Limitations

Despite the promising results, there are a few limitations to our study:

Scope of Application: While DR-Label is primarily tailored for a distinct geometric deep learning task—predicting target geometry from an initial state—it largely employs catalysis systems as its primary point of reference. This limits its direct applicability to other geometric learning pursuits such as protein docking (Méndez-Lucio et al. 2021), fluid simulation (Sanchez-Gonzalez et al. 2020), and various macroscopic engineering tasks utilizing geometric deep learning or generic graph neural networks (Satorras, Hoogeboom, and Welling 2021; Pfaff et al. 2020; Wang et al. 2022). Nonetheless, we are enthusiastic about broadening the horizons of our model to encompass a more diverse set of tasks in subsequent studies.

Precision Issues: Occasionally, the projection magnitude on edges is diminutive, which using half-precision can frequently lead to overflow problems. To address this, we are working on a version that employs mixed-precision training, aiming to optimize computational resources without compromising accuracy.

Graph structure variations

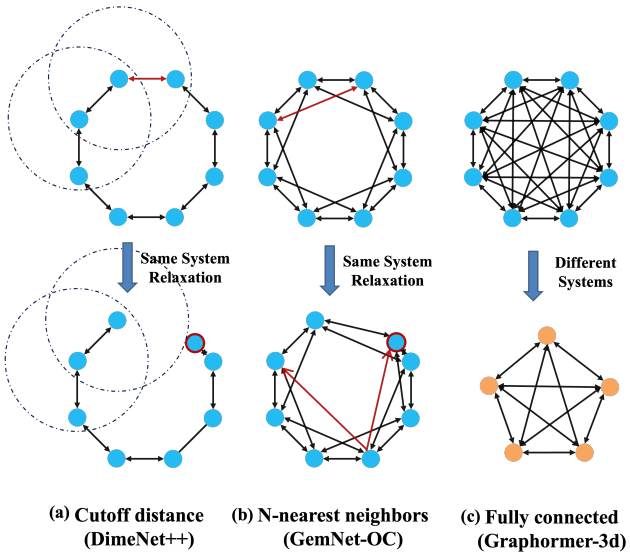

Figure 5: Graph construction algorithms for various models. When atomic systems undergo positional shifts, the edge set varies significantly for (a) cut-off distance based graph and (b) N-nearest neighbors graph. (c) The fully connected graph demonstrates significant differences in node degrees across distinct atomic systems.

As discussed in the limitation of linearly aggregated node prediction, atomic system graph construction algorithms lack unification. Here, we offer representative examples of graph constructions for popular models in Figure 5. Algorithms shown include (a) the cut-off distance-based graph construction used in (Gasteiger et al. 2020), (b) the N-nearest neighbor graph from (Gasteiger et al. 2022), and (c) the fully connected graph employed by (Shi et al. 2022).

The first row reveals that the same 8-atom atomic system can yield markedly different graph structures and statistics depending on the algorithm used. For Figure 5 (a) and (b), we observed that re-constructing the graph at different relaxation stages can result in significantly altered topologies compared to the original structure. Figure 5 (c) highlights that in fully connected graphs, changes in total atomic number directly affect node degrees. All these issues result in volatile graph structures and statistics when representing atomic system relaxation processes, augmenting the edge representation ambiguity and leading to node predictions that are highly sensitive to graph variations.

Pseudocode of Project and Fit

For clarity, we provide an in-depth pseudocode description of the Project and Fit algorithm. Algorithm 1 delineates the deconstruction phase, where node-wise displacement is transformed into a distinct set of edge-wise labels. Conversely, Algorithm 2 explicates the sphere fitting algorithm in the reconstruction phase, responsible for inferring node-wise displacement predictions from the edge-wise projection estimates.

Algorithm 1: Deconstruction: Label Projecting (ϕ)

Input: $\mathcal{P}^0 = \{p_1^0, p_2^0, ..., p_N^0\}$ and $\Delta\mathcal{P}^\star = \{\Delta p_i^\star = p_i^\star - p_i^0 | i \in 1, 2, ..., N\}$ - the initial position and positional displacement for the atomic system, $\mathcal{E}$ - directed edges between graph nodes.
for i in N **do**
  for j in $\mathcal{N}_i$ **do**
    $d_{ij} = (p_i^0 - p_j^0)/|p_i^0 - p_j^0|$
    $\mathbf{x}_{ij}^\star = (\Delta p_i^\star \cdot d_{ij}) d_{ij}$
  end for
end for
Return $\{\mathbf{x}_{ij}^\star \in \mathbb{R}^3 | e_{ij} \in \mathcal{E}\}$ - The set of projected vectors from nodes to edges

Algorithm 2: Reconstruction: Sphere Fitting (ϕ^{-1})

Input: $\{\hat{\mathbf{x}}_{ij} \in \mathbb{R}^3 | e_{ij} \in \mathcal{E}\}$ - prediction of positional shift projection on edges, $\mathcal{P}^0$ - initial position.
for i in N **do**
  Create $b = \mathbf{0} \in \mathbb{R}^3$, $A = \mathbf{0} \in \mathbb{R}^{3\times3}$
  for j in $|\mathcal{N}_i|$ **do**
    $b = b + (\hat{\mathbf{x}}_{ij} \cdot \hat{\mathbf{x}}_{ij})\hat{\mathbf{x}}_{ij}/|\mathcal{N}_i|$
    $A = A + \hat{\mathbf{x}}_{ij}\hat{\mathbf{x}}_{ij}^\top/|\mathcal{N}_i|$
  end for
  $C_i = (2A)^{-1}b$
  $\hat{p}_i = p_i^0 + 2C_i$
end for
Return $\hat{\mathcal{P}} = \{\hat{p}_i | i \in 1, 2, ..., N\}$ - The prediction of the positional displacement

Configurations of Graphormer and DRFormer Models

In this section, we specify the necessary details of Graphormer and DRFormer models for our experiments on OC20 dataset, with hyperparameters reported in Table 5

The projection fit block

For the 3d-Graphormer integrated with DR-Label, we retain all the hyperparameters and configurations as set in the official 3d-Graphormer implementation(Shi et al. 2022). The only difference is we are adding the projection fit block after performing all graphormer layers. Each projection fit block contains three non-linear layer operations.

Let $\mathbf{x}_{atom}$ denote the dimension of atomic embeddings, and $\mathbf{x}_{gbf}$ represent the number of Gaussian basis kernels. The non-linear transformations for sender and receiver node embeddings, respectively, are defined as

$$\mathbf{f}_{sender}(\cdot) = \mathbf{f}_{linear2}(GELU(\mathbf{f}_{linear1}(\cdot))), \quad (5)$$

$$\mathbf{f}_{receiver}(\cdot) = \mathbf{f}_{linear4}(GELU(\mathbf{f}_{linear3}(\cdot))), \quad (6)$$

where $\mathbf{f}_{linear1}, \mathbf{f}_{linear3} : \mathbb{R}^{\mathbf{x}_{atom}} \rightarrow \mathbb{R}^{\mathbf{x}_{atom}}$ and $\mathbf{f}_{linear2}, \mathbf{f}_{linear4} : \mathbb{R}^{\mathbf{x}_{atom}} \rightarrow \mathbb{R}^{\mathbf{x}_{gbf}/2}$. Here $GELU$ being the Gaussian Error Linear Units.

The resulting node embeddings concatenate with the edge embeddings in the order of: [GBF edge embedding||sender node embedding||receiver node embedding] $\in \mathbb{R}^{2\mathbf{x}_{gbf}}$. This result undergoes a further transformation via:

$$\mathbf{f}_{proj-magnitude}(\cdot) = \mathbf{f}_{linear6}(GELU(\mathbf{f}_{linear5}(\cdot))), \quad (7)$$

where $\mathbf{f}_{linear5} : \mathbb{R}^{2\mathbf{x}_{gbf}} \rightarrow \mathbb{R}^{\mathbf{x}_{gbf}}$ and $\mathbf{f}_{linear6} : \mathbb{R}^{\mathbf{x}_{gbf}} \rightarrow \mathbb{R}$. All linear layers above include bias terms.

Based on the hyperparameter setting in 3d-Graphormer+DR-Label where $\mathbf{x}_{gbf} = 128$ and $\mathbf{x}_{atom} = 768$, the additional parameter used for the projection fit block is $\sim 1.3M$, with only a $\sim 3\%$ increment of the original number of parameters for 3d-Graphormer on the OC20 task ($\sim 45M$).

The "InterPos" in Table 5 denotes whether the model incorporates explicit intermediate positional updates with the projection fit block throughout the model. The "Pos update freq" indicates the interval at which positional updates are performed within the DRFormer layers. In our case of "DRFormer-S" and "DRFormer", positional updates are conducted every 6 DRFormer layers.

Auxiliary Node Displacement Supervisions Using Noisy Nodes

Since minor perturbations to the initial atomic system configurations will yield similar equilibrium geometries, we borrow the concept of noisy nodes (Godwin et al. 2022) and implement two unique auxiliary supervision strategies in DRFormer: in-trace noise (TN) and equilibrium-state noise (EN), as depicted in Figure 6.

In-Trace Noise (TN): For atom v_i with positional displacement label $\Delta p_i^\star$, we initially execute a random interpolation between p_i^0 and $p_i^\star$, denoted as $\alpha \Delta p_i^\star = \alpha(p_i^\star - p_i^0), \alpha \in [0, 1]$. This is followed by the addition of a random 3D Gaussian noise, σ_{TN}, resulting in an instance of in-trace pseudo initial state (TN), $\widetilde{\mathcal{P}}'_{\mathcal{TN}}$:

$$\widetilde{p}_{i-TN}^0 = p_i^0 + \alpha \Delta p_i^\star + \sigma_i, \alpha \in [0, 1]. \quad (8)$$

For TN instances, the node-wise displacement supervision signal $\Delta \widetilde{p}_{i-TN}^\star$ is defined by the displacement $p_0^\star - \widetilde{p}_{i-TN}^0$.

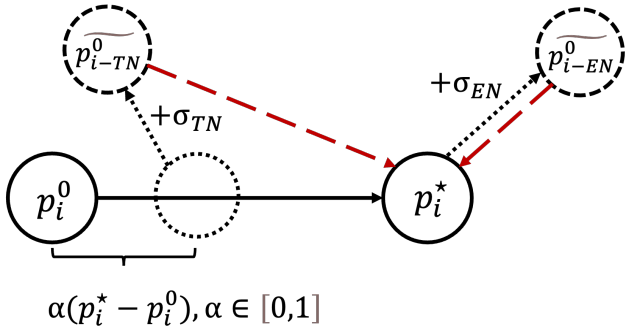

Figure 6: The in-trace (TN) noise instances and equilibrium state (EN) noise instances. Red dashed arrows signify the displacement supervision labels for corresponding instances.

Equilibrium-State Noise (EN): We introduce a random 3D Gaussian noise, σ_{EN}, directly to the equilibrium-state position, $p_i^\star$, generating an instance of pseudo initial state from the equilibrium state (EN), $\widetilde{\mathcal{P}}_{EN}^0$. For EN instances, we use $-\sigma_{EN}$ for node-wise displacement supervision.

Note that when implementing these noisy augmented supervisions, we treat the noise-augmented instances as separate instances alongside the original ones. For both "DRFormer w/o InterPos" and "DRFormer-S", only TN instances are included during training. While for "DRFormer", both TN instances and EN instances are included with the same standard deviation.

Periodic Boundary Passing Atom Calibration

Periodic boundary passing atoms (PBPA) common occur in systems with periodic boundary conditions (PBC), such as catalysis systems. We observed from experiments that during the relaxation process, numerous instances are observed where atoms traverse these periodic boundaries, which presents a challenge for displacement supervision. Figure 7 showcases nine repeating cells with a single atom v_i represented in the example. The initial and relaxed positions of the atom v_i are indicated by red and green circles, respectively, with the atom crossing the periodic boundary along the black arrow. Consequently, within the same cell, the relaxed atom reappears at the cell's opposite position. In this scenario, the original $\Delta p_i^\star = p_i^\star - p_i^0$ provided by the OC20 dataset equals to the displacement illustrated by the red arrow, which fails to accurately capture the physical relaxation process shown in the black arrow.

Given our observation that the positional displacement for atoms is invariably less than the length of the periodic boundary cell, we surmised that the actual relaxed position

Table 5: Model Configurations for Graphromer-based methods on OC20

Hyperparameter	**3d-Graphormer**	**3d-Graphormer+DR-Label**	**DRFormer w/o InterPos**	**DRFormer-S**	**DRFormer**
Optimizer	adam	adam	adam	adam	adam
Learning rate	$3e-4$	$3e-4$	$3e-4$	$3e-4$	$3e-4$
Warmup updates	10000	10000	10000	10000	10000
Batch size	64	64	64	32	64
Weight decay	0.001	0.001	0.001	0.001	0.001
Embedding dimension	768	768	960	960	768
FFN embedding dimension	768	768	960	960	768
Attention heads	48	48	48	48	48
Max update	1000000	1000000	1000000	1000000	1000000
Layers	12	12	18	12	12
Blocks	4	4	4	4	4
Node loss weight	15	15	25	25	25
GBF kernels	128	128	128	128	128
Use DR-Label	–	√	√	√	√
Edge loss weight	–	50	50	50	50
Noisy nodes TN	–	–	√	√	√
Noisy nodes EN	–	–	–	–	√
Standard deviation of Gaussian noise (Å)	–	–	0.5	0.3	0.3
InterPos	–	–	–	√	√
Pos update freq	–	–	–	6	6
PBPA-calib	–	–	√	√	√
Running time ($8 \times A100$ hour)	36	47	120	110	276
Total num of epochs	140	140	140	70	140

Table 6: Comparison of 3d-Graphormer model performance when adding the PBPA-Calib.

	MAE (eV) ↓				
Model	ID	OOD Ads.	OOD Cat.	OOD Both	Average
3d-Graphormer	0.4329	0.5850	0.4441	0.5299	0.4980
3d-Graphormer+PBPA-calib	0.4497	0.5620	0.4593	0.5088	0.4950

of boundary-passing atoms corresponds to the closest position among all equivalent relaxed atom positions in neighboring cells. Thus, we adopted the Periodic Boundary Passing Atom Calibration (PBPA-calib) algorithm as shown in 3 as a dataset cleaning operation before training.

Algorithm 3: Algorithm for Periodic Boundary Passing Atom Calibration (PBPA-calib)

Input: Initial position p_i^0, potentially boundary-passing relaxed position $p_i^{\star\prime}$, and lattice vectors on the directions of the periodic repeating plane $lattice_u, lattice_v \in \mathbb{R}^3$.
Initialize two empty lists: $Position$ and $Distance$.
for $i \in \{-1, 0, 1\}$ **do**
  for $j \in \{-1, 0, 1\}$ **do**
    Add $p_i^{\star\prime} + i \cdot lattice_u + j \cdot lattice_v$ **to** $Position$.
    Add $||p_i^{\star\prime} + i \cdot lattice_u + j \cdot lattice_v - p_i^0||$ **to** $Distance$.
  end for
end for
Compute $idx = \arg\min(Distance)$ – get the index of the element with the minimum value in $Distance$.
Set $p_i^\star = Position[idx]$.
Return $p_i^\star$ - Calibrated relaxed position of the atom .

To evaluate the efficacy of PBPA-calib, we conducted an ablation study that simply substituted the original OC20 dataset with the PBPA-calib processed dataset and trained the 3d-Graphormer. The outcome of this study, presented in Table 6, indicates that PBPA-calib aids models in supervising the actual relaxation displacement. We applied the PBPA-calib procedure to all DRFormer models, including those experiments using the SAA dataset, as well as the experiments ablation studies.

It is worth acknowledging that the contemporaneous work by Yuan et al. (Yuan et al. 2023) also addresses periodic boundary passing atoms and proposes a similar data correction algorithm. Our method was independently developed in parallel, which specifically focuses on calibrating using the two lattice vectors that form a periodically repeating infinite plane, rather than considering all three lattice vectors of the cell.

Model Configurations for DR-Label Integrated GemNet-OC and SCN

The integration of DR-Label with GemNet-OC and SCN generally follows the same algorithmic framework as its integration with Graphormer, with the majority of the model remaining unchanged. The testing set results for the direct

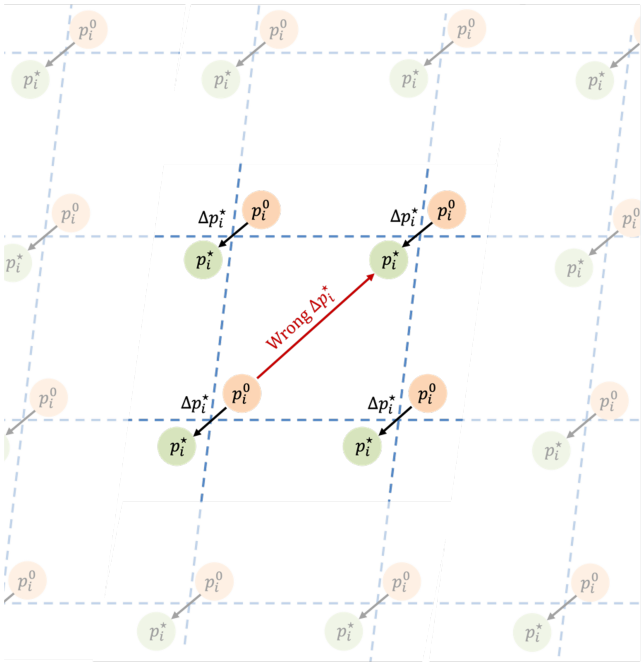

Figure 7: The periodic boundary passing node in a plane with 3x3 cells. The red arrow symbolizes the atomic positional displacement from the initial to relaxed state, as directly derived from the OC20 dataset. In contrast, the black arrow denotes the atom's actual trajectory during relaxation.

version of SCN and GemNet-OC were from Table 3 of paper(Zitnick et al. 2022).

Integration of DR-Label with GemNet-OC

The integration of the DR-Label strategy with GemNet-OC is rather straightforward. The original model employs stacks of GemNet-OC interaction blocks, each producing an embedding for every node and edge. These embeddings are then aggregated across blocks via an MLP layer to generate the final embeddings for the nodes and edges. For an edge e_{ij}, let z_{ij} represent the edge embedding, while z_i and z_j denote the embeddings of the sender and receiver nodes, respectively. We generate edge-wise magnitude predictions $\hat{\mathcal{M}}$ by applying $\hat{m}_{ij} = \mathbf{f}_{GemNet-OC}([z_{ij}||z_i||z_j])$, and use the sphere fitting algorithm to produce $\hat{\mathcal{P}}$. $\mathbf{f}_{GemNet-OC}$ is implemented by a simple linear layer with bias. GemNet-OC provides a direct force supervision head branch, which we replace with our node-level displacement supervision. Additionally, we included the edge-level projection magnitude supervision using the L1-loss function.

Integration of DR-Label with SCN

The integration of DR-Label with SCN differs slightly from the previous operation. SCN centers on a novel node representation, and generates 128 evenly distributed directions on a unit sphere as d_{ij}. We use these outward directions to perform the label decomposition and reconstruction process.

Table 7: Configurations for the GemNet-OC+DR-Label model

Hyperparameters	**Value or method**
No. spherical basis	7
No. radial basis	128
No. blocks	6
Atom embedding size	256
Edge embedding size	1024
Triplet edge embedding input size	64
Triplet edge embedding output size	128
Quadruplet edge embedding input size	64
Quadruplet edge embedding output size	32
Atom interaction embedding input size	64
Atom interaction embedding output size	64
Radial basis embedding size	32
Circular basis embedding size	16
Spherical basis embedding size	64
No. residual blocks before skip connection	2
No. residual blocks after skip connection	2
No. residual blocks after concatenation	4
No. residual blocks in atom embedding blocks	3
No. atom embedding output layers	3
Cutoff	12.0
Quadruplet cutoff	12.0
Atom edge interaction cutoff	12.0
Atom interaction cutoff	12.0
Max interaction neighbors	30
Max quadruplet interaction neighbors	8
Max atom edge interaction neighbors	20
Max atom interaction neighbors	1000
Activation	Silu
Direct forces	True
Quadruplet interaction	True
Atom edge interaction	True
Edge atom interaction	True
Atom interaction	True
Batch size	8
Learning rate initial	5.e-4
Optimizer	AdamW
Scheduler	ReduceLROnPlateau
Energy coefficient	1
EMA decay	0.999
Gradient clip norm threshold	10
Loss function - energy	mae
DR-Label integration	
Node loss weight	1
Edge loss weight	10
Loss function - edge	mae
Loss function - node	l2mae
Running time ($8 \times A100$ hour)	25

The SCN model ultimately produces a $\mathbb{R}^{(L+1)^2 \times \mathcal{C}}$ matrix for each node v_i, representing $\mathcal{C}$ channels of functions based on spherical harmonics. This representation is first transformed by $\mathbf{f}_{SH2D} : \mathbb{R}^{(L+1)^2 \times \mathcal{C}} \rightarrow \mathbb{R}^{128 \times \mathcal{C}}$ to convert the $\mathcal{C}$ channels spherical harmonics based representation of a continuous function on a sphere into a discrete representation $\mathbf{z}_{ij}$ over 128 evenly distributed directions. This is followed by $\mathbf{f}_{D2M} : \mathbb{R}^{128 \times \mathcal{C}} \rightarrow \mathbb{R}^{128 \times 1}$, transforming the $\mathbf{z}_{ij} \in \mathbb{R}^{\mathcal{C}}$ on the direction of d_{ij} into the magnitude prediction $\hat{m}_{ij}, j \in [1, 128]$. Similar to GemNet-OC, the direct force supervision head is replaced with node-level displacement and edge-level projection supervision.

Table 8: Configurations for the SCN+DR-Label model

Hyperparameters	**Value or method**
Num interactions	16
Hidden channels	1024
Sphere channels	128
Sphere channels reduce	128
Num sphere samples	128
Num basis functions	128
Distance function	gaussian
Max num neighbors	40
Cutoff	8.0
Maximum degree of the spherical harmonics	6
Use grid	True
Num bands	2
Basis width scalar	2.0
Batch size	2
Learning rate initial	0.0004
Optimizer	AdamW (amsgrad: True)
Learning rate gamma	0.3
Learning rate milestones	[218750, 281250, 343750]
Warmup steps	100
Warmup factor	0.2
Max epochs	21
Energy coefficient	2
Gradient clip norm threshold	100
EMA decay	0.999
Loss function - energy	mae
DR-Label integration	
Node loss weight	1
Edge loss weight	10
Loss function - edge	mae
Loss function - node	l2mae
Running time ($8 \times A100$ hour)	161

Detailed Configurations for Other Experiments

We provide additional details regarding other experiments conducted in Table 9 and 10, including experiments on the SAA datasets and ablation studies. For the SAA dataset experiments, we maintained the original structure of 3d-Graphormer and DRFormer-S, with adjustments made to the warmup updates and maximum update parameters to align with the reduced dataset size. In the ablation studies, we also modified the warmup updates and maximum update parameters to ensure a similar total number of epochs (around 140) compared to the experiments on the OC20-all dataset.

Table 9: Model Configurations for SAA experiments

Hyperparameter	**3d-Graphormer**	**DRFormer-S**	**DRFormer-S-FT**
Optimizer	adam	adam	adam
Learning rate	$3e-4$	$3e-4$	$3e-4$
Warmup updates	200	200	200
Batch size	2	2	2
Weight decay	0.001	0.001	0.001
Embedding dimension	768	960	960
FFN embedding dimension	768	960	960
Attention heads	48	48	48
Max update	64000	64000	64000
Layers	12	12	12
Blocks	4	4	4
Node loss weight	15	25	25
GBF kernels	128	128	128
Use DR-Label	–	√	√
Edge loss weight	–	50	50
Noisy nodes TN	–	√	√
Noisy nodes EN	–	–	–
Standard deviation of Gaussian noise (Å)	–	0.3	0.3
InterPos	–	√	√
Pos update freq	–	6	6
PBPA-calib	√	√	√
Running time ($1 \times V100$ hour)	2	4	4
Total num of epochs	70	70	70

Table 10: Model Configurations for Ablation Studies (OC20-10k)

Module Name	**Model1**	**Model2**	**Model3**	**Model4**	**Model5**
DR-Label	×	√	×	√	√
Noisy Nodes	×	×	√	√	√
Intermediate Pos	×	×	×	×	√
Hyperparameter					
Optimizer	adam	adam	adam	adam	adam
Learning rate	$3e-4$	$3e-4$	$3e-4$	$3e-4$	$3e-4$
Warmup updates	800	800	800	800	800
Batch size	64	64	64	64	64
Weight decay	0.001	0.001	0.001	0.001	0.001
Embedding dimension	768	768	768	768	768
FFN embedding dimension	768	768	768	768	768
Attention heads	48	48	48	48	48
Max update	21800	21800	21800	21800	21800
Layers	12	12	12	12	12
Blocks	4	4	4	4	4
Node loss weight	15	15	15	15	15
GBF kernels	128	128	128	128	128
Use DR-Label	–	√	–	√	√
Edge loss weight	–	50	–	50	50
Noisy nodes TN	–	–	√	√	√
Noisy nodes EN	–	–	–	–	–
Standard deviation of Gaussian noise (Å)	–	–	0.3	0.3	0.3
InterPos	–	–	–	–	√
Pos update freq	–	–	–	–	6
PBPA-calib	√	√	√	√	√
Running time ($8 \times A100$ hour)	1.5	1.5	1.8	1.9	5.1
Total num of epochs	140	140	140	140	140