

A APPENDIX

A.1 GROUP THEORY PRELIMINARIES

Definition A.1 (Group). A group is a set G equipped with a binary operation $\cdot : G \times G \rightarrow G$ obeying the following axioms:

- for all $g_1, g_2 \in G$, $g_1 \cdot g_2 \in G$ (closure).
- for all $g_1, g_2, g_3 \in G$, $g_1 \cdot (g_2 \cdot g_3) = (g_1 \cdot g_2) \cdot g_3$ (associativity).
- there is a unique $e \in G$ such that $e \cdot g = g \cdot e = g$ for all $g \in G$ (identity).
- for all $g \in G$ there exists $g^{-1} \in G$ such that $g \cdot g^{-1} = g^{-1} \cdot g = e$ (inverse).

Definition A.2 (Group invariant functions). Let G be a group acting on vector space V . We say that a function $f : V \rightarrow \mathbb{R}$ is G -invariant if $f(g \cdot x) = f(x) \ \forall x \in V, g \in G$.

Definition A.3 ((Left) Group Action). For a group G with identity element e , and X is a set, a (left) group action α of G on X is a function $\alpha : G \times X \rightarrow X$ that satisfies the following two conditions:

1. Identity: $\alpha(e, x) = x, \forall x \in X$
2. Compatibility: $\alpha(g, \alpha(h, x)) = \alpha(gh, x)$

We will use a short hand of $g \cdot x$ for $\alpha(g, x)$ when the action being considered is clear from context.

A.2 EXAMPLE DEMONSTRATING NON GROUP STRUCTURE OF A SET OF PROTEIN CONFORMATIONS

Consider X to be a protein with n atoms and m amino acids and let the set of viable conformations of X as $\{X_1, X_2, \dots, X_i, \dots, X_p\}$. Let X_1 be the conformation available in our dataset.

In each step of the transition, only a few atoms ($\ll n$) (atoms in a single amino acid of the entire protein- where the protein is made up of multiple hundreds of amino acids (m) in general) are subjected to an action from the $SO(3)$ group here. The positions of all other atoms (outside that amino acid) in the side chain remain unaltered. So, in the $\mathbb{R}^{n \times 3 \times 3}$ matrix – most of the 3×3 entries are the identity matrix of 3 dimension.

Let $T_1^i \in \mathbb{R}^{n \times 3 \times 3}$ where $i \in \{1, 2, \dots, p\}$ be the transformation which yields conformation X_i from X_1 .

Now consider T_1^2 (subscript of 1 since we start conformation X_1) which takes X_1 to X_2 and T_1^3 which takes X_1 to X_3 where T_1^2 and T_1^3 , do not act on the same amino acid in the protein. Now, however, consider the case where performing T_1^2 and T_1^3 (or the other way around) sequentially would result in a case where atoms in two different amino acids would overlap (or come too close to each other causing steric repulsion) - therefore resulting in a non viable conformation i.e. a composition of T_1^2 and T_1^3 acting on X_1 , would not be present in the set of all allowed transformations $T_1 = \{T_1^1, \dots, T_1^i, \dots, T_1^p\}$. Therefore the set T_1 is not closed and doesn't form a group.

Also for two different conformations X_1 and X_2 , their allowed transformation sets T_1, T_2 will not be identical (in the above example $T_1^3 \notin T_2$). It also easy to construct a case where $\bigcup_{i=1}^p T_i$ is not a group and all actions from this group acting on any given X_i will not necessarily result in viable/ valid conformation.

A.3 FLEXIBILITY ALLOWED BY OUR PROPOSED MODEL

Our proposed model allows every node in the directed tree to be rotated about its parents. For example, for the side chain shown in Figure 1 (Main Paper), the allowed flexibility is Figure 3.

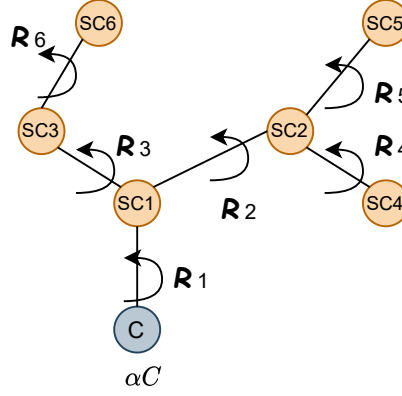


Figure 3: Maximum flexibility allowed by our candidate sampling process when the group associated with every node in the tree is $SO(3)$ (the special orthogonal group in 3 dimensions). While not candidate is likely to be accepted such a candidate generation process provides the flexibility for every node to be rotated about its immediate parent while preserving bond lengths.

A.4 PROOFS OF PROPOSITIONS

First, we restate and prove Proposition 3.3

Proposition A.4. *Given the CSMC Φ_p from Definition 3.2 whose transitions are governed by κ which is implicitly defined by Algorithm 1 as described above. For any pair of conformers $c_p, c'_p \in C_p$, there exists $\tau_p < \infty$, independent of c_p , such that $P_{\Phi_p}^{\tau_p}(c_p, c'_p) > 0$, where $P_{\Phi_p}^{\tau_p}$ is the τ_p step transition probability.*

Proof. Proof by construction. We prove the proposition by showing that one can construct a path $(c_p^{(1)} = c_p, \dots, c_p^{(t)} = c'_p)$ such that $c_p^{(i)} \in C_p$ and $\kappa(c_p^{(i+1)} | c_p^{(i)}) > 0$, for all $0 < i < t$, and $t \leq T_p$. The trivial case where $c_p \equiv c'_p$ is proved since every group contains the identity element — sampling the identity element for every node in the directed tree yields the same conformation. Since we consider only non backbone transforming conformations, for the non trivial case, a maximum of $m - 4n$ atoms can differ in positions between any two conformations - where m is the number of atoms in the protein and n is the the number of amino acids in the protein n . Both m, n are finite and we are dealing with continuous conformers (and continuous group actions about every node — groups are closed under their associated binary action, and $SO(3)$ is path connected). So we can traverse between conformers (until we reach the desired conformer) sequentially in a finite number of steps, by using the constructed directed forest - selecting a amino acid (which doesn't violate the viability), fixing the positions of all other amino acids in the protein and rotating the side chain atoms in a single conformer to the final desired state. While this process may result in some side chains being visited multiple times (due to viability constraints), considering continuous conformers and the $SO(3)$ group (which is path connected) ensures we will never reach a state of deadlock. The second condition is satisfied because every group action has an inverse and we only use transformations from $SO(3)$ for every node in the directed trees. \square

Next, we restate and and prove Proposition 3.4

Proposition A.5. *The CSMC Φ_p defined in Definition 3.2 is uniformly ergodic if Proposition 3.3 is satisfied. Specifically there exists a unique steady state distribution π_p such that for all $c_p \in C_p$, $\|P_{\Phi_p}^n(c_p, \cdot) - \pi_p(\cdot)\| \leq C R^n$, where $C < \infty$ and $R < 1$ are constants that depend on Φ_p , $P_{\Phi_p}^n$ is the n step transition probability and $\|\cdot\|$ is the ℓ_1 norm.*

Proof. By Proposition 3.3, Φ_p satisfies Doeblin’s condition as defined in page 396 of Meyn & Tweedie (2012) which states that for $c_p, c'_p \in C_p$, $P_{\Phi_p}^{T_p}(c_p, c'_p) > \epsilon$ for some $\epsilon > 0$ ¹. The uniform ergodicity then holds due to Theorems 16.2.3 and 16.2.1 from Meyn & Tweedie (2012). \square

Next, we restate and prove Proposition 4.2. We also restate the required assumptions for the proposition.

Assumption A.6. We make the following assumptions:

1. For any $\theta \in \Theta$ and $x_j^k \in S_j$, the function f is differentiable $\forall j$
2. $\sup_{\theta \in \Theta, x_j^k \in S_j} \{|\nabla_{\theta} \rho \circ f(x_j^k)|\} < +\infty$ i.e. the gradients are bounded.
3. $\forall x_j^k \in S_j, \forall \theta_1, \theta_2 \in \Theta, |\nabla_{\theta_1} \rho \circ f(x_j^k) - \nabla_{\theta_2} \rho \circ f(x_j^k)| < L|\theta_1 - \theta_2|$ for some $L \geq 0$ i.e., the gradients are L -Lipschitz.
4. $\mathbb{E}_{x_j^k \sim \pi_j} [\nabla_{\theta} \rho \circ f(x_j^k)] = \nabla_{\theta} \mathbb{E}_{x_j^k \sim \pi_j} [\rho \circ f(x_j^k)]$

Proposition A.7. Let the step sizes satisfy (5) and the function parameters θ be updated as (6) and Assumption 4.1 hold, then the MCGD optimization enjoys properties of almost sure convergence to the optimal θ .

Proof. Given that each protein has an associated time homogeneous Markov Chain with a unique steady state, independent of other proteins, the set of proteins in a mini-batch also form a Markov chain with a unique steady state. We then leverage Corollary 2 (Page 12) of Sun et al. (2018) along with Proposition 4.2 to ensure almost sure convergence to the optimal θ . \square

A.5 EXTENDED RELATED WORK

Here, we elaborate on the related works section in Section 6

Group Equivariant and Invariant Neural Networks: Group equivariant and invariant neural networks (Cohen & Welling, 2016; Lenssen et al., 2018; Kondor & Trivedi, 2018; Finzi et al., 2020; Hutchinson et al., 2021; Fuchs et al., 2020, 2021; Dehmamy et al., 2020) help capture discrete and continuous groups symmetries of elements (e.g. images, point clouds). In this work, we learn representations which are invariant to symmetries which are not just groups, but to input dependent sets of transformations. To the best of our knowledge, our work is the first to consider conditional (input dependent) invariances.

Prior works on learning invariant models have leveraged Monte Carlo procedures (Finzi et al., 2020; Murphy et al., 2019b) to learn to be invariant to transformations of the input. Alternatively, our work constructs a Markov Chain with a unique steady state and leverages MCGD (Sun et al., 2018) to make it computationally tractable. Secondly, the aforementioned approaches are limited to being invariant to transformations from any specified Lie group with a surjective exponential map/permutation group, while our work is not limited to groups. Thirdly, our theory ensures that every example/object in the dataset can have a different set of input dependent, conditional transformations - and in fact can be seen as a generalization of the above works. In fact, the ablation study that we perform (Results provided in Appendix A.8 - Table 4) uses a Monte Carlo estimator and our MCMC procedure yields better performance than the Monte Carlo estimator.

While group equivariant neural network capture global symmetries, local symmetries of manifold spaces can be captured via gauge equivariant networks (Cohen et al., 2019; De Haan et al., 2020). Gauge symmetries require the manifold to be smooth - which is not the case for proteins. Moreover, different proteins have different sets of viable conformations, which would not be able to captured by standard gauge equivariant neural networks.

(Lenssen et al., 2018; Gerken et al., 2021; Bronstein et al., 2021) provide a complete review of the theoretical aspects and a wide variety of applications of group equivariant neural networks. A more comprehensive theoretical analysis of input dependent conditionally invariant neural networks is planned for future work.

¹We note that this is a simplified version of the actual statement which is defined on the σ -algebra over C_p denoted by $\sigma(C_p)$. Our proof holds when $c'_p \in \sigma(C_p)$

Graph Neural Networks: Graph Neural Networks (GNNs) (Kipf & Welling, 2016; Hamilton et al., 2017; Battaglia et al., 2018; Xu et al., 2018) have gained renewed focus over the past few years and have found applications in recommender systems, biology, chemistry, and many other real world problems which can be formulated as graphs and currently serve as the state of the art in majority of node and graph classification/ regression tasks. Graph neural networks work on the principles of permutation equivariance/ invariances (Murphy et al., 2019a) (also groups) and have exploited a message passing framework to learn powerful and expressive representations of nodes/ graphs. Both molecular graphs (both small molecules and macromolecules) as well as graphs based on intra molecular distances have been used with GNNs, to achieve state of the art for many molecular datasets and tasks (Hu et al., 2020; Morris et al., 2020). Here, we leverage three graph based neural networks as baselines for our model.

Group Equivariant Graph Neural Networks: Group Equivariant GNNs combine continuous symmetries (lie groups such as $SE(3)$, $E(3)$, $SO(3)$) with permutation equivariances and has found applications with resounding success on small and large molecules (Anderson et al., 2019; Klicpera et al., 2020; Satorras et al., 2021; Batzner et al., 2021). However, the methods are only able to capture rigid body characteristics of molecules and while capturing the above lie group symmetries is also able to capture input dependent transformations. Employing (Farina & Slade, 2021), would make the neural network excessively invariant and allow the protein to be more flexible (allows unviable conformations) than it truly is. More recently, they have also been applied to learning representations of proteins, which is discussed below.

Monte Carlo and MCMC Methods for Sampling Protein Conformations: There has been a lot of prior work – (Boomsma et al., 2013; Olsson et al., 2013; Antonov et al., 2016; Irback & Mohanty, 2006; Vitalis & Pappu, 2009) which sample protein conformations by internal coordinate transformations. However, The goals of the existing MCMC methods are significantly different compared to ours. The existing methods define/inherit a distribution over the conformations (majorly based on the properties of the bonds, etc.) and then aim to sample highly probable conformations from this distribution. Our method, is much simpler and only requires that the chain being used is ergodic and is invariant to the actual form of the distribution. We note that the existing Markov chains can seamlessly be used as drop-in replacements to sample conformations as part of our framework as long as the MCMC is ergodic. We consider studying the impact of different MCMC methods (which sample from different distributions) and their influence on the performance in different tasks as important future work.

Neural Networks for Representation Learning of Proteins: Protein representation has gained a lot of attention especially with the tremendous successes of Alphafold and Alphafold2. A variety of neural network architectures including 3D CNNs, LSTM’s and Transformers (treating the protein as a sequence) as well as graph neural networks have been employed to exploit the rigid body symmetries of proteins (Karimi et al., 2019; Pagès et al., 2019; Ingraham et al., 2019; Strokach et al., 2020; Baldassarre et al., 2021; Hermosilla et al., 2021; Jing et al., 2020; 2021). In this work, while we use GNNs and Group Invariant GNNs as a part of the model, we note that we can equally replace them with CNNs, LSTMs, Transformers and other models used for proteins without any change in the underlying theory.

Generative Models: Protein conformation generation models (Mansimov et al., 2019; Simm et al., 2020; Ganea et al., 2021; Xu et al., 2021b;a; Shi et al., 2021; Luo et al., 2021) have also recently gained attend where the goal of the model is to predict 3d structure of molecules given input 2d structure - our objective in this work is completely different, but can be used to improve predictions of the aforementioned models. While our model is explicitly not a generative model, our framework can be leveraged towards generative modeling with the help of tools such as noise outsourcing (Chapter 6) (Kallenberg, 2006) and we see this as important future work.

Non-Rigid Body Dynamics: Non-Rigid Body Dynamics of objects has long been studied both by physicists and in the fields of computer vision to understand and capture the geometric deformations of objects (Taylor et al., 2010; Masci et al., 2015). To the best of our knowledge, there exists no prior work in deep learning which captures the non rigidity of protein molecules (which

cannot be modeled as C^k manifolds). As important future work, we would like to study the impact of leveraging input dependent conditional invariances for modeling other geometric objects (which are C^k manifolds) as well as images and robotics (e.g. the symmetries for a humanoid is different from that of a tractor).

Unrelated Work with similar names: Non classical and conditional symmetries of solutions to ODE’s, PDE’s have been discussed in the past - these works while they share a similar title, have very little in common as we are not dealing with jet spaces or manifolds (Joseph, 1968; Fushchich & Zhdanov, 1992; Olver & Vorob’ev, 1996).

A.6 DETAILS ABOUT DATASETS AND TASKS

In this section, we describe briefly each of the datasets (and their associated tasks). Information about the splits and license information is provided in Table 2

PSR: This task utilizes data from the structural models submitted to the Critical Assessment of Structure Prediction competition (CASP - Kryshchuk et al., 2019) - a blind protein structure prediction competition) to rank protein structures from the experimentally determined structure of the protein. The problem is formulated as a regression task, where we predict the global distance test of each structural model from the experimentally determined structure. As prescribed by the dataset authors, the dataset is split by competition years.

MSP: The goal of this task is to identify mutations that stabilize a protein’s interactions which forms an important step towards the design of new proteins. This task is significant as probing mutations experimentally techniques are labor-intensive. Atom3D (Townshend et al., 2020) derives this dataset by collecting single-point mutations from the SKEMPI database (Jankauskaitė et al., 2019) and model each mutation into the structure to produce a mutated structure. The learning problem is then formulated as a binary classification task where the goal is to predict whether the stability of the complex increases as a result of the mutation. We employ the same splits as suggested by the dataset authors wherein the protein complexes are split such that no protein in the test dataset has more than 30% sequence identity with any protein in the training dataset.

LBA: This task deals with the problem of predicting the strength (affinity) of a candidate drug molecule’s interaction with a target protein. The dataset is constructed using the PDDBind database (Wang et al., 2004; Liu et al., 2015), a curated database containing protein-ligand complexes from the PDB and their corresponding binding strengths (affinities). The task is formulated as a regression task with the goal to predict $pK = -\log_{10}(K)$, where K is the binding affinity in Molar units. The splits are created such that no protein in the test dataset has more than 30% sequence identity with any protein in the training dataset.

LEP: The shape of protein impacts whether a protein is in an on or off state which plays an important role in predicting the shape a protein will favor during drug design. This dataset is obtained by curating proteins from several families with both “active” and “inactive” state structures, and model in 527 small molecules with known activating or inactivating function using the program Glide (Friesner et al., 2004). The task is formulated as a binary classification task where the goal is to predict whether a molecule bound to the structures will be an activator of the protein’s function or not. We use the same split as recommended by the ATOM3D authors.

Table 2: Summary of the datasets

Task	# Train	# Val	# Test	Original Source	License
MSP	2864	937	247	SKEMPI (Jankauskaitė et al., 2019)	Creative Commons CC-BY
LBA	3563	448	452	PDDBind (Wang et al., 2004)	Creative Commons NonCommercial-NoDerivs (CC-BY-NC-ND)
LEP	304	110	104	PDB (Berman et al., 2000)	Creative Commons CC-BY
PSR	25400	2800	16099	CASP (Kryshchuk et al., 2019)	Creative Commons CC-BY

A.7 EXPERIMENTAL SETUP

The code for the baseline models (GVP-GNN (Jing et al., 2021), E(N) GNN (Satorras et al., 2021) and GNN(GCN) (Townshend et al., 2020; Kipf & Welling, 2016)) were used as provided by the

authors (licenses as dictated by the code authors). Our conformer invariance implementation is in PyTorch using Python 3.8. We also leverage networkx to create the directed forests. For all three models we tune the hyperparameters – learning rate ($\in \{0.1, 0.01, 0.001, 0.0001\}$) and mini batch size ($\in \{4, 8, 16, 32, 64\}$). For the E(n) GNN model - since there have been no previous models for the aforementioned protein tasks - we also tune the number of GNN layers ($\in \{4, 5, 6, 7\}$) as a hyperparameter. The experiments were all performed on Tesla V100 GPU’s. For more details refer to the code provided.

A.8 ADDITIONAL RESULTS

In Table 3, we present the results, including additional datasets and tasks, than presented in the main paper. In the LEP task, the proposed addition outperforms the baseline models for the E(n) GNN and the GVP-GNN, but not for the GNN(GCN). The LEP result for the GNN(GCN) is an oddity here, where a GNN which doesn’t incorporate any rigid body transformations outperforms all other models.

In Table 4, we present an ablation study, where we provide a strategy where all the transformations are created from “gold standard” X_p rather than via the MCMC method, i.e., the MCMC is restarted at every epoch during training. From the table, we note that the MCMC method tends to outperform the non MCMC method, which can be attributed to the guarantees it provides to the learning framework.

Table 3: GNN(GCN), GVP-GNN, E(N) GNN - Baseline vs Conformation Invariant Strategies for multiple different tasks on proteins from the ATOM3D dataset. Corresponding to the metric, \uparrow indicates that higher is better, while \downarrow indicates that lower is better. Bold values indicate best results for a given row. The values for GNN were obtained from Townshend et al. (2020) and for the GVP-GNN from Jing et al. (2021). Gray colored cells indicates that the augmented model outperforms the baseline model.

Task	Metric	Baseline (GNN(GCN))	MCMC Augmented GNN (Ours)	Baseline (GVP GNN)	MCMC Augmented GVP-GNN (Ours)	Baseline (E(N) GNN)	MCMC Augmented E(N) GNN (Ours)
PSR	Global $R_s \uparrow$	0.755 \pm 0.004	0.761 \pm 0.004	0.845 \pm 0.004	0.852 \pm 0.006	0.827 \pm 0.004	0.852 \pm 0.004
LEP	AUROC \uparrow	0.740 \pm 0.010	0.672 \pm 0.012	0.628 \pm 0.055	0.704 \pm 0.039	0.677 \pm 0.014	0.714 \pm 0.005
LBA	RMSE \downarrow	1.570 \pm 0.025	1.519 \pm 0.022	1.594 \pm 0.073	1.435 \pm 0.007	1.392 \pm 0.001	1.384 \pm 0.011
MSP	AUROC \uparrow	0.621 \pm 0.009	0.662 \pm 0.008	0.680 \pm 0.015	0.857 \pm 0.049	0.652 \pm 0.006	0.843 \pm 0.037

Table 4: GVP-GNN, GNN (GCN) - Baseline vs Ablation vs Conformation Invariant Strategies for four different tasks on proteins from the ATOM3D (Townshend et al., 2020) dataset. Corresponding to the metric, \uparrow indicates that higher is better, while \downarrow indicates that lower is better. Bold values indicate best results for a given row. Gray colored cells indicates that the augmented model outperforms the baseline model.

Task	Metric	Baseline (GVP GNN) Jing et al. (2021)	Non MCMC Augmented GVP-GNN (Ablation)	MCMC Augmented GVP-GNN (Ours)	Baseline (GNN) Townshend et al. (2020)	Non MCMC Augmented GNN (Ablation)	MCMC Augmented GNN (Ours)
PSR	Global $R_s \uparrow$	0.845 \pm 0.004	0.806 \pm 0.011	0.852 \pm 0.006	0.755 \pm 0.004	0.766 \pm 0.001	0.761 \pm 0.004
LEP	AUROC \uparrow	0.628 \pm 0.055	0.739 \pm 0.060	0.704 \pm 0.039	0.740 \pm 0.010	0.657 \pm 0.008	0.672 \pm 0.012
LBA	RMSE \downarrow	1.594 \pm 0.073	1.635 \pm 0.007	1.435 \pm 0.007	1.570 \pm 0.025	1.520 \pm 0.022	1.519 \pm 0.022
MSP	AUROC \uparrow	0.680 \pm 0.015	0.799 \pm 0.016	0.857 \pm 0.049	0.621 \pm 0.009	0.610 \pm 0.021	0.662 \pm 0.008

A.9 CASE AGAINST USING AVERAGE REPRESENTATIONS IN TRAINING AND FOR REQUIRING A SINGLE INVARIANT REPRESENTATION

The case for the requirement of a single conformer invariant representations – can be seen from the fact that different protein conformations, say X_1, X_2 of the same protein X , may be seen during train and test phases. Without conformer invariant representations - this may lead to different representations and therefore different predictions for the same protein (bad).

One may argue, that an approximate conformer invariant neural network can be learned by averaging representations of multiple Monte Carlo conformations during training. However, this is computationally expensive as this would need to back-propagate and update parameters for multiple conformations for every protein in every epoch – bad as this leads to an exponential overhead. On the other hand, our procedure with MCGD, in every epoch uses only one conformation from our Markov Chain and still ensures convergence to optimal parameters. It is important to note that, during inference we still do average representations over multiple conformations (from our Markov chain) to output conformer invariant representations - which again due to MCGD training procedure and the unique steady state of our Markov Chain, ensures conformer invariant representations are achieved – which yield better performance than current state of the art on multiple datasets and tasks. As stated in the main paper, our framework can work well with other Markov chains as well. We chose the proposed chain purely because it is simpler than the existing methods (in that it doesn't require a distribution over conformations to be assumed) and as such is easier to sample from.

A.10 TRAINING TIME

In Table 5, we present the training time (for 100 epochs) for each of the baseline models as well as well for models with our proposed conformer invariance framework addition. From the table, we note that, on an average the increase in training time over 100 epochs is \approx 3-4 minutes which is negligible in comparison to the training time (without the proposed framework addition).

Table 5: Training Time (in minutes for 100 epochs – rounded to closest integer) - GNN(GCN), GVP-GNN, E(N) GNN - Baseline vs Conformation Invariant Strategies for multiple different tasks on proteins from the ATOM3D dataset. Lower is better.

Task	Baseline (GNN[GCN])	MCMC Augmented GNN (Ours)	Baseline (GVP GNN)	MCMC Augmented GVP-GNN (Ours)	Baseline (E(N) GNN)	MCMC Augmented E(N) GNN (Ours)
PSR	184	190	1112	1118	485	492
LEP	6	8	8	9	10	11
LBA	26	28	34	38	59	62
MSP	58	61	45	49	125	128