / 45.38±0.01 / 42.33±0.00	97.31±0.00 / 97.49±0.00 / 97.19±0.00	54.19±0.01 / 58.89±0.02 / 53.07±0.00	69.95 / 63.85 / 73.23	0.966 / 0.960 / 0.967	8.35±0.08 / 8.92±0.07
	$C\alpha + \kappa, \alpha$	0.454 / 0.455 / 0.453	0.88 / 0.82 / 0.85	43.92±0.02 / 45.71±0.00 / 43.76±0.01	97.93 ±0.00 / 97.85 ±0.00 / 97.95±0.00	56.28±0.01 / 58.61±0.00 / 57.91±0.00	76.83 / 77.57 / 77.63	0.966 / 0.961 / 0.967	8.80±0.09 / 9.49±0.18
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega$	0.447 / 0.439 / 0.449	0.87 / 0.83 / 0.87	44.76±0.01 / 45.52±0.01 / 45.41±0.00	97.55±0.00 / 97.33±0.00 / 97.56±0.00	56.83±0.01 / 57.97±0.00 / 57.33±0.01	77.17 / 78.55 / 78.45	**0.968** / 0.962 / 0.968	**7.56**±0.11 / 8.60±0.09
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega + \chi_{1-4}$	0.428 / 0.449 / 0.410	0.85 / 0.86 / 0.80	42.01±0.02 / 42.52±0.01 / 43.6±0.01	96.68±0.00 / 96.55±0.00 / 97.36±0.00	53.4±0.01 / 53.14±0.01 / 54.55±0.00	74.51 / 70.64 / 73.77	**0.968** / 0.963 / 0.968	
TFN	$C\alpha$	0.464 / 0.475 / 0.476	**0.94** / 0.90 / **0.94**	31.16±0.00 / 37.81±0.00 / ——	94.59±0.00 / 97.35±0.00 / ——	41.61 ±0.00 / 53.37±0.01 / ——	67.86 / 81.05 / 68.79	0.966 / 0.962 / 0.966	
	$C\alpha$ + Seq.	0.460 / 0.486 / 0.469	0.88 / 0.88 / ——	36.80±0.00 / 39.33±0.00 / 38.80±0.00	96.14 ±0.00 / 96.72±0.00 / 96.38±0.00	48.40±0.01 / 52.48±0.01 / 48.86±0.00	74.03 / 72.60 / 71.21	0.960 / 0.959/ 0.961	10.34±0.03 / 10.84±0.04
	$C\alpha + \kappa, \alpha$	0.467 / 0.497 / 0.472	0.90 / —— / ——	38.89±0.01 / 42.56±0.00 / 40.93 ±0.00	97.41 ±0.00 / 97.84 ±0.00 / 97.59 ±0.00	53.49±0.01 / 57.54 ±0.00 / 54.27±0.00	72.73 / 77.37 / 73.33	0.965 / 0.961 / 0.963	10.02±0.05 / 10.46±0.11
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega$	0.462 / 0.484 / 0.461	0.90 / 0.89 / 0.88	41.09±0.00 / 43.11±0.00 / 39.68±0.00	97.65±0.00 / 97.66 ±0.00 / 97.63 ±0.00	54.00 ±0.00 / 59.65 ±0.00 / 56.19±0.00	75.83 / 75.38 / 75.99	0.966 / 0.962 / 0.965	8.73±0.02 / 9.73±0.01
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega + \chi_{1-4}$	0.464 / 0.474 / 0.461	0.88 / —— / ——	40.07±0.00 / 38.33 ±0.00 / ——	97.36±0.00 / 96.96 ±0.00 / ——	54.79±0.00 / 55.14±0.00 / ——	74.83 / 79.44 / 76.73	0.964 / 0.961 / 0.965	
MACE	$C\alpha$	—— / —— / 0.465	0.93 / 0.92 / 0.93	34.31±0.01 / 36.68±0.00 / 35.21±0.01	93.10±0.01 / 95.67±0.01 / 92.4±0.00	44.86±0.02 / 50.41±0.02 / 44.14±0.02	—— / —— / 70.87	0.963 / 0.960 / ——	
	$C\alpha$ + Seq.	—— / —— / 0.446	0.82 / 0.89 / 0.84	38.23±0.02 / 40.23±0.01 / 38.75±0.00	96.06±0.01 / 96.21±0.00 / 95.62±0.00	48.45±0.03 / 50.75±0.01 / 48.24±0.02	—— / —— / 73.71	0.963 / 0.960 / ——	9.95±0.10 / 10.3±0.04
	$C\alpha + \kappa, \alpha$	—— / —— / ——	0.72 / 0.90 / 0.78	39.20±0.02 / 40.39±0.01 / 39.02±0.02	95.92±0.01 / 96.78±0.01 / 96.15±0.01	50.16±0.01 / 53.32±0.01 / 51.46±0.02	—— / —— / 79.31	0.965 / 0.959 / ——	10.30±0.02 / 10.60±0.02
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega$	—— / —— / 0.452	0.83 / 0.89 / 0.84	42.17±0.02 / 40.55 ±0.02 / 41.29±0.02	97.53±0.01 / 97.23±0.01 / 97.34 ±0.00	55.24±0.02 / 55.15±0.01 / 54.17±0.01	—— / —— / 77.35	0.965 / 0.960 / ——	8.94±0.03 / 9.75±0.07
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega + \chi_{1-4}$	—— / —— / 0.454	0.82 / 0.88 / 0.80	39.931±0.01 / 39.62±0.01 / 41.74 ±0.02	97.33±0.00 / 96.13±0.00 / 97.49±0.00	54.65±0.02 / 51.27±0.01 / 55.05 ±0.01	—— / —— / 77.43	0.964 / 0.960 / ——	

3.2 Incorporating More Structural Detail Improves Pre-Training Performance

Table 2: **Validation performance for pretraining tasks on AlphaFoldDB.** Metrics: Inverse Folding: perplexity; pLDDT, Structure Denoising, Torsional Denoising: RMSE; Seq. Denoising: Accuracy. **Key takeaway:** Incorporating backbone geometry consistently improves pretraining performance.

	Method	Task				
		Inverse Folding (↓)	**pLDDT Pred. (↓)**	**Structure Denoising (↓)**	**Seq. Denoising (↑)**	**Torsional Denoising (↓)**
$C_\alpha + \kappa, \alpha$	SchNet	7.791	0.2397	0.0704	36.81	0.0586
	GearNet-Edge	6.596	0.2326	0.0672	43.76	0.0615
	EGNN	**6.016**	0.2406	0.0700	40.51	0.0586
	GCPNet	6.243	0.2395	0.0679	44.81	**0.0562**
$C_\alpha + \phi, \psi, \omega$	SchNet	5.562	0.2388	0.0603	45.61	0.0489
	GearNet-Edge	5.324	0.2402	0.0562	50.15	0.0538
	EGNN	5.962	0.2403	0.0593	53.80	0.0487
	GCPNet	**3.839**	0.2399	0.0561	**59.54**	**0.0443**

We then investigated protein structure pre-training in Table 2 to find out:

1. **Which input representation is best for pre-training?** Incorporating dihedral angles consistently improves validation metrics on pre-training tasks, more so than architecture.
2. **Which GNNs benefit from which pre-training task?** Inverse folding, sequence denoising, and torsional denoising benefit equivariant models the most in the context of pre-training, whereas pLDDT prediction and structure denoising benefit invariant models the most, suggesting that certain pre-training tasks benefit certain classes of models more than other tasks. Unfortunately, we were currently unable to pre-train spherical equivariant GNNs (TFN, MACE) due to the high computational requirements of these models.

3.3 Pre-training and Greater Structural Detail Benefit Downstream Tasks

Table 3: **Pretrained model benchmark results.** Results for each model and featurisation pair are given as: no pretraining / sequence denoising / structure denoising , except for inverse folding on CATH, which is pretrained with inverse folding on AFDB. **Key takeaway:** Equivariant models benefit most from pretraining and maximum structural detail.

Method	**Features**	**Fold (↑)**			**Inverse Folding (↓)**
		Fold	Family	Superfamily	
GearNet	$C\alpha + \kappa, \alpha$	—— / 36.00 / 40.54	—— / 94.00 / 97.68	—— / 49.58 / 55.60	12.35 / 7.84
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega$	39.92 / 38.45 / 42.94	96.26 / 96.15 / 96.17	53.00 / 51.79 / 54.51	11.61 / 7.29
GCPNet	$C\alpha + \kappa, \alpha$	43.92 / 44.76 / 48.37	97.93 / 98.41 / **98.27**	56.28 / 60.47 / 62.64	8.80 / 7.37
	$C\alpha + \kappa, \alpha + \phi, \psi, \omega$	44.76 / 47.53 / **49.53**	97.55 / 97.71 / 97.98	56.83 / 59.31 / **64.54**	7.56 / **6.55**

Following the observation that more fine-grained input representations improve pretraining performance, Table 3 explores:

1. **Whether these lessons from pretraining translate to downstream tasks?** Equivariant models benefit the most from pretraining on structure-based tasks, particularly when provided the greatest amount of structural detail.
2. **Which combination of parameters performs best on downstream tasks?** Providing a greater amount of structural detail compared to a strict Cα atom representation benefits downstream performance for *both* invariant and equivariant models. Notably, structure denoising consistently improves downstream performance for *both* types of models.

3.4 Conclusion

Our benchmark provides a unified implementation of several large pretraining corpora, featurisation schemes, models and benchmarking tasks to evaluate the effectiveness of protein structure encoding methods. Key findings include that structural pretraining, as well as auxiliary sequence denoising tasks, improve performance on a wide range of tasks and that incorporating more structural detail in the featurisation improves downstream performance.

Future work. Importantly, we would like to advocate for rigorous and continuous re-evaluation of protein representation learning to stay up to date with new advances in protein structure prediction, such as modelling quaternary structures and conformational ensembles. Our benchmark is flexible for including new tasks and datasets and is open to the wider research community.

The benchmark code is available under a permissive MIT License at `https://github.com/a-r-j/ProteinWorkshop` and accompanying documentation, preprocessed datasets and pretrained model weights will be hosted publicly in the future. Preprocessed datasets and pretrained model weights are deposited on Zenodo at the following URLs, respectively: `https://zenodo.org/record/8282470` and `https://zenodo.org/record/8287754`.

References

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 2021.

[2] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 2021.

[3] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, Augustin Žídek, Tim Green, Kathryn Tunyasuvunakool, Stig Petersen, John Jumper, Ellen Clancy, Richard Green, Ankur Vora, Mira Lutfi, Michael Figurnov, Andrew Cowie, Nicole Hobbs, Pushmeet Kohli, Gerard Kleywegt, Ewan Birney, Demis Hassabis, and Sameer Velankar. AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50(D1):D439–D444, November 2021. doi: 10.1093/nar/gkab1061.

[4] Liisa Holm. Dali server: structural unification of protein families. *Nucleic acids research*, 2022.

[5] Michel van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L. M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, May 2023. doi: 10.1038/s41587-023-01773-0.

[6] Joe G Greener and Kiarash Jamali. Fast protein structure searching using structure graph embeddings. *bioRxiv*, 2022.

[7] 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*, 2018.

[8] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *ICLR*, 2020.

[9] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *ICLR*, 2020.

[10] Kristof Schütt, Oliver Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *ICML*, 2021.

[11] Alex Morehead and Jianlin Cheng. Geometry-complete perceptron networks for 3d molecular graphs. *arXiv preprint arXiv:2211.02504*, 2022.

[12] Zuobai Zhang, Minghao Xu, Arian Rokkum Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *The Eleventh International Conference on Learning Representations*, 2023.

[13] Chaitanya K. Joshi, Cristian Bodnar, Simon V. Mathis, Taco Cohen, and Pietro Liò. On the expressive power of geometric graph neural networks. In *International Conference on Machine Learning*, 2023.

[14] Vladimir Gligorijević, P. Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian M. Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, 12(1), May 2021. doi: 10.1038/s41467-021-23303-9.

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

[16] Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, et al. Structure-based drug design with equivariant diffusion models. *arXiv preprint*, 2022.

[17] Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. Diffdock: Diffusion steps, twists, and turns for molecular docking. In *ICLR*, 2023.

[18] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *ICML*, 2021.

[19] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint*, 2018.

[20] Ilyes Batatia, Dávid Péter Kovács, Gregor NC Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. In *NeurIPS*, 2022.

[21] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[22] Inigo Barrio-Hernandez, Jingi Yeo, Jürgen Jänes, Tanita Wein, Mihaly Varadi, Sameer Velankar, Pedro Beltrao, and Martin Steinegger. Clustering predicted structures at the scale of the known protein universe. *Bioinformatics*, March 2023. doi: 10.1101/2023.03.09.531927.

[23] Hyunbin Kim, Milot Mirdita, and Martin Steinegger. Foldcomp: a library and format for compressing and indexing large protein structure sets. *Bioinformatics*, 39(4), March 2023. doi: 10.1093/bioinformatics/btad153.

[24] Jonathan Godwin, Michael Schaarschmidt, Alexander Gaunt, Alvaro Sanchez-Gonzalez, Yulia Rubanova, Petar Veličković, James Kirkpatrick, and Peter Battaglia. Simple gnn regularisation for 3d molecular property prediction & beyond. *arXiv preprint arXiv: Arxiv-2106.07971*, 2021.

[25] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using ProteinMPNN. *Science*, 378(6615):49–56, October 2022. doi: 10.1126/science.add2187.

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

[27] Pedro Hermosilla, Marco Schäfer, Matěj Lang, Gloria Fackelmann, Pere Pau Vázquez, Barbora Kozlíková, Michael Krone, Tobias Ritschel, and Timo Ropinski. Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. *arXiv preprint arXiv:2007.06252*, 2020.

[28] 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.

[29] Wengong Jin, Jeremy Wohlwend, Regina Barzilay, and Tommi Jaakkola. Iterative refinement graph neural network for antibody sequence-structure co-design. *arXiv preprint arXiv:2110.04624*, 2021.

[30] Limei Wang, Haoran Liu, Yi Liu, Jerry Kurtin, and Shuiwang Ji. Learning hierarchical protein representations via complete 3d graph networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[31] Stephan Eismann, Patricia Suriana, Bowen Jing, Raphael JL Townshend, and Ron O Dror. Protein model quality assessment using rotation-equivariant, hierarchical neural networks. *arXiv preprint arXiv:2011.13557*, 2020.

[32] Bowen Dai and Chris Bailey-Kellogg. Protein interaction interface region prediction by geometric deep learning. *Bioinformatics*, 37(17):2580–2588, 2021.

[33] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[34] Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R Eddy, Aurélien Luciani, Simon C Potter, Matloob Qureshi, Lorna J Richardson, Gustavo A Salazar, Alfredo Smart, Erik L L Sonnhammer, Layla Hirsh, Lisanna Paladin, Damiano Piovesan, Silvio C E Tosatto, and Robert D Finn. The pfam protein families database in 2019. *Nucleic Acids Research*, 47(D1):D427–D432, October 2018. doi: 10.1093/nar/gky995.

[35] Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Ma Chang, Runcheng Liu, and Jian Tang. Peer: A comprehensive and multi-task benchmark for protein sequence understanding. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 35156–35173. Curran Associates, Inc., 2022.

[36] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

[37] Christian Dallago, Jody Mou, Jody Mou, Kadina Johnston, Bruce Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

[38] 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), September 2022. doi: 10.1038/s41598-022-19608-4.

[39] Raphael Townshend, Martin Vögele, Patricia Suriana, Alex Derry, Alexander Powers, Yianni Laloudakis, Sidhika Balachandar, Bowen Jing, Brandon Anderson, Stephan Eismann, Risi Kondor, Russ Altman, and Ron Dror. Atom3d: Tasks on molecules in three dimensions. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

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

[41] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *JMLR*, 2023.

[42] Sheheryar Zaidi, Michael Schaarschmidt, James Martens, Hyunjik Kim, Yee Whye Teh, Alvaro Sanchez-Gonzalez, Peter Battaglia, Razvan Pascanu, and Jonathan Godwin. Pre-training via denoising for molecular property prediction. In *The Eleventh International Conference on Learning Representations*, 2023.

[43] Yufei Huang, Lirong Wu, Haitao Lin, Jiangbin Zheng, Ge Wang, and Stan Z Li. Data-efficient protein 3d geometric pretraining via refinement of diffused protein structure decoy. *arXiv preprint arXiv:2302.10888*, 2023.

[44] Gabriele Corso. Modeling molecular structures with intrinsic diffusion models. *arXiv preprint arXiv:2302.12255*, 2023.

[45] Shengchao Liu, Hongyu Guo, and Jian Tang. Molecular geometry pretraining with SE(3)-invariant denoising distance matching. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=CjTHVo1dvR`.

[46] Zuobai Zhang, Minghao Xu, Vijil Chenthamarakshan, Aurélie Lozano, Payel Das, and Jian Tang. Enhancing protein language models with structure-based encoder and pre-training. *arXiv preprint arXiv:2303.06275*, 2023.

[47] Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan Anishchenko, Ian R Humphreys, et al. Generalized biomolecular modeling and design with rosettafold all-atom. *bioRxiv*, pages 2023–10, 2023.

[48] DeepMind-Isomorphic. Performance and structural coverage of the latest, in-development alphafold model. *DeepMind*, 2023.

[49] Charles Harris, Kieran Didi, Arian Jamasb, Chaitanya Joshi, Simon Mathis, Pietro Lio, and Tom Blundell. Posecheck: Generative models for 3d structure-based drug design produce unrealistic poses. In *NeurIPS 2023 Generative AI and Biology (GenBio) Workshop*, 2023. URL `https://openreview.net/forum?id=Nf5BxVllgq`.

[50] Martin Buttenschoen, Garrett M Morris, and Charlotte M Deane. Posebusters: Ai-based docking methods fail to generate physically valid poses or generalise to novel sequences. *arXiv preprint arXiv:2308.05777*, 2023.

[51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[52] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

[53] William A Falcon. Pytorch lightning. *GitHub*, 2019.

[54] Arian Rokkum Jamasb, Ramon Viñas Torné, Eric J Ma, Yuanqi Du, Charles Harris, Kexin Huang, Dominic Hall, Pietro Lio, and Tom Leon Blundell. Graphein - a python library for geometric deep learning and network analysis on biomolecular structures and interaction networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[55] Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019.

[56] Mario Geiger and Tess Smidt. e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[58] Alex Morehead and Jianlin Cheng. Geometry-complete diffusion for 3d molecule generation. *arXiv preprint arXiv:2302.04313*, 2023.

[59] H. M. Berman. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, January 2000. doi: 10.1093/nar/28.1.235.

[60] Anthony R. Bradley, Alexander S. Rose, Antonín Pavelka, Yana Valasatava, Jose M. Duarte, Andreas Prlić, and Peter W. Rose. MMTF—an efficient file format for the transmission, visualization, and analysis of macromolecular structures. *PLOS Computational Biology*, 13(6):e1005575, June 2017. doi: 10.1371/journal.pcbi.1005575.

[61] Michael Knudsen and Carsten Wiuf. The CATH database. *Human Genomics*, 4(3):207, 2010. doi: 10.1186/1479-7364-4-3-207.

[62] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[63] S. E. Brenner. The ASTRAL compendium for protein structure and sequence analysis. *Nucleic Acids Research*, 28(1):254–256, January 2000. doi: 10.1093/nar/28.1.254.

[64] Lorna Richardson, Ben Allen, Germana Baldi, Martin Beracochea, Maxwell L Bileschi, Tony Burdett, Josephine Burgin, Juan Caballero-Pérez, Guy Cochrane, Lucy J Colwell, Tom Curtis, Alejandra Escobar-Zepeda, Tatiana A Gurbich, Varsha Kale, Anton Korobeynikov, Shriya Raj, Alexander B Rogers, Ekaterina Sakharova, Santiago Sanchez, Darren J Wilkinson, and Robert D Finn. MGnify: the microbiome sequence data analysis resource in 2023. *Nucleic Acids Research*, 51(D1):D753–D759, December 2022. doi: 10.1093/nar/gkac1080.

[65] Mohammed AlQuraishi. Parallelized natural extension reference frame: Parallelized conversion from internal to cartesian coordinates. *Journal of Computational Chemistry*, 40(7):885–892, January 2019. doi: 10.1002/jcc.25772.

[66] Carter J Wilson, Wing-Yiu Choy, and Mikko Karttunen. Alphafold2: a role for disordered protein/region prediction? *International Journal of Molecular Sciences*, 23(9):4591, 2022.

[67] Fergus Boyles, Charlotte M Deane, and Garrett M Morris. Learning from the ligand: using ligand-based features to improve binding affinity prediction. *Bioinformatics*, 36(3):758–764, August 2019. doi: 10.1093/bioinformatics/btz665.

[68] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92, 2021.

[69] Zuobai Zhang, Minghao Xu, Aurélie Lozano, Vijil Chenthamarakshan, Payel Das, and Jian Tang. Physics-inspired protein encoder pre-training via siamese sequence-structure diffusion trajectory prediction, 2023.

[70] Yu Yan, Jyun-Yu Jiang, Mingzhou Fu, Ding Wang, Alexander R. Pelletier, Dibakar Sigdel, Dominic C.M. Ng, Wei Wang, and Peipei Ping. MIND-s is a deep-learning prediction model for elucidating protein post-translational modifications in human diseases. *Cell Reports Methods*, 3(3):100430, March 2023. doi: 10.1016/j.crmeth.2023.100430.

[71] Jie Hou, Badri Adhikari, and Jianlin Cheng. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, 12 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx780.

[72] Arian R. Jamasb, Ben Day, Cătălina Cangea, Pietro Liò, and Tom L. Blundell. Deep learning for protein–protein interaction site prediction. In *Methods in Molecular Biology*, pages 263–288. Springer US, 2021. doi: 10.1007/978-1-0716-1641-3_16.

[73] Simon L. Dürr, Andrea Levy, and Ursula Rothlisberger. Metal3d: a general deep learning framework for accurate metal ion location prediction in proteins. *Nature Communications*, 14(1), May 2023. doi: 10.1038/s41467-023-37870-6.

[74] James Dunbar, Konrad Krawczyk, Jinwoo Leem, Terry Baker, Angelika Fuchs, Guy Georges, Jiye Shi, and Charlotte M Deane. Sabdab: the structural antibody database. *Nucleic acids research*, 42(D1): D1140–D1146, 2014.

[75] Xingyao Chen, Thomas Dougherty, Chan Hong, Rachel Schibler, Yi Cong Zhao, Reza Sadeghi, Naim Matasci, Yi-Chieh Wu, and Ian Kerman. Predicting antibody developability from sequence using machine learning. *bioRxiv*, June 2020. doi: 10.1101/2020.06.18.159798.

[76] Timothy M. Lauer, Neeraj J. Agrawal, Naresh Chennamsetty, Kamal Egodage, Bernhard Helk, and Bernhardt L. Trout. Developability index: A rapid in silico tool for the screening of antibody aggregation propensity. *Journal of Pharmaceutical Sciences*, 101(1):102–115, January 2012. doi: 10.1002/jps.22758.

[77] Dassault Systèmes. Biovia discovery studio, 2016.

Appendices

A Broader Impacts

Our benchmark unifies protein representation learning tasks, large-scale pretraining datasets, featurisation schemes, and models. The wide range of tasks studied in our benchmark can enable us to develop insight into effective pretraining strategies, and whether pretrained protein structural representations can have material impact in real-world computational biology and drug discovery. It is not lost on us that these models can play a role in developing harmful chemical matter in the hands of a bad actor. Additionally, training very large models can contribute to climate change. However, we hope that developing highly effective structural representations will have broad, positive implications across biology and medicine that significantly outweigh the potential for misuse.

B Further Discussions

B.1 Related Work

Protein Structure Representation Learning. Several structure-based encoders for proteins have been designed to extract structural information from different levels of granularity, such as residue-level, atom-level and surfaces. In particular, previous works have aimed to encode protein structural priors directly within neural network architectures to model proteins hierarchically [26, 27], as computationally efficient point clouds [15, 28], or as k-nearest neighbors (k-NN) geometric graphs [9, 29, 11, 30, 12] for tasks such as protein function prediction [14], protein model quality assessment [31], and protein interaction region prediction [32].

Protein Benchmarks. Several benchmarks have been proposed for evaluating the efficacy of learnt protein *sequence* representations. However, *structure-based* benchmarks are comparatively unaddressed. Rao et al. [33] developed the TAPE (Tasks Assessing Protein Embeddings) benchmark, providing a large pretraining corpus of protein sequences curated from Pfam [34], as well as a collection of five supervised benchmark tasks assessing the ability of protein language models to predict structural qualities (contact prediction and secondary structure prediction), and functional properties (fluorescence and stability prediction). Xu et al. [35] developed the PEER (Protein Sequence Understanding) benchmark, focussing on multitask evaluation of protein sequence models. Therapeutic Data Commons [36] provide several datasets relevant to therapeutic development, however the few protein structure-derived datasets it contains are cast as sequence-based tasks. Dallago et al. [37] developed FLIP, a sequence-based benchmark of protein of fitness landscapes. ProteinGLUE [38] is another sequenced-based benchmark focussing on per-residue classification and regression tasks.

To our best knowledge, the only protein structure-benchmark to date is ATOM3D [39], which proposes a collection of tasks largely assessing geometric methods at predicting graph-level properties of protein structures. TorchProtein [40] also provides a small collection of global-structural datasets. In summary, however, most existing benchmarks fail to exhaustively evaluate both the local and global representation learning power of proposed methods. As the field of protein representation develops, we identify a need for a consistent benchmarking framework of diverse tasks to ensure improving results reported in the literature map on to progress in the downstream problems we hope to address. Similar benchmarking efforts for general purpose GNNs have provided experimental rigour to architectural research [41].

Denoising-Based Pre-training and Regularisation. Several methods have been developed for pre-training GNNs, predominantly focussing on cases where 3D coordinate information is only implicitly encoded in the graph structures. In this work, we build on work by Godwin et al. [24] and Zaidi et al. [42] to investigate whether denoising-based auxillary and pre-training tasks are effective methods for pre-training geometric GNNs operating on protein structures, similar to concurrent works bridging the gap between denoising objectives for geometric neural networks and diffusion generative modeling for biomolecules [43, 44].

B.2 Future Work

On input featurisation schemes and completeness. The extent to which providing complete information about all atoms and side chain orientations of each residue in a protein is debatable, as the exact coordinates from PDB files are known to contain artifacts from structure determination via crystallography. This was most recently noted by Dauparas et al. [25] in the context of developing and experimentally validating an inverse folding model. Our benchmarking results provides similar insights – at present, letting models implicitly learn about side chain orientation by using backbone-only featurisation performs better or equally well as explicitly providing complete side chain atomic information.

On the choice of pre-training tasks. We currenly focussed on pre-training tasks that roughly fall under the category of denoising, i.e. corrupting information in the input (sequence identity, coordinates) and tasking the model with producing the uncorrupted input. We were particularly interested in self-supervised objectives that were (1) extremely scalable, so as to pre-train on the large-scale AlphaFoldDB of 2.4M structures; and (2) train protein representations at the fine-grained node level, so as to be general-purpose across the downstream tasks considered. We did not benchmark other tasks from the literature, such as contrastive learning and generative modelling-inspired objectives [45, 46]. Such tasks are generally (1) computationally heavier and more cumbersome to set up than corruption-type objectives, making them harder to scale up, and (2) only train protein representations for the global/graph level and do not operate at the node level. Naturally, we would like to continue exploring more pre-training strategies as we continue to expand *ProteinWorkshop*.

Beyond protein-only representation learning. Recently, [47] and [48] generalised protein structure prediction models to full biological assemblies including proteins, small molecules, nucleic acids, and other ligands. Advances in geometric GNN modelling and methodology should, in principle, be adaptable to modelling biomolecular complexes among proteins, small molecules, nucleic acids, and other entities. Most architectures modelling biomolecular complexes also represent these systems as geometric graphs with atoms/residues embedded as nodes in 3D Euclidean space [17, 16]. As such, geometric GNNs appear to be a natural choice of architecture for representation learning across biomolecular systems.

We currently focus on protein representation learning because (1) large scale datasets for self-supervised learning, as well as well-defined downstream tasks, are readily available and accepted by the community; and (2) we see protein representation learning as a fundamental task, improving upon which should also advance the modelling of proteins in complex with other biomolecules. Comparatively, the scale of data available for biomolecular complexes is smaller and there is less consensus among the community on evaluation [49, 50].

C *ProteinWorkshop* User Manual

C.1 Usage and Dependencies

Usage. The modular design of our benchmark means it can be readily adapted into different workflows easily. Firstly, the benchmark is `pip`-installable from PyPI and contains several importable modules, such as dataloaders, featurisers and models, that can be imported into new projects. This will aid in standardising the datasets and workflows used in protein representation learning. Secondly, the benchmark serves as an easily extendable template, which users can fork and work directly in, thereby reducing implementation burden. Lastly, we provide a CLI that can be used to quickly run single experiments and hyperparameter sweeps with minimal development time overhead.

Dependencies. The benchmark is developed using `PyTorch` [51], `PyTorch Geometric` [52], `PyTorch Lightning` [53], and `Graphein` [54]. Experiment configuration is performed using `Hydra` [55]. Certain architectures introduce additional dependencies, such as `TorchDrug` [40] and `e3nn` [56].

Computational Resources. All models are trained on 80GB NVIDIA A100 GPUs. All baseline and finetuning results are performed using a single GPU while pre-training is performed using four GPUs.

C.2 Featurisation Schemes

Protein structures are typically represented as geometric graphs, with researchers opting to use a coarse-grained Cα atoms graph as full atom representations can quickly become computationally intractable due to a large number of nodes. The extent to which coarse-graining schemes are 'complete' representation of the geometry and structure of the protein residue is variable. For instance, backbone-only features ignore the orientations of the side chain atoms in the residue, so models

Table 4: **Structural featurisation schemes.** Residue type is a one-hot encoding of the amino acid type for each node; positional encoding is a 16-dimensional transformer-like positional encoding [57]; and $\phi, \psi, \omega \in \mathbb{R}^6$ and $\chi_{1-4} \in \mathbb{R}^8$ are backbone dihedral angles and sidechain torsion angles, respectively, embedded on the unit circle. Similarly, $\kappa, \alpha \in \mathbb{R}^4$ are *virtual* torsion and bond angles defined over Cα atoms. In our experimental evaluation, we consistently use k-NN edge construction with $k = 16$. For each pretraining task and model combination we train two models with different featurisation schemes providing (1) C_α geometry with virtual angles and (2) backbone information.

Granularity	$C\alpha$ **Features**	**Backbone**	**Sidechain**
$C\alpha$	Residue Type		
$C\alpha$	Residue Type, Positional Encoding		
$C\alpha$	Residue Type, Positional Encoding, κ, α		
$C\alpha$	Residue Type, Positional Encoding, κ, α	ϕ, ψ, ω	
$C\alpha$	Residue Type, Positional Encoding, κ, α	ϕ, ψ, ω	$\chi_1, \chi_2, \chi_3, \chi_4$

must account for this information implicitly. However, providing complete information about all atoms and side chain orientations is debatable as exact coordinates from PDB files are known to contain crystallography artefacts [25]. Due to the computational burden incurred by operating on full-atom node representations, we focus primarily on Cα-based graph representations, investigating featurisation strategies to incorporate higher-level structural information. Note that we do provide utilities to enable users to work with backbone and full-atom graphs in the benchmark.

We represent protein structures as geometric graphs, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \vec{\mathbf{X}}, \mathbf{S}, \vec{\mathbf{V}})$, where $\mathcal{V}$ is a set of nodes, $\mathcal{E}$ is a set of edges, $\vec{\mathbf{X}} \in \mathbb{R}^{|\mathcal{V}| \times 3}$ is a matrix of Cartesian node coordinates, $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is a matrix of d-dimension scalar node features, and $\vec{\mathbf{V}} \in \mathbb{R}^{|\mathcal{V}| \times d \times 3}$ is a tensor of vector-valued features.

The five scalar featurisation schemes considered in the baselines are provided in 4. Pretraining experiments are performed on rows three and four.

Additionally, vector message passing methods such as GCPNet receive node orientation vectors (i.e., $\boldsymbol{v}_{o^{i-1}}$ and $\boldsymbol{v}_{o^{i+1}}$) and edge directional vectors (i.e., $\boldsymbol{v}_{d^{ij}}$) as vector input features for nodes and edges, respectively. These vector features, prior to being normalised as unit vectors, are defined as:

$$\vec{\boldsymbol{v}}_{o^{i-1}} = \vec{\boldsymbol{x}}_{i-1} - \vec{\boldsymbol{x}}_i, \; \vec{\boldsymbol{v}}_{o^{i+1}} = \vec{\boldsymbol{x}}_{i+1} - \vec{\boldsymbol{x}}_i, \text{ and } \vec{\boldsymbol{v}}_{d^{ij}} = \vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j. \tag{1}$$

C.3 Protein Structure Encoder Architectures

We provide a unified implementation of several rotationally invariant and equivariant architectures spanning the range of message passing body order and tensor order [13], including general-purpose as well as protein-specific models.

C.3.1 Invariant GNNs

SchNet [7] SchNet is one of the most popular and simplest instantiation of E(3) invariant message passing GNNs. SchNet constructs messages through element-wise multiplication of scalar features modulated by a radial filter conditioned on the pairwise distance $\|\vec{\boldsymbol{x}}_{ij}\|$ between two neighbours. Scalar features are update from iteration t to $t + 1$ via:

$$\boldsymbol{s}_i^{(t+1)} := \boldsymbol{s}_i^{(t)} + \sum_{j \in \mathcal{N}_i} f_1 \left(\boldsymbol{s}_j^{(t)}, \; \|\vec{\boldsymbol{x}}_{ij}\| \right) \tag{2}$$

Hyperparameters: number of layers = 6, hidden channels = 512.

DimeNet++ [8] DimeNet is an E(3) invariant GNN which uses both distances $\|\vec{\boldsymbol{x}}_{ij}\|$ and angles $\vec{\boldsymbol{x}}_{ij} \cdot \vec{\boldsymbol{x}}_{ik}$ to perform message passing among triplets, as follows:

$$\boldsymbol{s}_i^{(t+1)} := \sum_{j \in \mathcal{N}_i} f_1 \Big(\boldsymbol{s}_i^{(t)}, \; \boldsymbol{s}_j^{(t)}, \sum_{k \in \mathcal{N}_i \setminus \{j\}} f_2 \left(\boldsymbol{s}_j^{(t)}, \; \boldsymbol{s}_k^{(t)}, \; \|\vec{\boldsymbol{x}}_{ij}\|, \; \vec{\boldsymbol{x}}_{ij} \cdot \vec{\boldsymbol{x}}_{ik} \right) \Big) \tag{3}$$

Hyperparameters: number of layers = 6, hidden channels = 512.

GearNet-Edge [12] GearNet-Edge is an SE(3) invariant architecture leveraging relational graph convolutional layers and edge message passing. The original GearNet-Edge formulation presented in Zhang et al. [12] operates on multirelational protein structure graphs making use of several edge construction schemes (k-NN, euclidean distance and sequence distance based). Our benchmark contains full capabilities for working with multirelational graphs but use a single edge type (i.e. $|\mathcal{R}| = 1$) in our experiments to enable more direct architectural comparisons.

The relational graph convolutional layer is defined for relation type r as:

$$\boldsymbol{s}_i^{(t+1)} := \boldsymbol{s}_i^{(t)} + \sigma\left(\mathrm{BN}\left(\sum_{r\in\mathcal{R}} \mathbf{W_r} \sum_{j\in\mathcal{N}_r(i)} \boldsymbol{s}_j^{(t)})\right)\right) \tag{4}$$

The edge message passing layer is defined for relation type r as:

$$\boldsymbol{m}_{(i,j,r_1)}^{(t+1)} := \sigma\left(\mathrm{BN}\left(\sum_{r\in|R|\prime} \mathbf{W}_r' \sum_{(w,k,r_2)\in\mathcal{N}_r'((i,j,r_1))} \boldsymbol{m}_{(w,k,r_2)}^{(t)}\right)\right) \tag{5}$$

$$\boldsymbol{s}_i^{(t+1)} := \sigma\left(\mathrm{BN}\left(\sum_{r\in|R|} \mathbf{W}_r \sum_{j\in\mathcal{N}_r(i)} \left(s_j^{(t)} + \mathrm{FC}(\boldsymbol{m}_{(i,j,r)}^{(t+1)})\right)\right)\right), \tag{6}$$

where $\mathrm{FC}(\cdot)$ denotes a linear transformation upon the message function.

Hyperparameters: number of layers = 6, hidden channels = 512.

C.3.2 Equivariant GNNs in Cartesian coordinates

EGNN [18] We consider E(3) equivariant GNN layers proposed by Satorras et al. [18] which updates both scalar features $\boldsymbol{s}_i$ as well as node coordinates $\vec{\boldsymbol{x}}_i$, as follows:

$$\boldsymbol{s}_i^{(t+1)} := f_2\left(\boldsymbol{s}_i^{(t)}, \sum_{j\in\mathcal{N}_i} f_1\left(\boldsymbol{s}_i^{(t)}, \boldsymbol{s}_j^{(t)}, \|\vec{\boldsymbol{x}}_{ij}^{(t)}\|\right)\right) \tag{7}$$

$$\vec{\boldsymbol{x}}_i^{(t+1)} := \vec{\boldsymbol{x}}_i^{(t)} + \sum_{j\in\mathcal{N}_i} \vec{\boldsymbol{x}}_{ij}^{(t)} \odot f_3\left(\boldsymbol{s}_i^{(t)}, \boldsymbol{s}_j^{(t)}, \|\vec{\boldsymbol{x}}_{ij}^{(t)}\|\right) \tag{8}$$

Hyperparameters: number of layers = 6, hidden channels = 512.

GCPNet [11] GCPNet is an SE(3) equivariant architecture that jointly learns scalar and vector-valued features from geometric protein structure inputs and, through the use of geometry-complete frame embeddings, sensitises its predictions to account for potential changes induced by the effects of molecular chirality on protein structure. In contrast to the original GCPNet formulation presented in Morehead et al. [11], the implementation we provide in the benchmark incorporates the architectural enhancements proposed in Morehead et al. [58] which include the addition of a scalar message attention gate (i.e., $f_a(\cdot)$) and a simplified structure for the model's geometric graph convolution layers (i.e., $f_n(\cdot)$). With geometry-complete graph convolution in mind, for node i and layer t, scalar edge features $\boldsymbol{s}_{e^{ij}}^{(t)}$ and vector edge features $\boldsymbol{v}_{e^{ij}}^{(t)}$ are used along with scalar node features $\boldsymbol{s}_{n^i}^{(t)}$ and vector node features $\boldsymbol{v}_{n^i}^{(t)}$ to update each node feature type as:

$$(\boldsymbol{s}_{m^{ij}}^{(t+1)}, \boldsymbol{v}_{m^{ij}}^{(t+1)}) := f_e^{(t+1)}\left((\boldsymbol{s}_{n^i}^{(t)}, \boldsymbol{v}_{n^i}^{(t)}), (\boldsymbol{s}_{n^j}^{(t)}, \boldsymbol{v}_{n^j}^{(t)}), (f_a^{(t+1)}(\boldsymbol{s}_{e^{ij}}^{(t)}), \boldsymbol{v}_{e^{ij}}^{(t)}), \boldsymbol{\mathcal{F}}_{ij}\right) \tag{9}$$

$$(\boldsymbol{s}_{n^i}^{(t+1)}, \boldsymbol{v}_{n^i}^{(t+1)}) := f_n^{(t+1)}\left((\boldsymbol{s}_{n^i}^{(t)}, \boldsymbol{v}_{n^i}^{(t)}), \sum_{j\in\mathcal{N}(i)} (\boldsymbol{s}_{m^{ij}}^{(t+1)}, \boldsymbol{v}_{m^{ij}}^{(t+1)})\right), \tag{10}$$

where the geometry-complete and chirality-sensitive local frames for node i (i.e., its edges) are defined as $\boldsymbol{\mathcal{F}}_{ij} = (\boldsymbol{a}_{ij}, \boldsymbol{b}_{ij}, \boldsymbol{c}_{ij})$, with $\boldsymbol{a}_{ij} = \frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}, \boldsymbol{b}_{ij} = \frac{\boldsymbol{x}_i \times \boldsymbol{x}_j}{\|\boldsymbol{x}_i \times \boldsymbol{x}_j\|}$, and $\boldsymbol{c}_{ij} = \boldsymbol{a}_{ij} \times \boldsymbol{b}_{ij}$, respectively.

Hyperparameters: number of layers = 6, hidden scalar channels = 128, hidden vector channels = 16.

C.3.3 Equivariant GNNs in Spherical coordinates

Tensor Field Network [19] Tensor Field Networks are E(3) or SE(3) equivariant GNNs that have been successfully used in protein structure prediction [2] and protein-ligand docking [17]. These models use higher order spherical tensors $\tilde{\boldsymbol{h}}_{i,l} \in \mathbb{R}^{2l+1 \times f}$ as node features, starting from order $l = 0$ up to arbitrary $l = L$. The first two orders correspond to scalar features $\boldsymbol{s}_i$ and vector features $\vec{\boldsymbol{v}}_i$, respectively. The higher order tensors $\tilde{\boldsymbol{h}}_i$ are updated via tensor products $\otimes$ of neighbourhood features $\tilde{\boldsymbol{h}}_j$ for all $j \in \mathcal{N}_i$ with the higher order spherical harmonic representations Y of the relative displacement $\frac{\vec{\boldsymbol{x}}_{ij}}{\|\vec{\boldsymbol{x}}_{ij}\|} = \hat{\boldsymbol{x}}_{ij}$:

$$\tilde{\boldsymbol{h}}_i^{(t+1)} := \tilde{\boldsymbol{h}}_i^{(t)} + \sum_{j \in \mathcal{N}_i} Y\left(\hat{\boldsymbol{x}}_{ij}\right) \otimes_{\boldsymbol{w}} \tilde{\boldsymbol{h}}_j^{(t)}, \tag{11}$$

where the weights $\boldsymbol{w}$ of the tensor product are computed via a learnt radial basis function of the relative distance, *i.e.* $\boldsymbol{w} = f\left(\|\vec{\boldsymbol{x}}_{ij}\|\right)$.

Hyperparameters: choice of symmetry = SO(3), number of layers = 4, spherical harmonic tensor order = 2, hidden irreps per tensor type = “64x0e + 64x0o + 8x1e + 8x1o + 4x2e + 4x2o". We were particularly interested in benchmarking the impact of higher order tensors and SO(3) equivariance.

MACE [20] MACE is a higher order E(3) or SE(3) equivariant GNN originally developed for molecular dynamics simulations. MACE provides an efficient approach to computing high body order equivariant features in the Tensor Field Network framework via Atomic Cluster Expansion: They first aggregate neighbourhood features analogous to equation 11 (the A functions in Batatia et al. [20] (eq. 9)) and then take $k - 1$ repeated self-tensor products of these neighbourhood features. In our formalism, this corresponds to:

$$\tilde{\boldsymbol{h}}_i^{(t+1)} := \underbrace{\tilde{\boldsymbol{h}}_i^{(t+1)} \otimes_{\boldsymbol{w}} \cdots \otimes_{\boldsymbol{w}} \tilde{\boldsymbol{h}}_i^{(t+1)}}_{k-1 \text{ times}}, \tag{12}$$

Hyperparameters: choice of symmetry = O(3), number of layers = 2, spherical harmonic tensor order = 2, hidden irreps per tensor type = “32x0e + 32x1o + 32x2e", body order = 4.

C.4 Pretraining Datasets

The benchmark contains several large corpora of both experimental and predicted structural data that can be used for pretraining or inference. We provide utilities for configuring supervised tasks and splits directly from the PDB [59]. Additionally, we build storage-efficient dataloaders for large pretraining corpora of predicted structures including the AlphaFoldDB and ESM Atlas. We believe our codebase will considerably reduce the barrier to entry for pretraining and working with large structure-based datasets.

C.4.1 Experimental Structures

PDB. We provide utilities for curating datasets directly from the Protein Data Bank [59]. In addition to using the collection in its entirety, users can define filters to subset and split the data using a combination of structural similarity, sequence similarity or temporal strategies. Structures can be filtered by length, number of chains, resolution, deposition date, presence/absence of particular ligands and structure determination method. The benchmark supports working with PDB structures in both `.pdb` and `.mmtf` format [60], which significantly reduces the requirements for data storage.

CATH. We provide the dataset derived from CATH 4.2 40% [61] non-redundant chains developed by Ingraham et al. [62] as an additional, smaller, pretraining dataset. These data are split based on random assignment of the CATH topology classifications based on an 80/10/10 split.

ASTRAL. ASTRAL [63] provides compendia of protein *domain* structures, regions of proteins that can maintain their structure and function independently of the rest of the protein. Domains typically exhibit highly-specific functions and can be considered structural building blocks of proteins.

C.4.2 Predicted Structures

We provide ready-to-go dataloaders for several large-scale collections of predicted structures derived from both AlphaFold2 [1] and ESMFold [21]. This is facilitated by FoldComp [23], a (minimally) lossy compression scheme for predicted protein structures. FoldComp stores protein structures as a collection of discretised dihedral and bond angles which can be used to reconstruct the whole structure using fixed bond lengths and canonical amino acid geometry. FoldComp achieves a disk-space reduction of almost an order of magnitude, describing a residue with only 13 bytes – down from 97 bytes per-residue in a traditional uncompressed format. Whilst lossy, this procedure results in 0.08 Å and 0.14 Å RMSD for backbone and all-atom reconstruction, making it highly suitable for pretraining tasks which use input representations complete up to the backbone. Furthermore, this lightweight format enables the dataloaders in the benchmark to read structures *directly from disk* with no pre-processing or caching required.

AlphaFoldDB Representative Structures. This dataset contains 2.27 million representative structures, identified through large-scale structural-similarity-based clustering of the 214 million structures contained in the AlphaFold Database [3] using FoldSeek [5]. We additionally provide a subset of this collection — the so-called dark proteome — corresponding to the 31% of the representative structures that lack annotations.

ESM Atlas, ESM High Quality. These datasets are compressed collections of predicted structures produced by ESMFold. ESM Atlas is the full collection of all 772m predicted structures for the MGnify 2023 release [64]. ESM High Quality is a curated subset of high confidence (mean pLDDT) structures from the collection.

C.5 Pretraining Tasks

The benchmark contains a comprehensive suite of pretraining tasks. Broadly, these tasks can be categorised into: masked-attribute prediction, denoising-based and contrastive learning-based tasks. In most cases, these tasks can be used as both a pretraining objective or as auxiliary tasks in a downstream supervised task.

Sequence Denoising. The benchmark contains two variations based on two sequence corruption processes $C(\tilde{\mathcal{S}}|\mathcal{S}, \nu)$ that receive an amino acid sequence $\mathcal{S} \in [0,1]^{|\mathcal{V}|\times 23}$ and return a sequence $\mathcal{S} \in [0,1]^{|\mathcal{V}|\times 23}$ with fraction ν of its positions corrupted. The first scheme is based on mutating a fraction of the residues to a random amino acid and tasking the model with recovering the uncorrupted sequence. The second is a masked residue prediction task, where a fraction of the residues are altered to a mask value and the model is tasked to recover the uncorrupted sequence.

Structure Denoising. We provide two structure-based denoising tasks: coordinate denoising and torsional denoising. In the coordinate denoising task, noise is sampled from a normal or uniform distribution and scaled by noise factor, $\nu \in \mathbb{R}$, and applied to each of the atom coordinates in the structure to ensure structural features, such as backbone or sidechain torsion angles, are also corrupted. The model is then tasked with predicting either the per-node noise or the original uncorrupted coordinates. For the torsional denoising variant, the noise is applied to the backbone torsion angles and Cartesian coordinates are recomputed using pNeRF [65] and the uncorrupted bond lengths and angles prior to feature computation. Similarly to the coordinate denoising task, the model is then tasked with predicting either the per-residue angular noise or the original dihedral angles.

Sequence-Structure Co-Denoising. This is a multitask formulation of the previously described structure and sequence denoising tasks, with separate output heads for denoising each modality.

Masked Attribute Prediction Tasks We use inverse folding (Section D.1) as a pretraining task. The benchmark additionally incorporates the distance, angle and dihedral angle masked-attribute prediction tasks proposed in Zhang et al. [12] as well as a backbone dihedral angle prediction task.

pLDDT Prediction. Protein structure prediction models typically provide per-residue pLDDT (predicted Local Distance Difference Test) scores as local confidence measures in the quality of the prediction shown to correlate well with disordered regions [66]. We formulate a self-supervised node-level regression task on predicted structures, somewhat analogous to structure quality assessment (QA), where the model is tasked with predicting the scaled per-residue pLDDT $y \in [0,1]$ values.

C.6 Downstream Tasks

We curate several structure-based and sequence-based datasets from the literature and existing benchmarks[2]. The tasks are selected to evaluate not only the *global* structure representational power of each method, but also to evaluate the ability of each method to learn informative *local* representations for residue-level prediction and annotation tasks.

The raw structures are, where possible and accounting for obsolescence, retrieved directly from the PDB (or another structural source) as several processed datasets used by the community discard full atomic coordinates in favour of retaining only C_α positions making them unsuitable for in-depth experimentation. This provides an entry point for users to apply a custom sequence of pre-processing steps, such as deprotonation or fixing missing regions which are common in experimental data.

The following downstream tasks are available in our benchmark. Detailed documentation including composition, splitting details, metrics, and data sheets [68] are available in Appendix D

Node-level Tasks

- CATH Inverse folding.
- PPI Site Prediction.
- Metal Binding Site Prediction.
- Post-Translational Modification Site Prediction.

Graph-level Tasks

- Fold Prediction.
- Gene Ontology Prediction.
- Reaction Class Prediction.
- Antibody Developability Prediction.

C.7 SE(3) Equivariant Noise Predictor

Similar to Zhang et al. [69], for structure-based denoising tasks we use an SE(3) equivariant noise predictor network to predict per-residue perturbations from SE(3) invariant scalar embeddings and corrupted atomic coordinates $\vec{\tilde{\boldsymbol{X}}}$. Each edge $\boldsymbol{e}_{ij}$ is featurised by concatenating the two adjoining scalar node representations $\tilde{\boldsymbol{s}}_i, \tilde{\boldsymbol{s}}_j$ and the euclidean distance between them $\|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|_2$. We use a two-layer MLP to produce a score $\boldsymbol{m}_{ij}$:

$$\boldsymbol{m}_{ij} = \mathrm{MLP}(\tilde{\boldsymbol{s}}_i, \tilde{\boldsymbol{s}}_j, \mathrm{MLP}(\|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|_2)). \tag{13}$$

Subsequently, Equation 13 is used to aggregate normalised directional edge vectors over the neighbourhood $\mathcal{N}_i$ of each node:

$$\boldsymbol{\epsilon}_\theta(\tilde{\mathcal{G}}) = \sum_{j \in \mathcal{N}_i} \boldsymbol{m}_{ij} \cdot \frac{\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j}{\|\tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_j\|_2} \tag{14}$$

C.8 Hyperparameter Selection

Given the large number of models and featurisation schemes, we try our best to do a consistent and fair hyperparameter search. We fix a consistently high number of layers and large hidden dimension across models, as we wanted to focus on scaling model size and dataset size via pre-training. We use the Fold Classification task to select the best learning rate and dropout per model and featurisation scheme for downstream tasks. While our best performing models sometimes do not outperform the best reported results in the literature, we have obtained these results in a consistent experimental setup. Our goal was to demonstrate the utility of our benchmarking framework and uncover the impact of architectural considerations such as featurisation schemes, geometric GNN models, and pre-training/auxiliary tasks under fair and rigorous experimental settings.

[2] To retain focus on *protein* representation learning, we deliberately exclude commonly-used tasks based on protein-small molecule interactions as it is hard to disentangle the effect of the small molecule representation and the potential for bias [67]

Protein Structure Encoders

1. We use a consistent batch size of 32.
2. For all models, we try to consistently use six layers, each with 512 hidden channels. For equivariant GNNs, we reduced the number of layers and hidden channels to fit 80GB of GPU memory on one NVIDIA A100 GPU.
3. For each encoder and featurisation, we search over learning rates: $0.00001, 0.0001, 0.0003, 0.001$ and select the best based on the validation performance on the fold classification task.

Output Heads

1. All primary output heads use a three-layer MLP with 512 as the hidden dimension.
2. For all auxiliary tasks we use a two-layer MLP with 128 as the hidden dimension.
3. For all structure denoising tasks we use a two-layer SE(3) equivariant noise predictor network (Section C.7). The message and distance MLPs each consist of two layers of 128 hidden units.
4. For each encoder and featurisation, we search over decoder dropout: $0.0, 0.1, 0.3, 0.5$, and select the best based on the validation performance on the fold classification task.

D Documentation for Datasets

Table 5: **Overview of supervised tasks and datasets.**

	Task	**Dataset Origin**	**Structures**	**# Train**	**# Validation**	**# Test**	**Metric**
Node-level	Inverse Folding	Ingraham et al. [62]	Experimental	3.9 M	105 K	180 K	Perplexity
	PPI Site Prediction	Gainza et al. [15]	Experimental	478 K	53 K	117 K	AUPRC
	Metal Binding Site Prediction		Experimental	1.1 M	13.7 K	29.8 K	Accuracy
	Post-Trans. Mod. Site Prediction	Yan et al. [70]	Predicted	44 K	2.4 K	2.5 K	ROC-AUC
Graph-level	Fold Prediction	Hou et al. [71]	Experimental	12.3 K	0.7 K	1.3/0.7/1.3 K	Accuracy
	Gene Ontology Prediction	Gligorijević et al. [14]	Experimental	27.5 K	3.1 K	3.0 K	F_{max}
	Reaction Class Prediction	Hermosilla et al. [27]	Experimental	29.2 K	2.6 K	5.6 K	Accuracy
	Antibody Dev. Prediction	Huang et al. [36]	Experimental	1.7 K	0.24 K	0.48 K	AUPRC

Below, we provide detailed documentation for each dataset included in our benchmark. Each dataset will be made available for download in processed and raw forms from Zenodo upon publication. Note that, for all datasets, we authors bear all responsibility in case of any violation of rights regarding the usage of such datasets, whether they were compiled from existing sources or curated from scratch.

D.1 CATH - Inverse Folding

This is a common protein engineering task where the goal is to recover an amino acid sequence given a structure up to backbone completeness. Formally, this is a node-level classification task where the model learns a mapping for each residue r_i to an amino acid type $y \in \{1, \dots, n\}$, where n is the vocabulary size ($n = 20$ for the canonical set of amino acids).

Note that inverse folding is a generic task that can be applied to any dataset. In the literature, it is commonly evaluated on the CATH dataset compiled by Ingraham et al. [62]. We additionally use inverse folding on AlphaFoldDB as a pretraining task.

- **Motivation** Several generative methods for protein design produce backbone structures that require the design of an associated sequence. As a result, inverse folding is an important part of *de novo* design pipelines for proteins.
- **Collection** For this dataset, we adopt the commonly-used CATH dataset originally compiled by Ingraham et al. [62].
- **Composition** The dataset consists of protein structures randomly split into training, validation and test sets such that proteins in different sets do not share the same CATH topology classification (i.e., CAT code).

- **Hosting** A preprocessed version of the dataset can be downloaded from the benchmark's Zenodo data record.
- **Licensing** We have released a preprocessed version of the dataset under a Creative Commons Attribution 4.0 International license. The original dataset is available under a Creative Commons Attribution 4.0 International license at `http://cathdb.info`.
- **Maintenance** We will announce any errata discovered in or changes made to the dataset using the benchmark's GitHub repository.
- **Uses** This dataset can be used for multilabel node classification tasks where a model learns a mapping for each residue r_i to an amino acid type $y \in \{1, \ldots, n\}$, where n is the vocabulary size (e.g., $n = 20$ for the canonical set of amino acids).
- **Metric** Perplexity.

D.2 MaSIF-Site - PPI Site Prediction

This task is a node-level binary classification task where the goal is to predict whether or not a residue is involved in a protein-protein interaction interface.

- **Motivation** Identifying protein-protein interaction (PPI) sites has important applications in developing improved protein-protein interaction networks and docking tools, providing biological context to guide protein engineering and target identification in drug discovery campaigns [72].
- **Collection** We adopt the dataset of experimental structures curated from the PDB by Gainza et al. [15] and retain the original splits, though we modify the labelling scheme to be based on inter-atomic proximity (3.5 Å), which can be user-defined, rather than solvent exclusion.
- **Composition** The dataset is curated from the PDB by preprocessing such as the presence of one of the seven specified ligands (e.g., ADP or FAD), clustering based on 30% sequence identity and random subsampling. It contains 1,459 structures, which are randomly assigned to training (72%), validation (8%) and test set (20%). 12 (Å) radius patches were extracted from the generated structures and a patch labelled as part of a binding pocket if its centre point was < 3 (Å) away from an atom of the corresponding ligand.
- **Hosting** The original dataset is made available by the authors on Zenodo.
- **Licensing** We have released a preprocessed version of the dataset under a Creative Commons Attribution 4.0 International license. The original dataset is available under an Apache 2.0 license at `https://github.com/LPDI-EPFL/masif/blob/master/LICENSE`.
- **Maintenance** We will announce any errata discovered in or changes made to the dataset using the benchmark's GitHub repository.
- **Uses** This dataset can be used for binary node classification tasks where the goal is to predict whether or not a residue is involved in a protein-protein interaction interface.
- **Metric** ROC-AUC.

D.3 ccPDB - Metal Binding Site Prediction

This is a binary node classification task where each residue is mapped to a label $y \in \{0, 1\}$ indicating whether the residue (or its constituent atoms) is within 3.5 (Å) of a user-defined metal ion or ligand heteroatom, respectively.

- **Motivation** Several proteins coordinate transition metal ions to carry out their functions. As such, predicting the binding sites of metal ions can elucidate the role of metal binding on protein function.
- **Collection** The dataset is constructed from experimental structures curated from the PDB, where binding site assignments for each residue are computed on-the-fly. While the benchmark supports this task on arbitrary subsets of the PDB and ligands, we provide the Zinc-binding dataset from Dürr et al. [73] specifically for this task.

- **Composition** The dataset is constructed by sequence-based clustering of the PDB at 30% sequence identity to remove sequence and structural redundancy. Clusters with a member shorter than 3000 residues, containing at least one zinc atom with resolution better than 2.5 (Å) determined by x-ray crystallography and not containing nucleic acids are used to compose the dataset. If multiple structures fulfill these criteria, the highest resolution structure is used. The train (2,085) / validation (26) / test (59) splits are constructed such that proteins in the validation and test sets have no partial overlap with any protein in the training data.
- **Hosting** A preprocessed version of the dataset can be downloaded from the benchmark's Zenodo data record.
- **Licensing** We have released a preprocessed version of the dataset under a Creative Commons Attribution 4.0 International license. The original dataset is freely available without a license at `https://academic.oup.com/database/article/doi/10.1093/database/bay142/5298333#130010908`.
- **Maintenance** We will announce any errata discovered in or changes made to the dataset using the benchmark's GitHub repository.
- **Uses** This dataset can be used for binary node classification tasks where each residue is mapped to a label $y \in \{0, 1\}$ indicating whether the residue (or its constituent atoms) is within 3.5 (Å) of a user-defined metal ion or ligand heteroatom, respectively.
- **Metric** Accuracy.

D.4 PTM - Post-Translational Modification Site Prediction

We frame prediction of post-translational modification (PTM) sites as a multilabel classification task where each residue is mapped to a label $y \in \{1, \ldots, 13\}$ distinguishing between modifications on different amino acids (e.g. phosphorylation on S/T/Y and N-linked glycosylation on N).

- **Motivation** Identifying the exact sites where post-translational modifications (PTMs) occur is essential for understanding protein behaviour and designing targeted therapeutic interventions.
- **Collection** We adopt a dataset of 48,811 AlphaFold2-predicted structures curated by Yan et al. [70], where each structure contains the PTM metadata necessary to construct residue-wise site prediction labels.
- **Composition** The dataset is split into training (43,907), validation (2,393) and test (2,511) sets based on 50% sequence identity and 80% coverage. In total, there are 240,090 PTMs present in the dataset compared to 3,391,208 residues where PTMs could be possible but are not present. The most common PTMs are phosphorylations on serine (93,734) and N-linked glycosylation at asparagine (59,143) which together account for around 70% of the PTMs.
- **Hosting** A preprocessed version of the dataset can be downloaded from the benchmark's Zenodo data record.
- **Licensing** We have released a preprocessed version of the dataset under a Creative Commons Attribution 4.0 International license. The original dataset is available under a Creative Commons Attribution 4.0 International license at `https://zenodo.org/record/7655709`.
- **Maintenance** We will announce any errata discovered in or changes made to the dataset using the benchmark's GitHub repository.
- **Uses** This dataset can be used for multilabel node classification tasks where each residue is mapped to a label $y \in \{1, \ldots, 13\}$ distinguishing between modifications on different amino acids (e.g., phosphorylation on S/T/Y and N-linked glycosylation on N).
- **Metric** ROC-AUC.

D.5 FOLD - Fold Prediction

This is a multiclass graph classification task where each protein, $\mathcal{G}$, is mapped to a label $y \in \{1, \ldots, 1195\}$ denoting the fold class.

- **Motivation** The utility of fold prediction is that it serves as a litmus test for the ability of a model to distinguish different structural folds. It stands to reason that models that perform poorly on distinguishing fold classes likely learn limited or low-quality structural representations.
- **Collection** We adopt the fold classification dataset originally curated from SCOP 1.75 by [71].
- **Composition** This dataset provides three different test sets stratified based on topological similarity: Fold, in which proteins originating from the same superfamily are absent during training; Superfamily, in which proteins originating from the same family are absent during training; and Family, in which proteins from the same family are present during training.
- **Hosting** A preprocessed version of the dataset can be downloaded from the benchmark's Zenodo data record.
- **Licensing** We have released a preprocessed version of the dataset under a Creative Commons Attribution 4.0 International license. The original dataset is available under a Creative Commons Attribution 4.0 International license at `https://academic.oup.com/bioinformatics/article/34/8/1295/4708302`.
- **Maintenance** We will announce any errata discovered in or changes made to the dataset using the benchmark's GitHub repository.
- **Uses** This dataset can be used for multilabel graph classification tasks where each protein, $\mathcal{G}$, is mapped to a label $y \in \{1, \ldots, 1195\}$ denoting the fold class.
- **Metric** Micro-averaged accuracy.

D.6 GO - Gene Ontology Prediction

This is a multilabel classification task, assigning functional Gene Ontology (GO) annotation to structures. GO annotations are assigned within three ontologies: biological process (BP), cellular component (CC) and molecular function (MF).

- **Motivation** Predicting protein function in the form of functional annotations such as gene ontology (GO) terms has important applications in protein analysis and engineering, providing researchers with the ability to cluster functionally-related structures or to guide protein generation methods to design new proteins with desired functional properties.
- **Collection** We adopt the dataset of experimental structures originally curated from the PDB by Gligorijević et al. [14].
- **Composition** We retain the original multi-cutoff based dataset splits proposed by [14], with cutoff at 95% sequence similarity.
- **Hosting** A preprocessed version of the dataset can be downloaded from the benchmark's Zenodo data record.
- **Licensing** We have released a preprocessed version of the dataset under a Creative Commons Attribution 4.0 International license. The original dataset is available under a Creative Commons Attribution 4.0 International license at `https://www.nature.com/articles/s41467-021-23303-9`.
- **Maintenance** We will announce any errata discovered in or changes made to the dataset using the benchmark's GitHub repository.
- **Uses** This dataset can be used for multilabel graph classification tasks, assigning a functional Gene Ontology (GO) annotation to protein structures.
- **Metric** $\mathrm{F}_{\max}$ score.

D.7 EC Reaction - Reaction Class Prediction

This is a multiclass graph classification task where each protein, $\mathcal{G}$, is mapped to a label $y \in \{1, ..., 384\}$ denoting which class of reactions a given protein catalyzes; all four levels of the EC assignment are employed to define the reaction class label.

- **Motivation** As proteins' reaction classifications are based on their enzyme-catalyzed reaction according to all four levels of the standard Enzyme Commission (EC) number, methods that predict such classifications can help elucidate the function of newly-designed proteins as they are developed.
- **Collection** We adopt the reaction class prediction dataset originally curated from the PDB by Hermosilla et al. [27].
- **Composition** The dataset is split on the basis of sequence similarity using a 50% threshold.
- **Hosting** A preprocessed version of the dataset can be downloaded from the benchmark's Zenodo data record.
- **Licensing** We have released a preprocessed version of the dataset under a Creative Commons Attribution 4.0 International license. The original dataset is available under an MIT license at `https://github.com/phermosilla/IEConv_proteins/blob/master/LICENSE`.
- **Maintenance** We will announce any errata discovered in or changes made to the dataset using the benchmark's GitHub repository.
- **Uses** This dataset can be used for multilabel graph classification tasks where each protein, $\mathcal{G}$, is mapped to a label $y \in \{1, ..., 384\}$ denoting which class of reactions a given protein catalyzes.
- **Metric** Accuracy.

D.8 TDC - Antibody Developability Prediction

Therapeutic antibodies must be optimised for favourable physicochemical properties in addition to target binding affinity and specificity to be viable development candidates. Consequently, we frame prediction of antibody developability as a binary graph classification task indicating whether a given antibody is developable.

- **Motivation** From a benchmarking perspective, predicting the developability of a given antibody is important as it enables targeted performance assessment of models on a specific (immunoglobulin) fold, providing insight into whether general-purpose structure-based encoders can be applicable to fold-specific tasks.
- **Collection** We adopt the antibody developability dataset originally curated from SabDab [74] by Chen et al. [75].
- **Composition** This dataset contains 2,426 antibodies that have both sequences and PDB structures available, where each example contains both a heavy chain and a light chain with resolution < 3 (Å). Labels are based on thresholding the developability index (DI) ([76]) as computed by BIOVIA's platform ([77]), which relies on an antibody's hydrophobic and electrostatic interactions.
- **Hosting** A preprocessed version of the dataset can be downloaded from the benchmark's Zenodo data record.
- **Licensing** We have released a preprocessed version of the dataset under a Creative Commons Attribution 4.0 International license. The original dataset is available under a Creative Commons Attribution 3.0 Unported license at `https://tdcommons.ai/single_pred_tasks/develop/#sabdab-chen-et-al`.
- **Maintenance** We will announce any errata discovered in or changes made to the dataset using the benchmark's GitHub repository.
- **Uses** This dataset can be used for binary graph classification tasks indicating whether a given antibody is developable.
- **Metric** AUPRC.