REVIEW ARTICLE

A survey of geometric graph neural networks: data structures, models and applications

Jiaqi HAN[1,2*], **Jiacheng CEN**[1*], **Liming WU**[1*], **Zongzhao LI**[1], **Xiangzhe KONG**[3], **Rui JIAO**[3],
Ziyang YU[3], **Tingyang XU**[4,5], **Fandi WU**[6], **Zihe WANG**[1], **Hongteng XU**[1], **Zhewei WEI**[1],
Deli ZHAO[4,5], **Yang LIU** (✉)[3], **Yu RONG** (✉)[4,5], **Wenbing HUANG** (✉)[1]

1 Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China
2 Department of Computer Science, Stanford University, CA 94305, USA
3 Department of Computer Science and Technology, Institute for AI, Tsinghua University, Beijing 100084, China
4 DAMO Academy, Alibaba Group, Hangzhou 311121, China
5 Hupan Lab, Hangzhou 311121, China
6 Tencent AI Lab, Shenzhen 518100, China

Abstract Geometric graphs are a special kind of graph with geometric features, which are vital to model many scientific problems. Unlike generic graphs, geometric graphs often exhibit physical symmetries of translations, rotations, and reflections, making them ineffectively processed by current Graph Neural Networks (GNNs). To address this issue, researchers proposed a variety of geometric GNNs equipped with invariant/equivariant properties to better characterize the geometry and topology of geometric graphs. Given the current progress in this field, it is imperative to conduct a comprehensive survey of data structures, models, and applications related to geometric GNNs. In this paper, based on the necessary but concise mathematical preliminaries, we formalize geometric graph as the data structure, on top of which we provide a unified view of existing models from the geometric message passing perspective. Additionally, we summarize the applications as well as the related datasets to facilitate later research for methodology development and experimental evaluation. We also discuss the challenges and future potential directions of geometric GNNs at the end of this survey.

1 Introduction

Many scientific problems particularly in physics and biochemistry require to process data in the form of *geometric graphs* [1]. Distinct from typical graph data, geometric graphs

* These authors contributed equally to this work. Work done by Jiaqi Han during his visit to Renmin University of China.

additionally assign each node a special type of node feature in the form of geometric vectors. For example, a molecule/protein can be regarded as a geometric graph, where the 3D position coordinates of atoms are the geometric vectors; in a general multi-body physical system, the 3D states (positions, velocities or spins) are the geometric vectors of the particles. Notably, geometric graphs exhibit symmetries of translations, rotations and/or reflections. This is because the physical law controlling the dynamics of the atoms (or particles) is the same no matter how we translate or rotate the physical system from one place to another. When tackling this type of data, it is essential to incorporate the inductive bias of symmetry into the design of the model, which motivates the study of geometric Graph Neural Networks (GNNs).

Constructing GNNs that permit such symmetry constraints has long been challenging to methodological design. Pioneer approaches like DTNN [2], DimeNet [3], and GemNet [4], transform the input geometric graph into distance/angle/dihedral-based scalars that are invariant to rotations or translations, constituting the family of invariant GNNs. Noticing the limit on the expressivity of invariant GNNs, EGNN [5] and PaiNN [6] additionally involve geometric vectors in message passing and node update to preserve the directional information in each layer, leading to equivariant GNNs. With group representation theory as a helpful tool, TFN [7], SE(3)-Transformer [8], and SEGNN [9] generalizes invariant scalars and equivariant vectors by viewing them as steerable vectors parameterized by high-degree spherical tensors, giving rise to high-degree steerable GNNs. Built upon these fundamental approaches, geometric GNNs have made remarkable success in various applications of diverse systems, including physical dynamics simulation [10,11], molecular property prediction [5,8], protein structure prediction [12], protein generation [13,14], and RNA structure ranking [15]. Figure 1 illustrates the superior performance of geometric
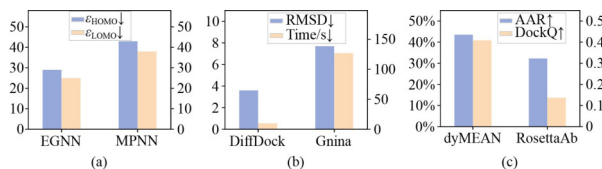
Fig. 1 Performance comparisons between geometric GNNs and traditional methods on molecular property prediction, protein-ligand docking, and antibody design. Notably, the methods based on geometric GNNs, including EGNN [5], DiffDock [16], and dyMEAN [17], remarkably outperform traditional MPNN [18], Gnina [19], and RossetaAb [20], on the datasets of QM9 [21], PDBBind [22], and SAbDab [23], respectively, verifying the effectiveness and efficiency of geometric GNNs over various tasks. (a) Property prediction; (b) ligand docking; (c) antibody design

GNNs against traditional methods on the representative tasks.

To facilitate the research of geometric GNNs, this work presents a systematic survey focusing both on the methods and applications[1], which is structured as the following sections: In Section 2, we introduce necessary preliminaries on group theory and the formal definition of equivariance/invariance; In Section 3, we propose geometric graph as a universal data structure that will be leveraged throughout the entire survey as a bridge between real-world data and the models, i.e., geometric GNNs; In Section 4, we summarize existing models into invariant GNNs (Section 4.2) and equivariant GNNs (Section 4.3), while the latter is further categorized into scalarization-based models (Section 4.3.1) and high-degree steerable models (Section 4.3.2); Besides, we also introduce geometric graph transformers in Section 4.4; In Section 5, we provide a comprehensive collection of the applications that have witnessed the success of geometric GNNs on particle-based physical systems, molecules, proteins, complexes, and other domains like crystals and RNAs.

The goal of this survey is to provide a general overview throughout data structure, model design, and applications (see Fig. 2), which constitutes an entire input-output pipeline that is instructive for machine learning practitioners to employ geometric GNNs on various scientific tasks. Recently, several related surveys have been proposed, which place main focus on methodology of geometric GNNs [36], pretrained GNNs for chemical data [37], representation learning for molecules [38,39], and general application of artificial intelligence in diverse types of scientific systems [40]. In contrast to all of them, this survey places an emphasis on geometric graph neural networks, not only encapsulating theoretical foundations of geometric GNNs but also delivering an exhaustive summary of the related applications in domains across physics, biochemistry, and material science. Meanwhile, we discuss future prospects and interesting research directions in Section 6. We also release the Github repository that collects the reference, datasets, codes, benchmarks, and other resources related to geometric GNNs.

2 Basic notion of symmetry

In this section, we will compactly introduce the basic notions related to symmetry. Readers can skip this section and get straight to the methodology part in Section 3 if they are familiar with the theoretical background.

2.1 Transformation and group

By defining symmetry, we indicate that an object of interest keeps invariant under a set of transformations. For instance, the distance between any two points in space remains constant, regardless of how we simultaneously rotate or translate these two points. Mathematically, a set of transformations forms a *group* (more details are referred to [41]).

Definition 1 (Group). A group G is a set of transformations with a binary operation "$\cdot$" satisfying these properties: (i) it is closed, namely, $\forall a, b \in G, a \cdot b \in G$; (ii) it is associative, namely, $\forall a, b, c \in G, (a \cdot b) \cdot c = a \cdot (b \cdot c)$; (iii) there exists an identity element $e \in G$ such that $\forall a \in G, a \cdot e = e \cdot a = a$; (iv) each element must have an inverse, namely, $\forall a \in G, \exists b \in G, a \cdot b = b \cdot a = e$, where the inverse b is denoted as a^{-1}.

We below provide some examples commonly used in the applications of this paper:

- E(d) is an Euclidean group [42] consisting of rotations, reflections and translations, acting on d-dimension vectors.
- T(d) is a subgroup of Euclidean group that consists of translations.
- O(d) is an orthogonal group that consists of rotations and reflections, acting on d-dimension vectors.
- SO(d) is a special orthogonal group that only consists of rotations.
- SE(d) is a special Euclidean group that consists of only rotations and translations.
- Lie Group is a group whose elements form a differentiable manifold. Actually, all the groups above are specific examples of Lie Group.
- S$_N$ is a permutation group whose elements are permutations of a given set consisting of N elements.

2.2 Group representation

While the group operation "$\cdot$" is defined abstractly above, it can be realized as matrix multiplication, with the help of group representation. A representation of G is a group homomorphism $\rho(g) : G \mapsto \text{GL}(\mathcal{V})$ that takes as input the group element $g \in G$ and acts on the general linear group of some vector space $\mathcal{V}$, satisfying $\rho(g)\rho(h) = \rho(g \cdot h), \forall g, h \in G$. When $\mathcal{V} = \mathbb{R}^d$, then $\text{GL}(\mathcal{V})$ contains all invertible $d \times d$ matrices and $\rho(g)$ assigns a matrix to the element g.

For the orthogonal group O(d), one of its common group representations is defined by orthogonal matrices $\boldsymbol{O} \in \mathbb{R}^{d \times d}$ subject to $\boldsymbol{O}^\top \boldsymbol{O} = \boldsymbol{I}$; for SO($d$), the group representation is restricted to orthogonal matrices of determinant 1, denoted as $\boldsymbol{R}$. The case of translation group T(d) is a bit tedious and can be derived in the projective space using homogeneous coordinates; here, for simplicity, we directly define translation as vector addition other than matrix multiplication. Note that

---

[1] This work is an extended survey of our previous short version [24].
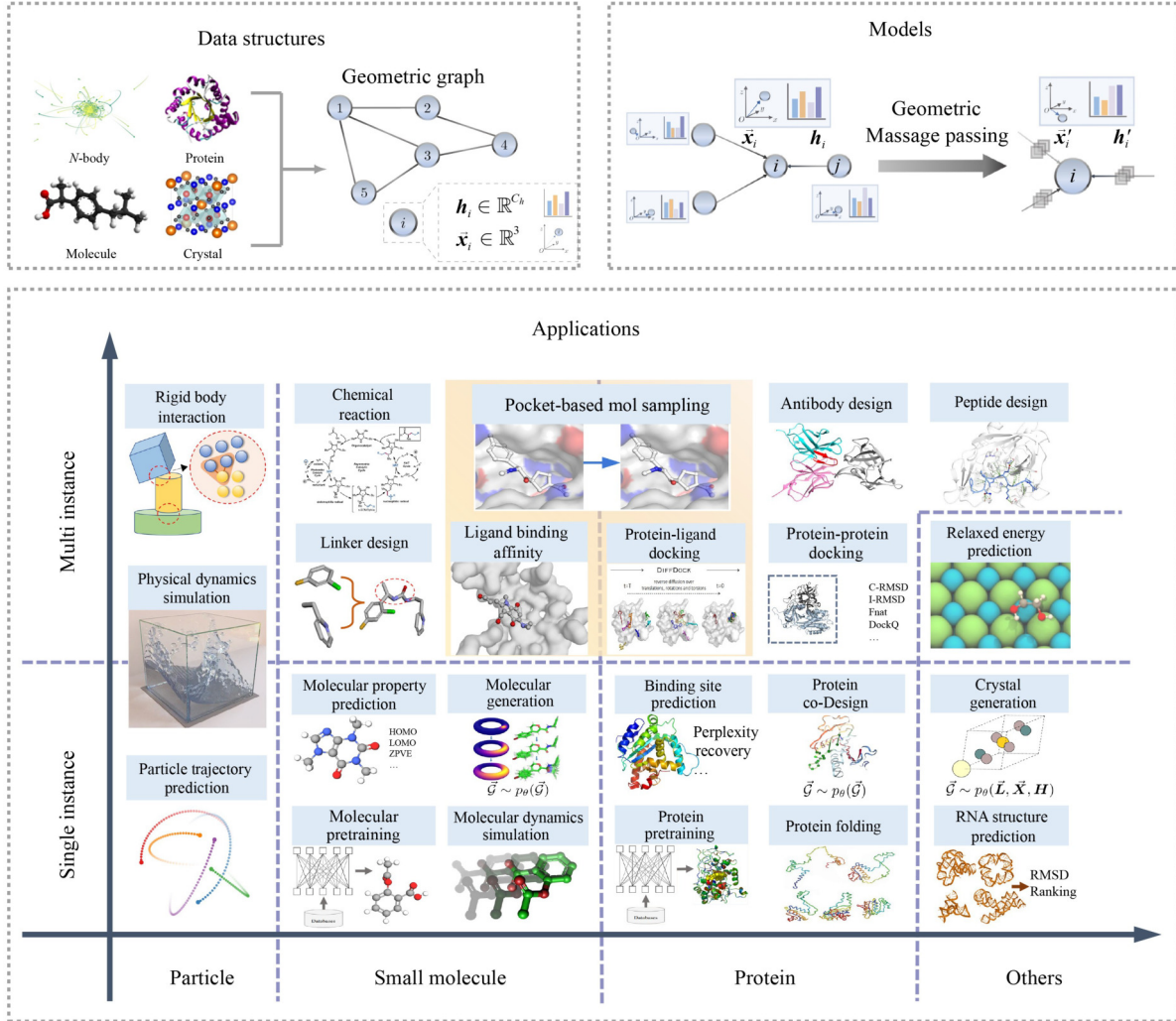
Fig. 2 Illustration of the complete input-output pipeline from data structures, models to applications. Note that most figures here for illustrating different applications are edited based on previous papers [16,25–35]. The term "instance" indicates a self-interacted system composed of multiple particles/atoms, such as a molecule or a protein. Pocket-Based Molecule Sampling, Ligand-Binding Affinity Prediction, and Protein-Ligand Docking are denoted with yellow shading to imply that all these tasks take as the multi-instance format "Molecule+Protein"

the representation of a group is not unique, which will be further illustrated in Section 4.3.2.

2.3 Equivariance and invariance

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output vector spaces, respectively. The function $\phi : \mathcal{X} \rightarrow \mathcal{Y}$ is called equivariant with respect to G if when we apply any transformation to the input, the output also changes via the same transformation or under a certain predictable behavior. In form, we have

Definition 2 (Equivariance). The function $\phi : \mathcal{X} \mapsto \mathcal{Y}$ is G-equivariant if it commutes with any transformation in G,

$$\phi(g \cdot x) = g \cdot \phi(x), \forall g \in G, \qquad (1)$$

which, by implementing the group operation $\cdot$ with group representation, can be rewritten as:

$$\phi(\rho_{\mathcal{X}}(g)x) = \rho_{\mathcal{Y}}(g)\phi(x), \forall g \in G, \qquad (2)$$

where $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$ are the group representations in the input and output space, respectively.

The choice of group representation facilitates the specialization of different scenarios. When both $\rho_{\mathcal{X}}$ and $\rho_{\mathcal{Y}}$ are trivial representations, namely, $\rho_{\mathcal{X}}(g) = \rho_{\mathcal{Y}}(g) = \boldsymbol{I}$ [2], ϕ becomes a trivial function; notably, when $\rho_{\mathcal{Y}}(g) = \boldsymbol{I}$, ϕ is called *an invariant function*, demonstrating that invariance is just a special case of equivariance.

It is able to verify that equivariance induces the following desirable properties. (i) **Linearity:** any linear combination of equivariant functions is still equivariant. (ii) **Composability:** the composition of two equivariant functions (if they can be composed) yields an equivariant function. Therefore, equivariance for each layer of a network implies that a whole network is equivariant. (iii) **Inheritability:** if a function is equivariant with respect to group G_1 and group G_2, then this function must be equivariant with respect to the direct product of these two groups, i.e., $G_1 \times G_2$ under a corresponding definition of product group operation or group representation.

---

[2] Note that the identity transformation $\boldsymbol{I}$ could have different dimensions in the input space $\mathcal{X}$ and output space $\mathcal{Y}$.

This implies that proving equivariance of each transformation individually is sufficient to prove equivariance of joint transformations.

In the following context, the variable x is instantiated as a geometric graph, the group transformation $\rho(g)$ becomes the transformation of geometric graphs, and the function ϕ is designed as an invariant/equivariant GNN.

3 Data structure: from graph to geometric graph

This section formally defines graph and geometric graph, and depicts how they differ from each other. Table 1 summarizes the notations we used throughout this paper.

3.1 Graph

Conventional studies on graphs [43,44] usually focus on their relational topology. Examples include social networks, citation networks, etc. In the domain of AI-Driven Drug Design (AIDD), they are usually referred to as 2D graphs [45].

Definition 3 (Graph). A graph is defined as $\mathcal{G}:=(A, H)$, where $A \in [0,1]^{N \times N}$ is the adjacency matrix with N being the number of nodes, and $H \in \mathbb{R}^{N \times C_h}$ is the node feature matrix with C_h being the dimension of the feature.

Concretely, the adjacency matrix A takes the value 1 at its (i, j)-entry a_{ij} when node i and j are connected by an edge and 0 otherwise. The ith row of H, i.e., $h_i \in \mathbb{R}^{C_h}$, represents the feature vector for node i, e.g., the one-hot embedding of the atomic number in a molecule graph. Along with the

definition of graph, we also describe some vital concepts derived. We denote the set of nodes as $\mathcal{V}$ and the set of edges as $\mathcal{E}$. Correspondingly, the neighborhood of node i, marked as $\mathcal{N}_i$, is specified to be $\mathcal{N}_i := \{j : (v_i, v_j) \in \mathcal{E}\}$. The graph can additionally contain some edge features $e_{ij} \in \mathbb{R}^{C_e}$ for edge (v_i, v_j).

Transformations on graphs: $g \cdot \mathcal{G}$. One can arbitrarily change the order of nodes without changing the topology of the graph. With the language of group representation, the permutation transformation of a graph is denoted as $g \cdot \mathcal{G} := (P_g A P_g^\top, P_g H)$, where P_g is the representation of the transformation $g \in S_N$ (i.e., the permutation matrix[3]). We denote the equivalence in terms of permutation as $\mathcal{G} \simeq g \cdot \mathcal{G}$.

As a concrete example, molecules can be viewed as graphs, where the nodes v_i are instantiated as the atoms, and the node features H are the one-hot encoding of the atomic numbers, a row for each atom. The edges A are either the existence of chemical bonds or constructed based on relative distance between atoms under a cut-off threshold, and the respective edge features e_{ij} can be assigned as the type of the chemical bond and/or the relative distance.

3.2 Geometric graph

In many applications, the graphs we tackle contain not only the topological connections and node features, but also certain *geometric information*. Again, in the example of a molecule, we may additionally be informed of some geometric quantities in the Euclidean space, e.g., the positions of the atoms in 3D

Table 1 Basic notations and definitions throughout this survey

Notation	Description
	Data structure
$\mathcal{G}:=(A, H)$	A graph $\mathcal{G}$ containing N nodes, with adjacency matrix $A \in \mathbb{R}^{N \times N}$ and node feature matrix $H \in \mathbb{R}^{N \times C_h}$.
$\vec{\mathcal{G}}:=(A, H, \vec{X})$	A geometric graph $\vec{\mathcal{G}}$ containing N nodes, with adjacency matrix A and node feature matrix H as above, and additionally a 3D coordinate matrix $\vec{X} \in \mathbb{R}^{N \times 3}$.
$\mathcal{N}_i$	The neighborhood of node i.
$h_i \in \mathbb{R}^{C_h}$	The scalar feature of node i.
$\vec{x}_i \in \mathbb{R}^3$	The 3D coordinate of node i.
$\vec{V}_i \in \mathbb{R}^{3 \times C}$	The multi-channel 3D vector of node i.
$\vec{V}_i^{(l)} \in \mathbb{R}^{(2l+1) \times C_l}$	The type-l irreducible vector of node i.
$\vec{\mathbb{V}}_i^{(\mathbb{L})} := \{\vec{V}_i^{(l)}\}_{l \in \mathbb{L}}$	The set consisting of irreducible vectors of all types $l \in \mathbb{L}$.
$e_{ij} \in \mathbb{R}^{C_e}$	The edge feature from node j to i.
	Operator
G, g	The group G and its group element g.
$\rho_X(g)$	The group representation $\rho_X(g)$ of the transformation g in the vector space X.
$\times, \otimes$	The operators between two vectors including cross product $\times$ and Kronecker product $\otimes$
$\otimes_{\text{cg}}, \otimes_{\text{cg}}^W, \otimes_{\text{cg}}^{\mathbb{W}}$	Clebsch-Gordan (CG) tensor product, optionally with a learnable parameter W and a learnable parameter set $\mathbb{W}$.
$Y^{(l)}(\vec{x}) \in \mathbb{R}^{2l+1}$	The type-l vector constructed by spherical harmonics of $\vec{x} \in S^2$: $Y^{(l)}(\vec{x}) = [Y_{-l}^{(l)}, Y_{-l+1}^{(l)}, \cdots, Y_{l-1}^{(l)}, Y_l^{(l)}]$.
$\mathbb{Y}^{(\mathbb{L})}(\vec{x}) := \{Y^{(l)}(\vec{x})\}_{l \in \mathbb{L}}$	A set consisting of spherical harmonics of all types $l \in \mathbb{L}$.
$D^{(l)}(g)$	The l-th degree Wigner-D matrix of the rotation transformation $g \in SO(3)$.
	Neural Network
$\phi, \psi, \varphi, \sigma$	Functions implemented with MLP.

---

[3] The permutation of A can also be written in the form of group representation by first vectorizing A as $\text{Vec}(A)$ and then conducting $(P_g \otimes P_g)\text{Vec}(A)$. Here $\otimes$ defines the Kronecker product, and $P_g \otimes P_g$ is the 2-order representation of the permutation matrix.

coordinates[4]. Such quantities are of particular interest in that they encapsulate rich directional information that depicts the geometry of the system. With the geometric information, one can go beyond working on limited perception of the graph topology, but instead to a broader picture of the entire configuration of the system in 3D space, where important information, such as the relative orientation of the neighboring nodes and directional quantities like velocities, could be better exploited. Hence, in this section, we begin with the definition of geometric graphs, which are usually referred to as 3D graphs [1].
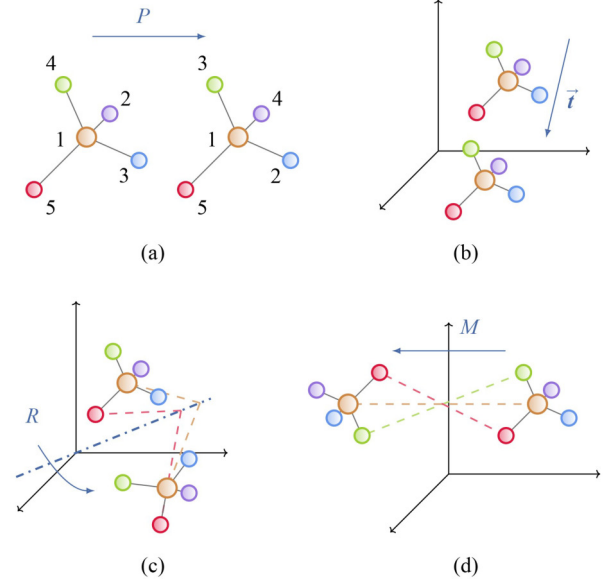
Definition 4 (Geometric Graph). A geometric graph is defined as $\vec{\mathcal{G}}:=(A, H, \vec{X})$, where $A \in [0,1]^{N \times N}$ is the adjacency matrix, $H \in \mathbb{R}^{N \times C_h}$ is the node feature matrix with dimension C_h, and $\vec{X} \in \mathbb{R}^{N \times 3}$ are the 3D coordinates of all nodes.

The ith rows of H and $\vec{X}$, namely, $h_i \in \mathbb{R}^{C_h}$ and $x_i \in \mathbb{R}^3$ denote the feature and 3D coordinate of node v_i, respectively. In the above definition, we distinguish the coordinate matrix $\vec{X}$ from other quantities A and H, and geometric graph $\vec{\mathcal{G}}$ from graph $\mathcal{G}$, with an over-right arrow "$\rightarrow$", indicating that they contain geometric and directional information. Note that there could be other geometric variables besides $\vec{X}$ in a geometric graph, such as velocity, force, and so on. Then the shape of $\vec{X}$ is extended from $N \times 3$ to $N \times 3 \times C$ where C denotes the number of channels. In this section, we assume $C = 1$ for conciseness, while more complete examples are shown in Section 5.

Transformations on geometric graphs: $g \cdot \vec{\mathcal{G}}$. In contrast to graphs, transformations on geometric graphs are not limited to node permutation. We summarize the transformations of interest below:

- **Permutation**, which is defined as $g \cdot \vec{\mathcal{G}}:=(P_g A P_g^\top, P_g H, P_g \vec{X})$, where P_g is the permutation matrix representation of $g \in S_n$;
- **Orthogonal transformation**, which is defined as $g \cdot \vec{\mathcal{G}}:=(A, H, \vec{X} O_g)$, where O_g is the orthogonal matrix representation of $g \in O(3)$, consisting of rotations and reflections;
- **Translation**, which is defined as $g \cdot \vec{\mathcal{G}}:=(A, H, \vec{X} + \vec{t}_g)$, where $\vec{t}_g$ is the translation vector of $g \in T(3)$.

We always have the equivalence $\vec{\mathcal{G}} \simeq g \cdot \vec{\mathcal{G}}$. We can combine orthogonal transformation and translation into Euclidean transformation on geometric graphs, namely, $g \cdot \vec{\mathcal{G}}:= (A, H, \vec{X} O_g + \vec{t}_g)$ for $g \in E(3)$. Here, the Euclidean group E(3) is a semidirect product [46] of orthogonal transformation and translation, denoted as $E(3) = T(3) \ltimes O(3)$. We can similarly define SE(3) transformation by considering only rotation and translation. We sometimes call H invariant features (or scalars), since they are independent to E(3) transformation, and call $\vec{X}$ equivariant features (or vectors) that correlate to E(3) transformations. Figure 3 demonstrates the example of transformation on geometric graph.

Fig. 3 Examples of transformations on geometric graphs. (a) Permutation; (b) translation; (c) rotation; (d) reflection

Geometric graphs are powerful and general tools to model a variety of objects in scientific tasks, including small molecules [5,47], proteins [14,48], crystals [49,50], physical point clouds [25,51], and many others.

We will provide more details in Section 5.

4   Model: geometric GNNs

In this section, we first recap the general form of Message Passing Neural Network (MPNN) on topological graphs. Then we introduce different types of geometric GNNs that extends the message passing paragidm of MPNNs to geometric graphs: invariant GNNs, equivariant GNNs, as well as geometric graph transformers. Finally, we briefly present the works that discuss the expressivity of geometric GNNs. Figure 4 presents the taxonomy of geometric GNNs in this section.

4.1   Message passing neural networks

Graph Neural Networks (GNNs) are favorable to operate on graphs with the help of the message-passing mechanism, which facilitates the information propagation along the graph structure by updating node embeddings through neighborhood aggregation. To be specific, message-passing GNNs implement $\phi(\mathcal{G})$ on topological graphs $\mathcal{G}$ by iterating the following message-passing process in each layer [18],

$$m_{ij} = \phi_{\mathrm{msg}}\left(h_i, h_j, e_{ij}\right), \tag{3}$$

$$h_i' = \phi_{\mathrm{upd}}\left(h_i, \{m_{ij}\}_{j \in \mathcal{N}_i}\right), \tag{4}$$

where $\phi_{\mathrm{msg}}(\cdot)$ and $\phi_{\mathrm{upd}}(\cdot)$ are the message computation and feature update function, respectively. The node features h_i, h_j and edge feature e_{ij} is first synthesized by the message function to obtain the message m_{ij}. The messages within the neighborhood are then aggregated with one set function and

---

[4] Although we mainly focus on 3D space, most of our analyses can be extended to d-dimensional space where d is an arbitrary integer.
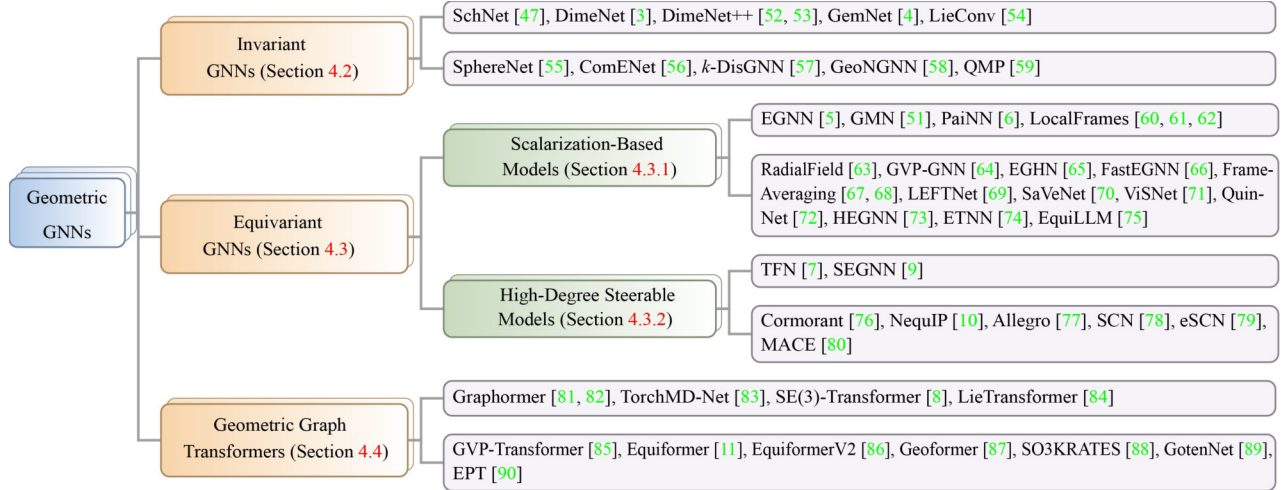
Fig. 4   Taxonomy of geometric GNNs introduced in Section 4

leveraged to update the node features $\boldsymbol{h}'_i$ combined with the input $\boldsymbol{h}_i$.

GNNs defined by Eqs. (3) and (4) are always permutation equivariant but not inherently E(3)-equivariant. When mentioning equivariance or invariance in what follows, this paper mainly discusses the latter unless otherwise specified.

4.2 Invariant graph neural networks

Moving forward to the geometric domain, there are various tasks that require the model we propose to be *invariant* with regard to Euclidean transformations. For instance, for the task of molecular property prediction, the predicted energy should remain unchanged regardless of any rotation/translation of all atom coordinates. Embedding such inductive bias is crucial as it essentially conforms to the physical rule of our 3D world.

In form, invariant GNNs update invariant features as $H' = \phi(\vec{\mathcal{G}})$ with the function ϕ satisfying:

$$\phi(g \cdot \vec{\mathcal{G}}) = \phi(\vec{\mathcal{G}}), \forall g \in E(3). \qquad (5)$$

To design such function, invariant GNNs usually transform equivariant coordinates $\vec{X}$ to invariant scalars that are unaffected by Euclidean transformations. Early invariant GNNs can date back to DTNN [2], MPNN [18], and MV-GNN [91], where relative distances are applied for edge construction. Recent works further elaborate the use of various invariant scalars ranging from relative distances to angles or dihedral angles between edges, upon the message passing mechanism in Eqs. (3) and (4). We introduce several representative works below.

SchNet [47]. This work designs a continues filter convolution conditional on relative distances $r_{ij} = \|\vec{x}_i - \vec{x}_j\|$. In particular, it re-implements Eq. (3) as

$$\boldsymbol{m}_{ij} = \sigma_2(r_{ij})\sigma_1(\boldsymbol{h}_j), \qquad (6)$$

where the message is calculated as the multiplication between the continues convolution filter and the neighbor embedding, and the functions σ are all Multi-Layer Perceptrons (MLPs).

DimeNet [3]. By observing that using relative distances alone is unable to encode directional information, DimeNet proposes directional message passing which takes as input not only relative distances but also angles between adjacent edges. The main component to compute the message embedding of each directional edge (from j to i) is given by:

$$\boldsymbol{m}'_{ji} = \sigma_{\text{msg}}\left(\boldsymbol{m}_{ji}, \sum_{k \in \mathcal{N}_j \setminus \{i\}} \sigma_{\text{int}}(\boldsymbol{m}_{kj}, \boldsymbol{e}^{(ji)}_{\text{RBF}}, \boldsymbol{e}^{(kji)}_{\text{CBF}}) \right), \qquad (7)$$

where $\boldsymbol{e}^{(ji)}_{\text{RBF}}$ denotes the radial basis function representation of relative distance d_{ji}; $\boldsymbol{e}^{kji}_{\text{CBF}}$ [5] computes the joint representation of relative distance d_{kj} and angle $\alpha_{(kj,ji)}$ between edge (v_k, v_j) and (v_j, v_i), with the help of spherical Bessel functions and spherical harmonics. In [3], Eq. (7) is applied as an interaction block before an embedding block that derives the message $\boldsymbol{m}_{ji}$ based on $\boldsymbol{e}^{(ji)}_{\text{RBF}}$ and hidden features $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$. The updated messages $\boldsymbol{m}'_{ji}$ of all neighbor nodes are then utilized to update hidden feature $\boldsymbol{h}_i$. A faster version of DimeNet is proposed later, dubbed DimeNet++ [52,53].

GemNet [4]. To achieve universal expressivity, GemNet further takes dihedral angles into account, formulating two-hop directional message passing based on quadruplets of nodes. Basically, it replaces the message embeddings from Eq. (7) in DimeNet [3] with the following form:

$$\boldsymbol{m}'_{ji} = \sigma_{\text{msg}}\left(\boldsymbol{m}_{ji}, \sum_{\substack{k \in \mathcal{N}_i \setminus \{j\} \\ l \in \mathcal{N}_k \setminus \{i, j\}}} \boldsymbol{m}_{jikl} \right), \qquad (8)$$

$$\boldsymbol{m}_{jikl} = \sigma_{\text{int}}\left(\boldsymbol{m}_{lk}, \boldsymbol{e}^{(lk)}_{\text{RBF}}, \boldsymbol{e}^{(ikl)}_{\text{CBF}}, \boldsymbol{e}^{(jikl)}_{\text{SBF}} \right), \qquad (9)$$

where $\boldsymbol{e}^{(lk)}_{\text{RBF}}$ and $\boldsymbol{e}^{(ikl)}_{\text{CBF}}$ are defined as above; $\boldsymbol{e}^{(jikl)}_{\text{SBF}}$ [6] are calculated by, the spherical Bessel function of relative distance d_{ji}, and spherical harmonics of angle $\alpha_{ji,ik}$ and dihedral angle $\alpha_{ji,kl}$. The input of Eq. (8) additionally integrates hidden features $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ for more expressivity in its original formulation. Note that GemNet can be modified to enable equivariant output by multiplying the output with the

---

[5] Here CBF is short for Circular Bessel Function.
[6] Here SBF is short for Spherical Bessel Function.

associated direction, which belongs to scalarization based equivariant GNNs introduced in the next subsection.

LieConv [54]. LieConv is formulated as follows.

$$m_{ij} = \sigma\left(\log(u_i^{-1}u_j)\right)h_j, \tag{10}$$

$$h_i' = \frac{1}{|\mathcal{N}(i)|+1}\left(h_i + \sum_{j\in\mathcal{N}(i)}m_{ij}\right), \tag{11}$$

where $u_i \in G$ is a lift of $\vec{x}_i$, the logarithm log maps each group member onto the Lie Algebra $\mathfrak{g}$ that is a vector space, and σ is a parametric MLP. Besides, Eq. (11) conducts normalization by the division of the number of all nodes, i.e., $|\mathcal{N}(i)|+1$. It is clear that LieConv only specifies the update of node features h_i while keeping the geometric vectors $\vec{x}_i$ unchanged. That means LieConv is invariant.

In addition to the above models, SphereNet [55] is another prevailing invariant GNN. Similar to GemNet, SphereNet also exploits relative distances, angles, and torsion angles for geometric modeling, which is able to distinguish almost all 3D graph structures. Moreover, its proposed spherical message passing (SMP) enables both fast and accurate 3D molecular learning on large-scale molecules. ComENet [56] is another type of invariant model which incorporates 3D information completely and efficiently. It ensures global completeness of model only with message passing in 1-hop neighborhood to avoid time-consuming calculations like torsion in SphereNet or dihedral angles in GemNet. k-DisGNN [57] relies solely on invariant relative distance information, yet adopts high-order message-passing frameworks from traditional graph learning (e.g., k-WL or k-FWL), achieving completeness for $k=2$. GeoNGNN [58], the geometric extension of the simplest subgraph GNN (NGNN [92]), effectively utilizes local subgraph information and also attains completeness with only distance features. There are also some other studies [59,93–95] exploiting the quaternion algebra to represent the 3D rotation group, which mathematically ensures SO(3) invariance during the inference. Specifically, QMP [59] constructs quaternion message-passing module to distinguish the molecular conformations caused by bond torsions.

4.3   Equivariant graph neural networks

In contrast to invariant GNNs that only conduct the update of invariant features, equivariant GNNs simultaneously update both invariant features and equivariant features, given that many practical tasks (such as molecular dynamics simulation) requires equivariant output. More importantly, as proved in [96], equivariant GNNs are strictly more expressive than invariant GNNs particularly for sparse geometric graphs.

In form, equivariant GNNs design the function over geometric graphs as $(\boldsymbol{H}', \vec{\boldsymbol{X}}') = \phi(\vec{\mathcal{G}})$ satisfying:

$$\phi(g \cdot \vec{\mathcal{G}}) = g \cdot \phi(\vec{\mathcal{G}}), \forall g \in \mathrm{E}(3). \tag{12}$$

Specifically, through the lens of message-passing in Eqs. (3) and (4), the geometric message is derived as

$$m_{ij}, \vec{m}_{ij} = \phi_{\mathrm{msg}}\left(h_i, h_j, \vec{x}_i, \vec{x}_j, e_{ij}\right). \tag{13}$$

In subsequent, the computed geometric messages $\vec{m}_{ij}$ are aggregated within the neighborhood $\mathcal{N}_i$ specified by the connectivity or adjacency matrix of the graph, and updated by taking the input features into account. This update process is formally summarized as

$$h_i', \vec{x}_i' = \phi_{\mathrm{upd}}\left(h_i, \{(m_{ij}, \vec{m}_{ij})\}_{j\in\mathcal{N}_i}\right). \tag{14}$$

The functions ϕ_{msg} and ϕ_{upd} should ensure that all invariant/equivariant output to be invariant/equivariant with respect to any E(3) transformation of the input.

There are different ways to realize the specific form of ϕ_{msg} and ϕ_{upd}. Below, we categorize current famous equivariant GNNs into two classes: scalarization-based models and high-degree steerable models.

4.3.1   Scalarization-based models

This line of works first translates 3D coordinates into invariant scalars, which is similar to the design of invariant GNNs, but it refines beyond invariant GNNs by further recovering the direction of the processed scalars for the update of equivariant features.

EGNN [5]. EGNN is one of the most famous scalarization based models, and it can be considered as an equivariant enhancement of two prior works, SchNet [47] and Radial Field [63]. For its message function $\phi_{\mathrm{msg}}(\cdot)$, it first applies the relative distance for the update of invariant message, which is then multiplied back with the relative coordinate to derive directional message. The form of $\phi_{\mathrm{msg}}(\cdot)$ is as follows:

$$m_{ij} = \sigma_1\left(h_i, h_j, \|\vec{x}_i - \vec{x}_j\|^2, e_{ij}\right), \tag{15}$$

$$\vec{m}_{ij} = (\vec{x}_i - \vec{x}_j)\sigma_2\left(m_{ij}\right), \tag{16}$$

while the update function $\phi_{\mathrm{upd}}(\cdot)$ takes the following form,

$$h_i' = \sigma_3\left(h_i, \sum_{j\in\mathcal{N}_i}m_{ij}\right), \tag{17}$$

$$\vec{x}_i' = \vec{x}_i + \gamma\sum_{j\in\mathcal{N}_i}\vec{m}_{ij}, \tag{18}$$

where $\sigma_1, \sigma_2, \sigma_3$ are all instantiated as Multi-Layer Perceptrons (MLPs), and γ is a predefined constant.

GMN [51]. In practice, each node is usually associated with multiple geometric features besides 3D position, such as velocity and force. Therefore, GMN proposes a multi-channel version of EGNN by defining a multi-channel vector $\vec{V}_i \in \mathbb{R}^{3\times C}$ for node i, where different channel (column) indicates different kind of geometric vector. In the message computation, the multi-channel vectors interact through inner product and are properly normalized for more training stability just before they are fed into the MLP, i.e.,

$$m_{ij} = \sigma_1\left(h_i, h_j, \frac{\vec{V}_{ij}^\top\vec{V}_{ij}}{\|\vec{V}_{ij}^\top\vec{V}_{ij}\|_F}, e_{ij}\right), \tag{19}$$

$$\vec{M}_{ij} = \vec{V}_{ij}\sigma_2\left(m_{ij}\right), \tag{20}$$

where $\vec{V}_{ij}$ is a translation-invariant directional matrix related to $\vec{V}_i$ and $\vec{V}_j$; for instance, if we have $\vec{V}_i = [\vec{x}_i, \dot{\vec{x}}_i]$ where $\dot{\vec{x}}_i \in \mathbb{R}^3$ defines the velocity, then we can either choose the direct subtraction $\vec{V}_{ij} = \vec{V}_i - \vec{V}_j$, or the concatenate form

$\vec{V}_{ij} = [\vec{V}_i, \vec{V}_j]$ where the first channel of $\vec{V}_i$ and $\vec{V}_j$ is made translation invariant by subtracting the mean coordinate [65]. The update process is analogous to Eqs. (17) and (18), but extended to the multi-channel fashion as well.

PaiNN [6]. By initializing the multi-channel equivariant features to be zeros, namely, letting $\vec{V}_i = \vec{0} \in \mathbb{R}^{3 \times C}$, PaiNN iteratively updates $\vec{V}_i$ as well as invariant feature h_i via the fixed relative position of the input coordinates $\vec{x}_{ij} = \vec{x}_i - \vec{x}_j$ in each layer, with the help of residual connection and gated non-linearity. We rewrite and somehow generalize the original form proposed by [6] using our consistent denotations. The messages are given by:

$$m_{ij} = \sigma_1\left(h_j, \|\vec{x}_{ij}\|^2, e_{ij}\right), \tag{21}$$

$$\vec{M}_{ij} = \vec{V}_j \sigma_2\left(m_{ij}\right) + \vec{x}_{ij} \sigma_3\left(m_{ij}\right), \tag{22}$$

and the update functions are calculated as:

$$m_i = h_i + \sum_{j \in \mathcal{N}(i)} m_{ij}, \tag{23}$$

$$\vec{M}_i = \vec{V}_i + \sum_{j \in \mathcal{N}(i)} \vec{M}_{ij}, \tag{24}$$

$$h'_i = m_i + \sigma_4\left(m_i, \|\vec{M}_i\|\right), \tag{25}$$

$$\vec{V}'_i = \vec{M}_i + \vec{M}_i \sigma_5\left(m_i, \|\vec{M}_i\|\right), \tag{26}$$

where the functions $\sigma_1-\sigma_5$ are non-linear invariant scaling functions. In Eqs. (25) and (26), "$\|\cdot\|$" outputs a multi-channel scalar each channel of which computes the vector norm of each channel of the input matrix.

Local frames [60–62]. These methods construct local *frames* (i.e., reference frames) $\vec{F} \in \mathbb{R}^{3 \times 3}$ that are equivariant to rotations and can be utilized to project the geometric information into invariant representations. In particular, LoCS [61] and Aether [61] leverage the angular position $\vec{w}_i \in \mathbb{R}^3$ of each node i to construct node-wise local frames $\vec{F}_i = R(\vec{w}_i)$ where $R(\vec{w}_i) \in \mathbb{R}^{3 \times 3}$ is the corresponding rotation matrix of the angular position $\vec{w}_i$. ClofNet [60] instead builds up edge-wise local frames $\vec{F}_{ij} = [\vec{a}_{ij}, \vec{b}_{ij}, \vec{c}_{ij}]$, with

$$\vec{a}_{ij} = \frac{\vec{x}_{ic} - \vec{x}_{jc}}{\|\vec{x}_{ic} - \vec{x}_{jc}\|}, \vec{b}_{ij} = \frac{\vec{x}_{ic} \times \vec{x}_{jc}}{\|\vec{x}_{ic} \times \vec{x}_{jc}\|},$$
$$\vec{c}_{ij} = \vec{a}_{ij} \times \vec{b}_{ij}. \tag{27}$$

Here $\vec{x}_{ic} = \vec{x}_i - \vec{x}_c$ is translation-invariant by subtracting the center of mass $\vec{x}_c = \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i$ so that the frame $\vec{F}_{ij}$ is also translation-invariant.

With local frames, the invariant message m_{ij} is generated as

$$m_{ij} = \sigma_1\left(h_i, h_j, \vec{V}_{ij}^\top \vec{F}_{ij}\right), \tag{28}$$

where $\vec{V}_{ij}$ is the translation-invariant geometric information between node i and j, similar to the considerations in GMN (Eq. (20)). ClofNet additionally considers to project the invariant message into an equivariant counterpart:

$$\vec{M}_{ij} = \vec{F}_{ij} \sigma_2\left(m_{ij}\right). \tag{29}$$

There are other works that exploit the scalarization

technique to permit equivariance. GVP-GNN [64] first performs channel-wise linear projection of the input vector to align the channel dimension, and then computes the normalization of the projected vector as the scalar that is multiplied with the vector as the output vector. During this process, GVP-GNN does not pass the information from the input scalars, which is different from EGNN where the input scalars also influence the update of the vector. EGHN [65], built upon GMN, leverages a hierarchical encoder-decoder mechanism to represent the multi-body interaction with specially-designed equivariant pooling and unpooling modules. FastEGNN [66] addresses large-scale geometric graph scenarios by employing a small ordered set of virtual nodes, which minimizes the number of required edges and enhances computational efficiency. In LEFTNet [69], a local hierarchy of 3D isomorphism is proposed to evaluate the expressive power of equivariant GNNs and investigate the process of representing global geometric information from local patches. This work leads to two crucial modules for designing expressive and efficient geometric GNNs: local substructure encoding and frame transition encoding. SaVeNet [70] enhances the numerical stability of the model by introducing gradually decaying directional noise during the training phase. ViSNet [71] employs vector-scalar interactive message passing to implicitly extract various geometric features. QuinNet [72] integrates many-body interactions, extending this modeling to include interactions of up to five bodies. Furthermore, HEGNN [73] leverages the inner product of high-degree steerable features to enhance scalar messaging, thereby achieving a balance between efficiency and effectiveness. Additionally, as scalars can be combined with various other invariant information, ETNN [74] further amplifies the expressiveness of the model by introducing deep topological learning constructs. EquiLLM [75] enhances the representation of invariant scalars through knowledge injection from large language models, and can be flexibly generalized to various geometry tasks.

For all above methods, the scalarization process is implemented via the inner-product operator. In contrast to this, Frame-Averaging [67] proposes to ensure equivariance via this averaging process: $\frac{1}{|G|}\sum_{g \in G} g \cdot \sigma(g^{-1} \cdot \vec{x})$, where σ is an arbitrary MLP and the term $g^{-1} \cdot \vec{x}$ makes the input invariant. To deal with the case when the cardinality of G is large, [67] instead conduct an average over a carefully selected subset that is obtained by the so-called frame function. The idea of Frame-Averaging is latter exploited in the field of material design [68].

4.3.2 High-degree steerable models

For the aforementioned scalarization-based models, the node variables to be updated include invariant scalars h_i and equivariant vectors $\vec{x}_i$ (or $\vec{V}_i$ for the multi-channel case), and the 3D rotation representation throughout the network is the rotation matrix R_g. It will be observed that scalars and vectors are respectively type-0 and type-1 steerable features, and the rotation matrix is the 1st degree matrix of a more general rotation representation. We will show that it is possible to derive high-degree representations of steerable features

beyond scalars and vectors in equivariant GNNs.

Prior to the introduction of high-degree models, we first introduce the concepts: **1.** Wigner-D matrices [97] to convert 3D rotations to group representations of different degree; **2.** spherical harmonics [98] to convert 3D vectors to steerable features of different type; **3.** Clebsch-Gordan (CG) tensor product [99] to perform equivariant mapping between steerable features.

Wigner-D matrices. In the general high-degree case, a widely studied genre of the representation for the rotation group SO(3) is the irreducible representation [97]:

$$\rho(g):=\boldsymbol{D}^{(l)}(g) \in \mathbb{R}^{(2l+1)\times(2l+1)}, g \in \text{SO}(3), \tag{30}$$

where $\boldsymbol{D}^{(l)}(g)$ is the Wigner-D matrix[7] of degree l, and $l \in \mathbb{N} = \{0, 1, 2, \dots\}$. In particular, $\boldsymbol{D}^{(0)}(g) = 1$ reduces to trivial representation and $\boldsymbol{D}^{(1)}(g) = \boldsymbol{R}_g$ takes the form of the rotation matrix. The steerability of a type-l feature $\vec{\boldsymbol{x}}^{(l)} \in \mathbb{R}^{2l+1}$ is defined as $\boldsymbol{D}^{(l)}(g)\vec{\boldsymbol{x}}^{(l)}$, which naturally unifies the aforementioned invariant features and equivariant features by restricting $l = 0$ and $l = 1$, separately. Provided that there could be steerable features of multiple types and multiple channels, we provide a general form of steerable features:

$$\vec{\mathbb{V}}^{(\mathbb{L})} := \left\{ \vec{\boldsymbol{V}}^{(l)} \in \mathbb{R}^{(2l+1)\times C_l} \right\}_{l \in \mathbb{L}}, \tag{31}$$

where $\mathbb{L}$ is the set consisting of all possible types and C_l is the number of channels for type l. Since we are addressing geometric graphs in this paper, we will specify the steerable features of node i as $\vec{\mathbb{V}}_i^{(\mathbb{L})}$ and its type-l component as $\vec{\boldsymbol{V}}_i^{(l)}$.

Spherical harmonics. We have defined how to steer type-l features via Wigner-D matrices, but we do not know yet how to obtain type-l features given 3D coordinates. Spherical harmonics are such tools to serve this purpose. Spherical harmonics are a set of Fourier basis on the unit sphere S^2. They map 3D vectors on the unit sphere S^2 into $(2l+1)$-dimensional vector space[8]. That is,

$$Y^{(l)}(\vec{\boldsymbol{x}}) : S^2 \mapsto \mathbb{R}^{2l+1}, \tag{32}$$

where $\vec{\boldsymbol{x}}$ is a unit vector on the sphere, and the elements in $Y^{(l)}$ are usually used together and denoted as $[Y_{-l}^{(l)}, Y_{-l+1}^{(l)}, \cdots, Y_{l-1}^{(l)}, Y_l^{(l)}]$ where different element is called different order. It can also be generalized to take arbitrary 3D vector as input by properly normalizing the vector as $\frac{\vec{\boldsymbol{x}}}{\|\vec{\boldsymbol{x}}\|}$ prior to feeding into the spherical harmonics. This offers a unified view of transition to vector spaces of arbitrary type, where scalars correspond to $Y^{(0)}(\vec{\boldsymbol{x}}) = 1$ when $l = 0$, and vectors correspond to $Y^{(1)}(\vec{\boldsymbol{x}}) = \vec{\boldsymbol{x}} \in \mathbb{R}^3$ when $l = 1$. More importantly, spherical harmonics are equivariant in terms of Wigner-D matrices:

$$Y^{(l)}(\boldsymbol{R}_g\vec{\boldsymbol{x}}) = \boldsymbol{D}^{(l)}(g)Y^{(l)}(\vec{\boldsymbol{x}}), g \in \text{SO}(3), \tag{33}$$

where $\boldsymbol{R}_g \in \mathbb{R}^{3\times3}$ is the rotation matrix and $D(g) \in \mathbb{R}^{(2l+1)\times(2l+1)}$ refers to the Wigner-D matrix of degree l. To create multi-type multi-channel steerable features, we apply $Y^{(l)}$ over multiple copies for each type in $\mathbb{L}$, yielding $\mathbb{Y}^{(\mathbb{L})}$.

Clebsch-Gordan (CG) tensor product. Although spherical harmonics offer a way to design equivariant mapping from 3D coordinates (type-1 features) to type-l features, they are unable to depict the interactions between steerable features of arbitrary types, which, however, is central to the design of equivariant functions when their input contains steerable features of various types. Fortunately, CG tensor product provides a tractable solution to this issue [99]. It derive $\vec{\boldsymbol{V}}^{(l)} \in \mathbb{R}^{(2l+1)\times C}$ from two multi-channel steerable features $\vec{\boldsymbol{V}}^{(l_1)} \in \mathbb{R}^{(2l_1+1)\times C_1}, \vec{\boldsymbol{V}}^{(l_2)} \in \mathbb{R}^{(2l_2+1)\times C_2}$ by:

$$\vec{\boldsymbol{V}}^{(l)} = \left[\vec{\boldsymbol{V}}^{(l_1)} \otimes_{\text{cg}}^{W} \vec{\boldsymbol{V}}^{(l_2)} \right]^{(l)}, \tag{34}$$

which can be expanded in details by:

$$v_{m,c}^{(l)} = \sum_{\substack{c_1=1 \\ c_2=1}}^{C_1,C_2} w_{c_1c_2c} \sum_{\substack{m_1=-l_1 \\ m_2=-l_2}}^{l_1,l_2} Q_{(l_1,m_1)(l_2,m_2)}^{(l,m)} v_{m_1,c_1}^{(l_1)} v_{m_2,c_2}^{(l_2)}, \tag{35}$$

where $v_{m,c}^{(l)}$ indicates the mth order and cth channel of $\vec{\boldsymbol{V}}^{(l)}$; $Q_{(l_1,m_1)(l_2,m_2)}^{(l,m)}$ are the Clebsch-Gordan (CG) coefficients [99] and are zeros unless $|l_1 - l_2| \leqslant l \leqslant l_1 + l_2$; $w_{c_1c_2c}$ is the learnable parameter in the parameter matrix $\boldsymbol{W} \in \mathbb{R}^{C_1\times C_2\times C}$, and when $\boldsymbol{W}$ are all ones, Eq. (35) reduces to the traditional non-parametric CG tensor product.

One promising property of CG tensor product is that it is SO(3)-equivariant regarding Wigner-D matrices, implying that $\forall g \in \text{SO}(3)$,

$$\boldsymbol{D}^{(l)}(g)\vec{\boldsymbol{V}}^{(l)} = \left[\left(\boldsymbol{D}^{(l_1)}(g) \vec{\boldsymbol{V}}^{(l_1)} \right) \otimes_{\text{cg}}^{W} \left(\boldsymbol{D}^{(l_2)}(g) \vec{\boldsymbol{V}}^{(l_2)} \right) \right]^{(l)}. \tag{36}$$

For simplicity, the steerable variables in Eq. (34) are all of a single type. It is tractable to generalize Eq. (34) to the multi-type case by employing it over each combination of input-output type, and assigning different learnable parameters accordingly, which leads to a general form as follows:

$$\vec{\mathbb{V}}^{(\mathbb{L})} = \vec{\mathbb{V}}^{(\mathbb{L}_1)} \otimes_{\text{cg}}^{\mathbb{W}} \vec{\mathbb{V}}^{(\mathbb{L}_2)}. \tag{37}$$

With the above building blocks, we below introduce several prevailing high-degree steerable models where the updated steerable variables for each node are $\vec{\mathbb{V}}_i^{(\mathbb{L})}$.

TFN [7]. With our formulation for the high-degree steerable operations, Tensor Field Network (TFN) computes the following equivariant point convolution:

$$\vec{\mathbb{M}}_{ij}^{(\mathbb{L})} = \mathbb{Y}^{(\mathbb{L})}\left(\frac{\vec{\boldsymbol{x}}_{ij}}{\|\vec{\boldsymbol{x}}_{ij}\|} \right) \otimes_{\text{cg}}^{\mathbb{W}} \vec{\mathbb{V}}_j^{(\mathbb{L})}, \tag{38}$$

where $\vec{\boldsymbol{x}}_{ij} = \vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j$ is the radial vector, and the element in $\mathbb{W}$ is generated by a radial MLP $f(\|\vec{\boldsymbol{x}}_{ij}\|)$ upon the distance $\|\vec{\boldsymbol{x}}_{ij}\|$. Here $\vec{\boldsymbol{x}}_i$ are fixed as the initial coordinates of the input data. The update of each node is implemented as a series of operations including aggregation:

$$\vec{\mathbb{U}}_i^{(\mathbb{L})} = \vec{\mathbb{V}}_i^{(\mathbb{L})} + \sum_{j \in \mathcal{N}(i)} \vec{\mathbb{M}}_{ij}^{(\mathbb{L})}, \tag{39}$$

and self-interaction:

$$\vec{\mathbb{V}}_i^{(\mathbb{L})} = \{\vec{\boldsymbol{U}}^{(l)}\boldsymbol{W}^{(l)}\}_{l \in \mathbb{L}}, \tag{40}$$

---

[7] Wigner-D matrices lie in the complex space, but they can be transformed to the real space under appropriate bases.
[8] Similar to Wigner-D matrices, the output of spherical harmonics are complex but can be transformed into real space under certain bases.

where $\boldsymbol{W}^{(l)} \in \mathbb{R}^{c_l \times c_l}$ is the learnable channel-mixing matrix for each type l, and node-wise non-linearity:

$$\vec{\mathbb{V}}'^{(\mathbb{L})} = \left\{ \vec{\boldsymbol{V}}^{(l)} \sigma \left(\|\vec{\boldsymbol{V}}^{(l)}\|_2 + \boldsymbol{b}^{(l)} \right) \right\}_{l \in \mathbb{L}}, \tag{41}$$

where $\sigma(\cdot)$ is an activation function, "$\|\cdot\|_2$" is the L_2 vector norm over the order dimension (with size $(2l+1)$) of $\vec{\boldsymbol{V}}^{(l)}$, and $\boldsymbol{b}^{(l)} \in \mathbb{R}^{c_l}$ is the bias for type l.

SEGNN [9]. SEGNN enhances TFN from equivariant point convolution to general equivariant message passing. Firstly, SEGNN involves high-degree geometric features from both node i and j in message computation by deriving $\vec{\mathbb{V}}_{ij}^{(\mathbb{L})} = \vec{\mathbb{V}}_i^{(\mathbb{L})} \oplus \vec{\mathbb{V}}_j^{(\mathbb{L})} \oplus \{\|\vec{\boldsymbol{x}}_{ij}\|^2\}$, where, again, $\vec{\boldsymbol{x}}_{ij} = \vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j$ is the radial vector, and "$\oplus$" denotes concatenation along the channel dimension for the steerable features with the same type $l \in \mathbb{L}$. For example,

$$\vec{\mathbb{V}}_1^{(\mathbb{L})} \oplus \vec{\mathbb{V}}_2^{(\mathbb{L})} := \{\vec{\boldsymbol{V}}_1^{(l)} \| \vec{\boldsymbol{V}}_2^{(l)}\}_{l \in \mathbb{L}}. \tag{42}$$

Here, "$\|$" stands for concatenation along the channel. Subsequently, the high-degree linear message passing specified in Eq. (38) is extended to a non-linear fashion via gated non-linearities [100]:

$$\vec{\mathbb{V}}_{ij}^{(\mathbb{L})}, \boldsymbol{g}_{ij} = \mathbb{Y}^{(\mathbb{L})} \left(\frac{\vec{\boldsymbol{x}}_{ij}}{\|\vec{\boldsymbol{x}}_{ij}\|} \right) \otimes_{\mathrm{cg}}^{\mathbb{W}_1} \vec{\mathbb{V}}_{ij}^{(\mathbb{L})}, \tag{43}$$

$$\vec{\mathbb{M}}_{ij}^{(\mathbb{L})} = \mathrm{Gate}\left(\vec{\mathbb{V}}_{ij}^{(\mathbb{L})}, \mathrm{Swish}(\boldsymbol{g}_{ij}) \right), \tag{44}$$

where $\mathrm{Gate}(\cdot)$ is the gated non-linearity introduced in [100], $\mathrm{Swish}(\cdot)$ is the Swish activation [101], and $\boldsymbol{g}_{ij}$ is a scalar read out from the CG tensor product that will further be leveraged to control the scale in the non-linearity of Eq. (44). Notably, the CG product and non-linearity in Eqs. (43) and (44) are performed twice in the implementation of [9]. Analogous to the design of multi-layer perceptrons (MLPs), they are dubbed the steerable MLP.

The update function also employs the proposed steerable MLP. In detail,

$$\vec{\mathbb{V}}_i^{(\mathbb{L})}, \boldsymbol{g}_i = \left(\sum_{j \in \mathcal{N}_i} \mathbb{Y}^{(\mathbb{L})} \left(\frac{\vec{\boldsymbol{x}}_{ij}}{\|\vec{\boldsymbol{x}}_{ij}\|} \right) \right)$$
$$\otimes_{\mathrm{cg}}^{\mathbb{W}_2} \left(\vec{\mathbb{V}}_i^{(\mathbb{L})} + \sum_{j \in \mathcal{N}_i} \vec{\mathbb{M}}_{ij}^{(\mathbb{L})} \right), \tag{45}$$

$$\vec{\mathbb{V}}_i'^{(\mathbb{L})} = \vec{\mathbb{V}}_i^{(\mathbb{L})} + \mathrm{Gate}\left(\vec{\mathbb{V}}_i^{(\mathbb{L})}, \mathrm{Swish}(\boldsymbol{g}_i) \right). \tag{46}$$

Besides those have been introduced above, there are still many methods to build equivariant models with high-degree steerable features. Cormorant [76] utilizes channel-wise CG product (a reduced and more efficient form of Eq. (34) that acts on each input channel independently) and channel concatenations to formulate one-body and two-body interactions among the input graph systems. NequIP [10] improves the convolutional layer in TFN [7] by further introducing the radial Bessel functions and a polynomial envelope function used in DimeNet [3] to get a better embedding of interaction distance, thereby improving the performance of the model. SCN [78] regards each node embedding as a set of spherical functions (i.e., the spherical harmonics), then conducts message passing by rotating the embeddings based on the 3D edge orientation, and finally updates the node embeddings via discrete spherical Fourier Transform. Its following work, eSCN [79] proposes to reduce the computation complexity of the equivariant convolution on SO(3) with a mathematically equivalent one on SO(2). To enable higher body interaction beyond the two-body modeling in most previous papers, MACE [80] and Allegro [77], propose a simplified algorithm to construct the tensor product item, motivated by a new technology in physics called Atomic Cluster Expansion (ACE) [102–104].
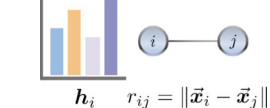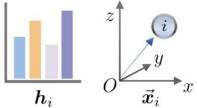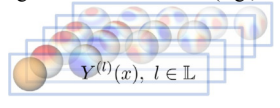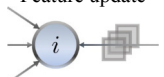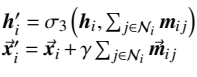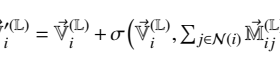
An illustrative comparison of invariant GNNs, scalarization-based equivariant GNNs, and high-degree steerable equivariant GNNs is summarized in Table 2.

4.4 Geometric graph transformers

Inspired by the significant success of Transformers [105,106] in many areas, such as natural language processing and computer vision, there have been efforts to apply these self-attention-based architectures to data structure like graphs or even geometric graphs in the scope of this survey. Summarized in Fig. 4, these methods stem from different types of geometric representations, including invariant representation, scalarization-based equivariant representation, and high-degree steerable representation, which have been elaborated in Section 4. Below we discuss these Transformers in detail.

Graphormer [81,82]. Graphormer has been firstly proposed as a powerful Transformer architecture operating on graphs,

Table 2 Illustrations of representative models for invariant GNNs, scalarization-based GNNs and high-degree steerable GNNs. Notably, these three types of models are able to process geometric features of different degrees

	Invariant GNNs (e.g., SchNet [47])	Scalarization-Based Models (e.g., EGNN [5])	High-Degree Steerable Models (e.g., TFN [7])
	$\boldsymbol{h}_i \quad r_{ij} = \|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\|$	$\boldsymbol{h}_i \quad \vec{\boldsymbol{x}}_i$	$Y^{(l)}(x),\ l \in \mathbb{L}$
Message computation	$\boldsymbol{m}_{ij} = \sigma_2(r_{ij})\sigma_1(\boldsymbol{h}_j)$	$\boldsymbol{m}_{ij} = \sigma_1\left(\boldsymbol{h}_i, \boldsymbol{h}_j, \|\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j\|^2, \boldsymbol{e}_{ij}\right)$ $\vec{\boldsymbol{m}}_{ij} = (\vec{\boldsymbol{x}}_i - \vec{\boldsymbol{x}}_j)\sigma_2\left(\boldsymbol{m}_{ij}\right)$	$\vec{\mathbb{M}}_{ij}^{(\mathbb{L})} = \mathbb{Y}^{(\mathbb{L})}\left(\frac{\vec{\boldsymbol{x}}_{ij}}{\|\vec{\boldsymbol{x}}_{ij}\|}\right) \otimes_{\mathrm{cg}}^{\mathbb{W}} \vec{\mathbb{V}}_j^{(\mathbb{L})}$
Feature update	$\boldsymbol{h}_i' = \sigma_3\left(\boldsymbol{h}_i, \sum_{j \in \mathcal{N}_i} \boldsymbol{m}_{ij}\right)$	$\boldsymbol{h}_i' = \sigma_3\left(\boldsymbol{h}_i, \sum_{j \in \mathcal{N}_i} \boldsymbol{m}_{ij}\right)$ $\vec{\boldsymbol{x}}_i'' = \vec{\boldsymbol{x}}_i + \gamma \sum_{j \in \mathcal{N}_i} \vec{\boldsymbol{m}}_{ij}$	$\vec{\mathbb{V}}_i'^{(\mathbb{L})} = \vec{\mathbb{V}}_i^{(\mathbb{L})} + \sigma\left(\vec{\mathbb{V}}_i^{(\mathbb{L})}, \sum_{j \in \mathcal{N}(i)} \vec{\mathbb{M}}_{ij}^{(\mathbb{L})}\right)$

equipped with centrality encoding, spatial encoding, and edge encoding [81]. With its success on challenging 2D graph datasets, e.g., the OGB-LSC Challenge [107], it has been subsequently extended to work on geometric graphs with special designs in computing the encodings. To be specific, the spatial encoding, which aims to measure the spatial relation between node i and j in $\vec{\mathcal{G}}$, is chosen to be the Euclidean distance $\|\vec{x}_i - \vec{x}_j\|_2$ transformed by Gaussian basis functions [108]. The centrality encoding is derived as a summation of the spatial encodings over the connected edges for each node. The encodings are then utilized in computing the self-attention, and layer normalization is also adopted for the intermediate features. Notably, all representations are E(3)-invariant under the construction of Graphormer. In order to make it suitable for E(3)-equivariant prediction tasks, [81] proposes to use a projection head as the final block, which aggregates the edge vectors, scaled by their corresponding attention weights to obtain a node-wise vector as output:

$$\vec{f}_i = \sum_{j \neq i} a_{ij}(\vec{x}_i - \vec{x}_j), \tag{47}$$

where a_{ij} is the E(3)-invariant attention weight between node i and j.

TorchMD-Net [83]. TorchMD-Net is an equivariant Transformer that tackles general multi-channel geometric vectors in a scalarization-based manner, akin to PaiNN [6]. Yet, in the process of attention computation, only invariant representations h_i and distances $\|\vec{x}_{ij}\|$ are involved. Specifically, the distance is firstly embedded by two MLPs σ_{d_K} and σ_{d_V} for the key and value, respectively:

$$d_{ij}^K = \sigma_{d_K}(e_{\text{RBF}}^{(ij)}), \quad d_{ij}^V = \sigma_{d_V}(e_{\text{RBF}}^{(ij)}), \tag{48}$$

where $e_{\text{RBF}}^{(ij)}$ is the radial basis function representation of distance $\|\vec{x}_{ij}\|$, similar to Eq. (7). The query, key, and value are given by linear transformations of the input scalar features:

$$q_i = h_i W_Q, k_{ij} = h_i W_K \odot d_{ij}^K, v_{ij} = h_j W_V \odot d_{ij}^V, \tag{49}$$

where "$\odot$" is the element-wise product. Instead of traditionally adopted Softmax operator [105], TorchMD-Net simplifies to SiLU non-linearity:

$$a_{ij} = \sum_{C_h} \text{SiLU}\left(q_i \odot k_{ij}\right) \cdot \text{Cutoff}\left(\|\vec{x}_{ij}\|\right), \tag{50}$$

with Cutoff($\cdot$) being a cosine cutoff on the distance and the summation being over the channels of these invariant features. Finally, the output of the attention is yielded as

$$h_i' = \left(\sum_{j \in \mathcal{N}_i} a_{ij} v_{ij}\right) W_O, \tag{51}$$

with W_O being a linear transformation for the output.

SE(3)-**Transformer** [8]. Different from Graphormer and TorchMD-Net that limit the representation to scalars and vectors with degree $l \in \{0, 1\}$, SE(3)-Transformer employs attention mechanism on general steerable features with high degree. Following our notations introduced in Section 4.3.2, we describe the attention computation as follows.

The point-wise query $\vec{\mathbb{Q}}_i^{(\mathbb{L})}$ and pairwise key $\vec{\mathbb{K}}_{ij}^{(\mathbb{L})}$ and value $\vec{\mathbb{V}}_{ij}^{(\mathbb{L})}$ are derived as:

$$\vec{\mathbb{Q}}_i^{(\mathbb{L})} = \mathbf{1} \otimes_{\text{cg}}^{\mathbb{W}_Q} \vec{\mathbb{V}}_i^{(\mathbb{L})},$$

$$\vec{\mathbb{K}}_{ij}^{(\mathbb{L})} = \mathbb{Y}^{(\mathbb{L})}\left(\frac{\vec{x}_{ij}}{\|\vec{x}_{ij}\|}\right) \otimes_{\text{cg}}^{\mathbb{W}_K} \vec{\mathbb{V}}_j^{(\mathbb{L})},$$

$$\vec{\mathbb{V}}_{ij}^{(\mathbb{L})} = \mathbb{Y}^{(\mathbb{L})}\left(\frac{\vec{x}_{ij}}{\|\vec{x}_{ij}\|}\right) \otimes_{\text{cg}}^{\mathbb{W}_V} \vec{\mathbb{V}}_j^{(\mathbb{L})}. \tag{52}$$

The attention coefficient a_{ij} is computed as a Softmax aggregation over the neighbors with message being the inner products of the queries and keys, ensuring rotation invariance:

$$\alpha_{ij} = \frac{\exp\left(\vec{\mathbb{Q}}_i^{(\mathbb{L})} \cdot \vec{\mathbb{K}}_{ij}^{(\mathbb{L})}\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\vec{\mathbb{Q}}_i^{(\mathbb{L})} \cdot \vec{\mathbb{K}}_{ik}^{(\mathbb{L})}\right)}. \tag{53}$$

The attention is then utilized to aggregate the values and update the node feature:

$$\vec{\mathbb{V}}_i'^{(\mathbb{L})} = \mathbf{1} \otimes_{\text{cg}}^{\mathbb{W}_1} \vec{\mathbb{V}}_i^{(\mathbb{L})} + \sum_{j \in \mathcal{N}_i} \alpha_{ij} \vec{\mathbb{V}}_{ij}^{(\mathbb{L})}. \tag{54}$$

With the invariant attention, the updated feature is easily guaranteed to satisfy SE(3)-equivariance.

Besides, LieTransformer [84] extends the idea of LieConv [54] by building attentions on top of lifting and sampling on Lie groups. GVP-Transformer introduced in [85] leverages GVP-GNN [64] as the structural encoder and applies a generic Transformer over the extracted representation, exhibiting strong performance in learning inverse folding of proteins. Equiformer [11] proposes to replace dot product attention in Transformers by MLP attention and non-linear message passing, building upon the space of high-degree steerable tensors. EquiformerV2 [86] further incorporates eSCN [79] in the architecture for efficient modeling and introduces more technical enhancements like specially designed attention re-normalization and layer normalization for better empirical performance. Geoformer [87] develops an invariant module called Interatomic Positional Encoding (IPE) based on the invariant basis from ACE, in order to enhance the expressiveness of many-body contributions in the attention blocks. Recently, SO3KRATES [88] proposed a technique aimed at leveraging the advantages of high-degree representations while simplifying the complexity inherent in tensor products. This approach focuses on the design of a model that utilizes only the paths that yield scalars in tensor products. Later, GotenNet [89] broadened the scope of the inner product form, creating a multi-channeled version and referring to models that employ this methodology as *spherical-scalarization models*. GotenNet integrated the inner product with the original attention mechanism, resulting in an efficient equivariant transformer architecture.

As previous transformers typically focus on a specific domain, either proteins or small molecules. EPT [90] proposes a novel pretraining framework designed to harmonize the geometric learning of small molecules and proteins. It unifies the geometric modeling of multi-domain molecules via block-enhanced representation upon an PaiNN-based transformer framework.

4.5 Theoretical analysis on expressivity

In machine learning, an important criterion for measuring the expressiveness of a network is whether it has *universal*

approximation property. In the task of learning on geometric graphs, this is whether any function of geometric graphs can be approximated by geometric GNNs with arbitrary accuracy.

An initial attempt to explore this problem is conducted by [109], which proves the universality of the high-degree steerable model, i.e., TFN [7], over point clouds (namely fully-connected geometric graphs) by showing that TFN can fit any equivariant polynomials. GemNet [4] further demonstrates that the universality holds with just spherical representations other than the full SO(3) representations that are required in the proof of [109]. Later, the GWL framework [96] defines a geometric version of the Weisfeiler-Lehman (WL) test [110] to study the expressive power of geometric GNNs operating on sparse graphs from the perspective of discriminating geometric graphs, and discuses the difference of the expressivity between various invariant and equivariant GNNs, both theoretically and experimentally. One crucial conclusion drawn by the GWL paper is that GWL is strictly more powerful than invariant GWL, showing the advantage of equivariant GNNs against invariant GNNs. For fully-connected geometric graphs, invariant GWL has the same expressive power as GWL. More recently, HEGNN [73] has provided both theoretical and experimental insights into the necessity of employing high-degree steerable features on symmetric graphs. Specifically, under the strict equivariance constraint, the degradation of representations of certain degrees on symmetry graphs cannot be avoided unless it is circumvented by relaxing some conditions (e.g., probabilistic symmetry breaking in SymPE [111]). Furthermore, HEGNN establishes a connection between high-degree steerable features and Legendre polynomials, indicating that inner-product of sufficiently high-degree representations can recover all angular information present in geometric graphs.

There are other works that only investigate the universality of the message computation function [46,51]. They explore the expressivity of the scalarization-based models (e.g., EGNN), and [46] confirms that the scalarization-based methods can universally approximate any invariant/ equivaraint functions of vectors. Besides, SGNN [25] generalizes from equivariance to subequivariance that depicts the case when part of the symmetry is broken by external force field, e.g., gravity, and finally design an universal form of subequivariant functions.

5 Applications

In this section, we systematically review the applications related to geometric graph learning. We classify existing methods according to the system types they work on, which leads to the categorization of tasks on particle, (small) molecule, protein, molecule + molecule (Mol + Mol), molecule + protein (Mol + Protein), protein + protein, and other domains, as summarized in Table 3. We also provide a summary of all related datasets of single- and multiple-instance tasks in Tables 4 and 5, respectively. **It is worth mentioning that our discussion primarily focuses on the methods utilizing geometric GNNs, although other methods, such as sequence-based approaches, may be applicable in certain applications**.

5.1 Tasks on particles

The particle representation serves as an abstract and unified concept in the context of dynamic modeling in physics. Rigid bodies, elastic bodies and even fluid can be modeled as a set of particles [25]. Under such a particle-based modeling, a physical object of interest corresponds to a geometric graph $\vec{\mathcal{G}}$ as specified in Definition 4, where different particles are modeled as different nodes, and physical interactions between particles such as attraction/repulsion force, collision, rolling, and sliding are denoted as edge connections.

5.1.1 Physical dynamics simulation

Geometric GNNs have been widely applied to characterize the process of general physical dynamics. One typical example is N-body simulation, which is originally proposed by [27] and targets at modeling the dynamics of a prototype system composed of N interacting particles. While it is built under an ideal condition, an N-body system is capable of representing various physical phenomena across a spectrum encompassing quantum physics through to astronomy, by accommodating diverse interactions. Other examples include the simulation of physical scenes that involves more complex objects including fluids, rigid-bodies, deformable-bodies, and human motions.

Task definition: Given the initial state of the system represented by a geometric graph $\vec{\mathcal{G}}^{(0)}$, the future states of all N particles after a period of k steps are predicted by a parametric function:

$$\vec{X}^{(t+k)} = \phi_\theta(\vec{\mathcal{G}}^{(t)}). \tag{55}$$

In contrast to the above single-state prediction setting, one may also conduct a "roll-out" simulation by recurrently taking the predicted output of current state as the input for the prediction of the next state. Furthermore, it can also be extended to the spatio-temporal setting by taking the historical geometric graphs within a window of size w (namely $\vec{\mathcal{G}}^{(t-w+1:t)}$) as input, rather than a single input frame (namely $\vec{\mathcal{G}}^{(t)}$) in Eq. (55).

Symmetry preserved: This is an E(3)-equivariant task, as the transformation of the initial state results in the same transformation of the predicted state. It means $g \cdot \phi_\theta(\vec{\mathcal{G}}) = \phi_\theta(g \cdot \vec{\mathcal{G}}), \forall g \in \mathrm{E}(3)$.

Datasets: The datasets used in current methods belong to the following classes: **1)** N-**body dataset series.** The original N-body dataset [27] presents an environment capable of simulating three types of system, including 1D phase-coupled oscillators, 2D springs, and 2D charged balls. The authors in [8] further generalize N-body to encompass 3D cases. Recently, the work [51] designs Constrained N-body by adding geometric constraints between particles, leading to a combination of diverse systems with isolated particles, sticks and hinges. Later, the systems derived by [65] further introduce the interactions between complex objects that are composed of multiple particles interconnected by rigid sticks. **2) Scene simulation datasets.** The paper [118] proposes four simulation environments: FluidFall, FluidShake, BoxBath, and RiceGrip, where the former two focus on fluid modeling, the third one tests fluid-rigid interactions, and the final one involves modeling deformable objects with elastic/plastic

Table 3 Summary of various geometric GNNs for different tasks. The generative tasks indicates the ones addressable by generative models, otherwise referred to as the non-generative tasks. The ones can be solved with either generative or non-generative models are dubbed as the mixed tasks

Data type	Task name	Task type	Methods
		Physics	
Particle	*N*-body simulation	Non-generative	NRI [27], IN [112], E-NFs [5], EGNN [5], SEGNNs [9], GMN [51], EGHN [65], HOGN [113], NCGNN [114], FastEGNN [66], HEGNN [73]
	Scene simulation	Non-generative	SGNN [25], GNS [26], GNS* [115], C-GNS [116], HGNS[117], DPI-Net [118], HRN [119], FIGNet [120], EGHN [65], LoCS [61], EqMotion [121], ESTAG [31], SEGNO [122], FastEGNN [66], HEGNN [73], EquiLLM [75]
		Biochemistry	
Small molecule	Molecular property prediction	Non-generative	Cormorant [76], TFN [7], SE(3)-Transformer [8], NequIP [10], SEGNNs [9], LieConv [54], Lietransformer [84], SchNet [47], DimeNet [3], GemNet [4], PaiNN [6], TorchMD-Net [83], Equiformer [11], SphereNet [123], EGNN [5], Graphormer [81,82], SCN [78], eSCN [79], GNN-LF [124], LEFTNet [69], SaVeNet [70], ViSNet [71], QuinNet [72], SO3KRATES [88], Gaunt [125], GotenNet [89]
	Molecular dynamics	Mixed	E-CNF [126], EGNN [5], NequIP [127], GMN [51], EGHN [65], NCGNN [114], ESTAG [31], EGNO [127], SEGNO [122], ITO [128], E-ACF [129], GeoTDM [130], HEGNN [73], StABlE [131], [132]
	Molecular generation	Generative	GeoDiff [133], GeoLDM [134], ConfVAE [135], ConfGF [136], G-SchNet [137], cG-SchNet [138], MDM [139], MolDiff [140], DGSM [141], E-NFs [142], EDM [143], GeoMol [144], Torsional Diffusion [30], MPerformer [145], EEGSDE [146], DMCG [147], HierDiff [148], EquiFM [149], CoarsenConf [150], GeoBFN [151], MolCRAFT [152]
	Pretraining	Mixed	3D-EMGP [153], GeoSSL-DDM [154], GraphMVP [155], GNS-TAT [156], MGMAE [157], 3D-Infomax [158], Uni-Mol [159], Transformer-M[160], MoleculeSDE [161], SliDe [162], Frad [163], DenoiseVAE [164], MolSpectra [165]
Protein	Protein property prediction	Non-generative	LM-GVP [166], DeepFRI [167], GearNet [168], 3DCNN [169], TM-align [170], GVP-GNN [64], PAUL [171], EDN [172], EnQA [173], ScanNet [174], EquiPocket [175], PocketMiner [176]
	Protein inverse folding	Generative	GVP-GNN [64], [177], ESM-IF1 [85], GCA [178], ProteinMPNN [179], PiFold [180], LM-Design [181], KW-Design [182]
	Protein folding	Generative	AlphaFold [33], AlphaFold2 [183], RosettaFold [12], RosettaFold2 [48], RFAA [184], EigenFold [185], RFdiffusion [13], Chroma [14], ESMFold [186], HelixFold-Single [187]
	Protein co-design	Generative	Chroma [14], RFdiffusion [13], PROTSEED [188], ReQFlow [189]
	Pretraining	Mixed	ProtTrans [190], xTrimoPGLM [191], ProtGPT2 [192], HJRSS [193], GearNet [168], ProFSA [194], PromptProtein [195], DrugCLIP [196], ESM-1b [197], ESM2 [186], Guo et al. [198], PAAG [199]
Mol + Mol	Linker design	Generative	DiffLinker [200], DeLinker [201], 3DLinker [28]
	Chemical reaction	Generative	OA-ReactDiff [202], TSNet [203]
Mol + Protein	Ligand binding affinity prediction	Non-generative	TargetDiff [29], MaSIF [204], GET [205], ProtNet [206], HGIN [207], BindNet [208], BADGER [209], DeepTernary [210]
	Protein-ligand docking	Mixed	EquiBind [211], DiffDock [16], TankBind [212], DESERT [213], FABind [214], Re-Dock [215]
	Pocket-based mol sampling	Mixed	Pocket2Mol [216], TargetDiff [29], DiffBP [217], SBDD [218], GraphBP [219], FLAG [220], DESERT [213], D3FG [221], MolCRAFT [152], MolJO [222], DiffBP [217], VoxBind [223]
Protein + Protein	Protein interface prediction	Non-generative	DeepInteract [224], dMaSIF [225], SASNet [226]
	Binding affinity prediction	Non-generative	mmCSM-PPI [227], GeoPPI [228], GET [205]
	Protein-protein docking	Mixed	EquiDock [229], HMR [230], HSRN [231], DiffDock-PP [232], SyNDock [233], AlphaFold-Multimer [234], dMaSIF [235], ElliDock [236], EBMDock [237]
	Antibody design	Mixed	DiffAb [238], MEAN [32], dyMEAN [17], RefineGNN [239], PROTSEED [240], AbBERT[240], ADesigner [241], AbODE[242], AbDiffuser [243], tFold[244], GeoAB [245], RAAD [246], EquiLLM [75]
	Peptide design	Mixed	HelixGAN [247], RFDiffusion [13], PepGLAD [35], PPFlow [248]
		Other domains	
Others	Crystal property prediction	Non-generative	CGCNN [249], MEGNet [250], ALIGNN [251], ECN [252], Matformer [253], Crystal Twins [254], MMPT [255], CrysDiff [256]
	Crystal generation	Generative	CDVAE [257], SyMat [49], DiffCSP [50], DiffCSP++ [258], MatterGen [259], PXRDGen [260], EquiCSP [261], FlowMM [262], CrysBFN [263]
	RNA structure ranking	Non-generative	ARES [35], PaxNet [264], EquiRNA [265]

properties. Similar to BoxBath, Water-3D created by [26] randomly initializes the water states and constructs a high-resolution water scenario. Beyond the simulation of particle-level interaction in previous datasets, Kubric [266] and MIT Pushing [268] can be utilized to evaluate face interactions. Physion [267] is a large-scale dataset that involves more realistic and diverse objects driven by more complex physical interactions, including gravity, friction, elasticity, and other factors.

Methods: Plenty of studies have been devoted to learning to simulate complex physical systems using GNNs, including Interaction Network [112], NRI [27], HRN [119], DPI-Net [118], HOGN [113], GNS [26], C-GNS [116], HGNS [117], GNS* [115], and FIGNet [120]. However, all these methods adopt typical GNNs that are unaware of full symmetry in 3D world, and only a subset of them considers translation-equivariance. Since the work of SE(3)-Transformer [8], roto-translation equivariance is introduced upon the attention-based

Table 4 Summary of typical datasets and benchmarks for the single instance applications

Dataset	Number of samples	Task	Benchmark
Particle			
N-Body [27]	70K	N-body Simulation	NRI [27]
3D N-Body [5]	7K	N-body Simulation	EGNN [5]
Constrained N-Body [51]	5.5K	N-body Simulation	GMN [51]
Hierarchical N-Body [5]	9K	N-body Simulation	EGHN [5]
Water3D [26]	0.8K	Scene Simulation	GNS [26]
Kubric MOVi-A [266]	0.02K	Scene Simulation	GNS* [115]
Physion [267]	16K	Scene Simulation	SGNN [25]
MIT Pushing [268]	6K	Scene Simulation	FIGNet [120]
FluidFall [118]	3K	Scene Simulation	DPI-Net [118]
FluidShake [118]	2K	Scene Simulation	DPI-Net [118]
BoxBath [118]	3K	Scene Simulation	DPI-Net [118]
RiceGrip [118]	5K	Scene Simulation	DPI-Net [118]
Small molecule			
QM9 [21]	134K	Molecule Property Prediction	ATOM3D [269]
		Molecule Generation	GEOM-QM9 [270]
		Molecule Pretraining	3D-Infomax [158]
MD17 [271]	3.6M	Molecule Property Prediction	SchNet [47]
		Molecule Dynamics	GMN [51]
OCP [272]	9.8M	Molecule Property Prediction	eSCN [79]
		Molecular Dynamics	GemNet [4]
Adk [273]	4.1K	Molecular Dynamics	EGHN [5]
DW-4 [126]	10K	Molecular Dynamics	EQ-Flow [126]
LJ-13 [126]	10K	Molecular Dynamics	EQ-Flow [126]
Fast-folding proteins [274]	5M	Molecular Dynamics	ITO [128]
GEOM [275]	450K	Molecule Property Prediction	SchNet [47]
		Molecule Generation	GEOM-Drugs [270]
		Molecule Pretraining	GMN [51]
PCQM4Mv2 [107]	3.3M	Molecule Pretraining	3D PGT [276]
QMugs [277]	665K	Molecule Pretraining	3D-Infomax [158]
Uni-Mol [159]	209M	Molecule Pretraining	Uni-Mol [159]
Protein			
GENE Ontology [278]	33.5K	Protein Property Prediction	GearNet [168]
ENZYME [279]	18.5K	Protein Property Prediction	GearNet [168]
CATH [280]	189K	Protein Inverse Folding	GVP-GNN [166]
		Protein Pretraining	S2F [281]
		Protein Co-Design	PROTSEED [188]
SCOPe [282]	108K	Protein Inverse Folding	ProstT5 [283]
		Protein Pretraining	ProSE [284]
		Protein Property Prediction	TAPE [285]
AlphaFoldDB [286]	200M	Protein Folding	ESMFold [186]
		Protein Inverse Folding	AlphaDesign [287]
		Protein Pretraining	GearNet [168]
UniProt [288]	216M	Protein Pretraining	Prottrans [190]
		Protein Property Prediction	DeepLoc [289]
BFD [290]	2100M	Protein Pretraining	Prottrans [190]
NetSurfP-2.0 [291]	11.3K	Protein Pretraining	PEER [292]
CASP [293]	45.7K	Protein Structure Ranking	ATOM3D [269]
PDB [294]	1.2M	Protein Residue Identity	ATOM3D [269]
		Protein Folding	ESMFold [186]

geometric GNNs to address the N-body problem. Later, EGNN [5] proposes a more effective E(n)-equivariant GNN by using the scalarization-based strategy as already detailed in Section 4.3.1. In contrast to EGNN, SEGNN [9] proposes a general SE(3)-equivariant message passing by making use of high-order degree representations. Recently, GMN [51] have developed multi-channel equivariant modeling specifically for constrained N-body systems consisting of sticks or hinges.

Upon GMN, EGHN [65] designs equivariant pooling and equivariant unpooling to handle the complex system with a hierarchical structure. In the meantime, SGNN [25] generalizes and relaxes the symmetry from equivariance to sub-equivariance, which plausibly grants it the capability to excel in scenarios influenced by other factors like gravity. As conventional approaches utilize a fixed velocity estimation throughout the time interval, NCGNN [114] instead estimates

Table 5   The summary of typical datasets and benchmarks for the multi-instance applications

Dataset	Number of samples	Task	Benchmark
		Mol + Mol	
ZINC [295]	727K	Linker Design	3DLinker [28]
CASF [296]	0.28K	Linker Design	DeLinker [201]
GEOM [275]	450K	Linker Design	DiffLinker [200]
SN2-TS [203]	0.11K	Chemical Reaction	TSNet [203]
Transition1x [297]	9.6M	Chemical Reaction	OA-ReactDiff [202]
		Mol + Protein	
CrossDocked 2020 [298]	22.5M	Ligand Affinity	GNINA [298]
		Pocket-Based Molecule Sampling	TargetDiff [29]
PDBBind [22]	23.5K	Ligand Affinity	ATOM3D [269]
		Protein-Ligand Docking	EquiBind [211]
		Protein + Protein	
DIPS [226]	42.8K	Protein Interface Prediction	ATOM3D [269]
		Protein-Protein Docking	EquiDock [229]
DIPS-plus [299]	42.1K	Protein Interface Prediction	DeepInteract [224]
Biogrid [300]	1.7M	Protein Interface Prediction	SYNTERACT [301]
		Protein Interface Prediction	ATOM3D-PIP [269]
DB5.5 [302]	0.23K	Protein-Protein Docking	EquiDock [229]
		Binding Affinity Prediction	GET [205]
PDBBind [22]	23.5K	Binding Affinity Prediction	GeoPPI [228]
SAbDab [23]	8.1K	Antibody Design	RefineGNN [239]
RAbD [20]	0.06K	Antibody Design	RefineGNN [239]
SKEMPI 2.0 [303]	7.1K	Antibody Design	ATOM3D [269]
		Binding Affinity Prediction	mmCSM-PPI [227]
Cov-abdab [304]	2.4K	Antibody Design	RefineGNN [239]
PepBDB [305]	13K	Peptide Design	CAMP [306]
LNR [307]	0.09K	Peptide Design	PDAR [307]
PepGLAD [35]	6K	Peptide Design	PepGLAD [35]
PPFlow [248]	13K	Peptide Design	PPBench2024 [248]
		Others	
Materials Project [308]	154K	Protein Crys. Property Prediction	CGCNN [249]
		Crystal Generation	CDVAE [257]
Perov-5 [309,310]	18.9K	Crys. Generation	CDVAE [257]
Carbon-24 [311]	10.1K	Crys. Generation	CDVAE [257]
ARVIS-DFT [312]	41K	Crys. Property Prediction	JARVIS-ML [313]
FARFAR2-Puzzles [314]	18K	RNA Struct. Ranking	ARES [269]
rRNAsolo [265]	92K	RNA Struct. Ranking	EquiRNA [265]

velocities at multiple time points using Newton-Cotes numerical integration. There are also other works that approach physical simulation based on the spatio-temporal setting. LoCS [61] utilizes GRU to record the memory of past frames and additionally incorporates rotation-invariance to improve the model's generalization ability; EqMotion [121] distills the history trajectories of each node into a multi-dimension vector and then designs an equivariant module and an interaction reasoning module to predict future frames; ESTAG [31] employs equivariant discrete Fourier Transform along with the equivariant spatio-temporal attention mechanism to model the physical dynamics. SEGNO [315] incorporates the second-order graph neural ODE with equivariant property to reduce the roll-out error of long-term physical simulation.

5.2   Tasks on small molecules

By representing atom coordinates as node positions and bonds as edges, a molecule naturally becomes a geometric graph where $\vec{X} \in \mathbb{R}^{N \times 3}$ represents the positions of N atoms in the molecular, $H \in \mathbb{R}^{N \times C_h}$ indicates the atom types or other properties of the atoms, and $A \in \{0, 1\}^{N \times N}$ represents the existence of bonds. Usually, the edge feature $e_{ij} \in \{0, 1, 2, 3\}$ is defined by the bond type of the edge from node i to j. In addition to chemical edges, the relative distance d_{ij} between two atoms is also utilized for constructing k-NN spatial edges by selecting for each atom the k nearest atoms as its neighbors, and the spatial edge feature is defined as $e_{ij} = \sigma(d_{ij})$ where σ is a non-linear function, such as RBF.

Prior to the use of geometric graph, a molecule could be typically represented by a 1D string (e.g., SMILES [316] and SMARTS [317]) or a 2D topological graph, both of which lose sight of the geometric information of the molecule, resulting in defective performance for the tasks that involve crucial spatial interactions between atoms. Here, we only introduce the works that apply geometric graphs to represent molecules.

5.2.1   Molecular property prediction

Molecular property prediction has been a fundamental task in computational biochemistry and machine learning. As pinpointed by MoleculeNet [45], common properties can be

subdivided into four categories: quantum mechanics, physical chemistry, biophysics, and physiology. With the help of geometric GNNs, we are now able to additionally consider the molecular geometries which have been demonstrated to be crucial in determining the quantum chemistry properties of molecules.

Task definition: With the input molecule characterized as a geometric graph $\vec{\mathcal{G}}$, the task is to learn a model ϕ_θ to predict a scalar property y and/or a vectorial property $\vec{y}$:

$$y, \vec{y} = \phi_\theta(\vec{\mathcal{G}}). \tag{56}$$

While most works mainly focus on the single-task setting by predicting each individual type of property independently, it is also possible to leverage the multi-task setting by predicting multiple types of property simultaneously.

Symmetry preserved: It is an SE(3)-invariant task in terms of y since it remains unaffected by any rotation or translation exerted on the molecule, i.e., $\phi_\theta(\vec{\mathcal{G}}) = \phi_\theta(g \cdot \vec{\mathcal{G}}), \forall g \in \mathrm{SE}(3)$. As for $\vec{y}$, we enforce SE(3)-equivariance into the model: $g \cdot \phi_\theta(\vec{\mathcal{G}}) = \phi_\theta(g \cdot \vec{\mathcal{G}}), \forall g \in \mathrm{SE}(3)$.

Datasets: There are currently three popular data sources for the evaluation of this task, including QM9 [21], MD17 [271], and Open Catalyst Project (OCP) [272]. The QM9 dataset contains 131K small organic molecules with up to nine heavy atoms from CONF, and each molecular is annotated with 13 property labels ranging from the highest occupied molecular orbital to the norm of the dipole moment. MD17 is a collection of molecular dynamic simulations for eight small organic molecules, whose goal is to predict both the energy and atomic forces of each molecule, given the atom coordinates in the non-equilibrium and slightly moving system. OCP consists of more than 100M atomic structures for catalysts to help address climate change, each composed of a molecule called adsorbate placed on a slab named catalyst. OCP provides two datasets OC20 [34] and OC22 [272] for benchmarking, and there are three kinds of tasks in OCP where Initial Structure to Relaxed Energy (IS2RE) taking an initial structure as input to predict the relaxed energy is a highly challenging task.

Methods: Most of the methods introduced in Section 4 are evaluated on molecular property prediction tasks. Here, to avoid redundant introduction, we no longer describe each method in detail and only specify which of the three mentioned benchmarks they are evaluated on. Specifically, invariant GNNs (including SchNet [47], DimeNet [3], SphereNet [123], and GemNet [4]), equivariant GNNs (including Cormorant [76] and PaiNN [6]) and equivariant graph transformers (e.g., TorchMD-Net [83] and Equiformer [11]) employ both QM9 and MD17 for performance comparisons. Other methods like NequIP [10] are conducted on MD17, while EGNN [5], LieConv [54] and SE(3)-Transformer [8] are evaluated on QM9. SEGNN [9], Graphormer [81,82], Equiformer [11], SCN [78], and eSCN [79] leverage more challenging benchmarks, namely, OC20 and even OC22 for performance assessment, revealing encouraging effectiveness of applying geometric GNNs to catalyst design.

5.2.2 Molecular dynamics simulation

Molecular Dynamics (MD) simulation aims to simulate the temporal evolution process of molecules driven by internal interactions between atoms within the same molecule, external interactions among different molecules, or environmental interactions from solvents and force fields.

Task definition: Given an input molecular graph at time t, i.e., $\vec{\mathcal{G}}^{(t)}$, this task simulates the dynamical evolution of the molecular over some time. In general, the future coordinates $\vec{X}^{(t+k)}(k > 0)$ are estimated by

$$\vec{X}^{(t+k)} = \phi_\theta(\vec{\mathcal{G}}^{(t)}). \tag{57}$$

Similar to general physical dynamics simulation in Section 5.1.1, one may also conduct a roll-out prediction setting or the spatio-temporal input setting. Besides, in contrast to the direct trajectory prediction here, MD can be alternatively addressed with the methods designed for molecular property prediction as described in the last subsection. We can first predict the node-level force $\vec{F} \in \mathbb{R}^{N \times 3}$ or the graph-level system energy $E \in \mathbb{R}$ for the given state of the system $\vec{\mathcal{G}}$, and then use these estimated quantities to update the molecular dynamics by solving the differential equations that describe molecular dynamics.

Symmetry preserved: Clearly, the output coordinate matrix $\vec{X}^{(t+k)}$ is E(3)-equivariant.

Datasets: MD17 [271], AdK [273], OCP [272], DW-4 [126], fast-folding proteins [274], and LJ-13 [126] are available datasets for MD simulation in the machine learning community. MD-17 [271] which is usually used for molecular property prediction also contains the trajectories of eight molecules generated via DFT. The AdK equilibrium trajectory dataset simulated by CHARMM27 force field in the MDAnalysis software [318] involves the MD trajectory of apo adenylate kinase with explicit water and ions in NPT at 300 K and 1 bar, where the atom positions of the protein are saved every 240 ps for a total of 1.004 μs. Besides the common relaxed energy prediction task, OCP releases a dataset split for MD, which computes short, high-temperature ab initio MD trajectories on a randomly sampled subset of the relaxed states. DW-4 is a relatively simple system consisting of only 4 particles embedded in a 2D space which are governed by an energy function between pairs of particles, while LJ-13 is given by the *Leonnard-Jones* potential, consisting of 13 particles embedded in a 3D space. Both energy functions in DW-4 and LJ-13 satisfy E(3)-equivariance. The fast-folding proteins dataset includes 12 structurally diverse proteins, such as Chignolin, Trp-Cage, and BBA. The simulations were conducted in explicit solvent, with frame spacing ranging from 100 μs to 1 ms.

Methods: As a multi-channel version of EGNN [5], GMN [51] focuses specifically on the physical dynamics by considering the geometric constraints (such as chemical bonds) between atoms and achieves promising results on the MD simulation task in MD17. EGHN [65] develops an equivariant version of UNet [319] equipped with equivariant pooling/unpooling layers to better reveal the hierarchy of large molecules such as proteins, leading to state-of-the-art

performance on AdK dataset. NequIP [127] learns interatomic potentials and forces using high-order geometric tensors and E(3)-equivariant convolution layers, achieving high data efficiency and quantum chemical level accuracy for MD17. By observing that GMN and other related geometric GNN methods only learn constant integration of the velocity, Newton–Cotes GNN [114] predicts the integration based on several velocity estimations with Newton–Cotes formulas and proves its effectiveness theoretically and empirically. ESTAG [31] reformulates dynamics simulation as a spatio-temporal prediction task by employing the trajectory in the past period to recover the Non-Markovian interactions. EGNO [127] models the MD trajectory as a function over time using neural operators. SEGNO [315] leverages the second-order continuity information to enhance the performance of GeoTDM [130] further leverages the diffusion model to perform trajectory generation on molecular dynamics.

Considering the uncertainty of molecular dynamics at the quantum scale, some methods aim to fit the equilibrium distribution of molecules rather than predicting a single molecular conformation. By leveraging the continuous normalizing flows, E-CNF [126] predicts SE(3)-equivariant molecular conformers through the invariant CoM prior density and equivariant vector fields, showing better generation capabilities compared to invariant flows. Later, E-ACF [129] employs the augmented normalizing flow [320] to learn the target distribution of molecules from MD trajectories, which retains SE(3)-equivariance by projecting the atomic Cartesian coordinates into the SE(3)-invariant vector space. Furthermore, ITO [128] utilizes the score matching diffusion model for stochastic dynamics across multiple time-scales, with extended SE(3)-equivariant PaiNN architecture [321], showcasing considerable generalization ability for different molecular scales.

5.2.3 Molecular generation

Molecule generation plays a central role in drug discovery and material design. Its goal is to generate novel molecules with properties of interest by using machine learning.

Task definition: Basically, the methods for molecular generation learn a parametric probability distribution $p_\theta(\vec{\mathcal{G}})$ from an observed dataset $\mathbb{D}:=\{\vec{\mathcal{G}}_i\}$. A novel molecular geometric graph is then sampled from the learned distribution:

$$\vec{\mathcal{G}} \sim p_\theta(\vec{\mathcal{G}}). \tag{58}$$

Instead of generating a whole geometric graph (namely de novo generation), there are part of methods investigating the conditional generalization paradigm by generating the 3D coordinates $\vec{X}$ given the 2D topological graph $\mathcal{G}(\boldsymbol{H}, \boldsymbol{A})$, forming the so-called conformation generation problem $\vec{X} \sim p_\theta(\vec{X} \mid \boldsymbol{H}, \boldsymbol{A})$.

Symmetry preserved: The generative model $p_\theta(\vec{\mathcal{G}})$ should be E(3)-invariant, i.e., $p_\theta(g \cdot \vec{\mathcal{G}}) = p_\theta(\vec{\mathcal{G}}), \forall g \in \mathrm{E}(3)$. This is to ensure that the probability distribution is unaffected by the specific choice of the coordinate system to describe a molecule. In some methods as presented latter, $p_\theta(\vec{\mathcal{G}})$ is marginalized from a joint distribution $p_\theta(\vec{\mathcal{G}}, \vec{\mathcal{G}}^{(0)}) =$ $p_\theta(\vec{\mathcal{G}} \mid \vec{\mathcal{G}}^{(0)}) p(\vec{\mathcal{G}}^{(0)})$ where $p(\vec{\mathcal{G}}^{(0)})$ denotes a certain initial distribution. In this scenario, the initial distribution $p(\vec{\mathcal{G}}^{(0)})$ should be E(3)-invariant and the likelihood distribution $p_\theta(\vec{\mathcal{G}} \mid \vec{\mathcal{G}}^{(0)})$ should be E(3)-equivariant, to guarantee the E(3)-invariance of $p_\theta(\vec{\mathcal{G}})$ [143].

Datasets: QM9 [21] and GEOM [275] are two prevailing datasets used for molecular generation. In particular, QM9 consisting of about 134K organic molecules contains the molecular 3D structures (e.g., the coordinates of each atom in 3D space) and a wide range of chemical properties for each molecule. GEOM is a comprehensive dataset containing over 37 million molecular conformations, offering diverse conformation ensembles for each 2D molecular structure.

Methods: Current methods can be divided into two classes, namely, conformation generation and *de novo* generation. Conformation generation is to generate 3D conformation given the 2D graph representation. Traditional methods [321] focus on the two-stage strategy: first predicting distances and then reconstructing coordinates, which yet could lead to unrealistic structures if the predicted distances are invalid. To avoid this issue, ConfVAE [135] reformulates the generation task as a bilevel optimization problem under the framework of VAE [322], where the distance prediction and conformation generation are optimized jointly in an end-to-end manner. At the same time, ConfGF [136] estimates the gradient fields of inter-atomic distances by using denoising score matching, and then samples the conformations via annealed Langevin dynamics. Later, DGSM [141] further extends ConfGF by modeling long-range interactions between non-bond atoms additionally. Instead of optimizing force field expensively, GeoMol [144] predicts the local 3D geometries including bond distances and torsion angles simultaneously in an SE(3)-invariant way. Without predicting intermediate values like inter-atomic distances, DMCG [147] generates the 3D atomic coordinates by iteratively refining the initial coordinate predictions while accounting for invariance through its designed loss function. Due to the success of diffusion models, GeoDiff [133] leverages graph field network to learn SE(3)-invariant distribution, and Torsional Diffusion [30] operates in torsion angle space rather than in Euclidean space.

As for *de novo* generation, a series methods have been proposed thanks to the fruitful progress of generative models [323]. Built upon Schnet [47], G-SchNet [137] introduces an autoregressive model to directly generate 3D molecular structures, while maintaining physical constraints. cG-SchNet [138] further extends G-SchNet to property-guided generation. Leveraging the generative capabilities of flow models, E-NFs [142] reformulates generation as the task of solving a continuous-time ODE, where the dynamics are predicted by EGNN [5]. By harnessing the power of diffusion, EDM [143] exploits E(3) equivariance by employing EGNN [5] to enhance the diffusion process across both continuous and discrete features. GeoLDM [134] further maps the geometric features into the latent space where latent diffusion is performed. Rooted in EDM, EEGSDE [146] formulates the generation process as an equivariant SDE and employs a meticulously designed energy function to guide the

generation. Recently, MDM [139] takes into account inter-atomic forces at varying distances (e.g., van der Waals forces), and injects variational noises to enhance performance for large molecules and improve generation diversity. To address atom-bond inconsistency problem, MolDiff [140] introduces a joint atom-bond diffusion framework and bond guidance to make sure atoms are better suited for bonding. HierDiff [148] adopts a hierarchical diffusion which first generates the coarse positions of molecular fragments and then fills in the fine-grained atomic geometry. EQUIFM [149] further explores *de novo* generation with flow matching, utilizing different probability paths for atom type and structure generation.

5.2.4 Molecular pretraining

Given that molecular labeling is expensive to obtain, pretraining molecular representation models without labels becomes fundamental and indispensable in real applications. These pretrained models can then be directly transferred or fine-tuned for specific downstream tasks, such as predicting binding affinity and molecular stability, thereby alleviating data scarcity and improving training efficiency. Previous research primarily focused on pretraining models utilizing non-geometric information, including SMILES notations [324], chemical graphs [325], functional groups [326], etc. Recently, there has been a growing interest in self-supervised pretraining on the 3D geometric structure of molecules.

Task definition: Suppose $\phi_\theta(\vec{\mathcal{G}})$ to be the representation model, and $\mathcal{L}\big(\hat{\boldsymbol{y}}(\vec{\mathcal{G}}), \phi_\theta(\vec{\mathcal{G}})\big)$ to be the self-supervised training objective where $\hat{\boldsymbol{y}}(\vec{\mathcal{G}})$ denotes the pseudo label created based on the structure of $\vec{\mathcal{G}}$. The representation model is optimized to minimize the self-supervised objective as

$$\theta = \arg\min_\theta \mathcal{L}\big(\hat{\boldsymbol{y}}(\vec{\mathcal{G}}), \phi_\theta(\vec{\mathcal{G}})\big). \tag{59}$$

Symmetry preserved: The representation model $\phi_\theta(\vec{\mathcal{G}})$ is E(3)-equivariant if $\hat{\boldsymbol{y}}(\vec{\mathcal{G}})$ is a steerable vector, and is E(3)-invariant if $\hat{\boldsymbol{y}}(\vec{\mathcal{G}})$ consists of scalars.

Datasets: PCQM4Mv2 [327] is a comprehensive quantum chemistry dataset consisting of 3.37 million molecules derived from the OGB benchmark, which was originally curated as part of the PubChemQC project [328]. QM9 [21] is another popular dataset that encompasses quantum chemistry structures and properties, featuring 134K molecules. QMugs[277] expands QM9 by offering a more extensive collection of drug-like molecules, totaling 665K molecules. GEOM [275] is an energy-annotated molecular conformation dataset containing 37 million molecular conformations sourced from multiple datasets, such as QM9 and CREST program [329]. Uni-Mol [159] constructs a conformation dataset containing 19 million molecules. It utilizes ETKGD with Merck Molecular Force Field optimization in RDKit to generate 11 random conformations for each molecule, resulting in a total of 209 million conformations.

Methods: A variety of studies investigate the denoising objective, pretraining the model by recovering the original signal from a perturbed input. Specifically, GeoSSL-DDM [154] formulates the denoising objective based on atomic distance. Uni-Mol [159] proposes position denoising and joint training between 3D molecular conformations and candidate protein binding pockets. GNS-TAT [156] establishes a connection between coordinate denoising and the potential energy of molecular conformations. MGMAE [157] proposes a reconstruction strategy to train on the heterogeneous atom-bond graph with a high mask ratio. 3D-EMGP [153] further proposes to predict the atomic pseudo force field which is estimated by an Riemann-Gaussian denoising distribution to ensure E(3)-invariant pretraining loss. Apart from the denoising objective, GraphMVP [155] leverages the correlation between 2D molecular graphs and 3D conformations, constructing a contrastive objective for the model pretraining. Similar to GraphMVP, Transformer-M [160] leverages positional encodings and attention biases to encode the 2D and 3D structures in one Transformer model. Meanwhile, 3D-Infomax [158] exploits this correspondence by attempting to maximize the mutual information between 2D molecular graph embeddings and learned representations of the corresponding 3D graphs. MoleculeSDE [161] extends 3D-Infomax [158] and leverages group symmetric stochastic differential equation models to establish a connection between 3D geometries and 2D topologies, with a tighter MI bound. Frad [163] decomposes molecules into fragments to fix the rigid parts and pretrains the model via denoising on the flexible parts. SliDe [162] explores pretraining with denoising from a distribution that encodes physical principles. DenoiseVAE [164] utilizes a learnable noise generation strategy to adaptively acquire atom-specific noise distributions for different molecules, which results in more accurate force field learning.

5.3 Tasks on proteins

Proteins are large biomolecules that are composed of one or more long chains of amino acid residues. All proteinogenic amino acids share common structural features, including an α-carbon to which an amino group, a carboxyl group, and a variable side chain are bonded. Most proteins fold into unique 3D structures that determine the function and activity of proteins in biological processes. Owing to the hierarchical structures of proteins, there are mainly two different ways to leverage geometric graph $\vec{\mathcal{G}}$ to represent proteins. For one thing, we can treat each residue as a node, the positions of α-carbons as the coordinate matrix $\vec{\boldsymbol{X}}$ and the residue-level features as $\boldsymbol{H}$. For another thing, we can apply the full-atom setting by considering each atom as a node, the positions of all atoms as $\vec{\boldsymbol{X}}$ and atom-level features as $\boldsymbol{H}$. In both ways, the edges can be created via either the chemical bonds or cut-off distances. There are plenty of works that develop machine learning methods to process proteins. While some of them focus on 1D residue sequences, this survey is mainly interested in the study of 3D structures and will demonstrate several relevant tasks in the following.

5.3.1 Protein property prediction

Similar to molecular property prediction, protein property prediction is a crucial E(3)-invariant task in computational biology. Most previous works solely employ residue

sequences to predict protein properties. Thanks to the development of geometric structure modeling, more and more attentions are paid to using geometric GNNs to estimate the functional property of proteins via exploring 3D structures. In terms of the prediction granularity, the task of protein property prediction is classified as protein-level, residue-level and atom-level prediction, with the details provided below.

Protein-level prediction: Many tasks aim to predict the functions or certain scores given the protein structure. **(1) Enzyme Commission (EC)** number prediction [167] is a prevailing protein-level classification task which aims to predict the catalyzed reaction class of the given enzyme. **(2) Gene Ontology (GO)** term prediction [167] seeks to predict the functional classes concerning gene ontology given the protein structure, whose data is usually split into three tracks: molecular function (MF), biological process (BP), and cellular component (CC). **(3) Protein structure ranking** learns a quality score function of the given protein structure to estimate the structural similarity between the candidate protein and the native structure. It plays a vital role in computational biology, as it assists researchers in pinpointing the most accurate or biologically significant protein conformations from a collection of potential structures. **(4) Protein localization prediction** targets at forecasting the subcellular locations of proteins [289], which is essential to understand the function of a protein and helps investigate the pathogenesis of many human diseases [330]. **(5) Fitness landscape prediction** primarily focuses on the prediction of the effects of residue mutations on the fitness of proteins. Typical target functions include β-lactamase [292], Adeno-Associated Virus (AAV), Thermostability [331], and Fluorescence and Stability [285].

Abundant protein-level representation models are available in existing literature. DeepFRI [167] and LM-GVP [166] propose a two-stage architecture, which adopts language models to extract amino acid sequence information and graph-based model to learn the interactions between amino acids simultaneously. Notably, LM-GVP utilizes equivariant model GVP [64] as the graph-based model. GearNet [168] proposes a relational graph convolution layer to better capture the 3D geometry of proteins, and exploits multi-view contrastive pretraining to better utilize unlabeled data. As for structure ranking, TM-Align [170] is a typical but not DL-based method, which is time-consuming. Thanks to the expressive ability of geometric GNN, [64,172,173] adopt equivariant GNN models such as GVP [64] and TFN [7] to fulfill model quality assessment (MQA). In addition, TFN [7] is also used for ranking protein-protein complex in PAUL [171].

Residue-level prediction: Atom3D [269] proposes Residue Identity (RES) prediction, which aims to predict the amino acid types at the center of a given local context. The performance on this task measures whether a model can capture the structural dependencies between individual amino acids, which is vital for protein engineering.

Atom-level prediction: The main form of atom-level prediction lies in pocket detection, which requires predicting whether an atom on the protein belongs to the binding site in terms of a potential ligand. Previous methods usually design

algorithms to find and rank the cavities on the protein surface [332,333], or voxelize the protein structure and use 3D-CNN for supervised training [334,335]. Notably, a series of works are exploiting the geometric GNNs to achieve much better performance (ScanNet [174], EquiPocket [175], and PocketMiner [176]).

5.3.2 Protein generation

In terms of what to generate, the approaches for protein generation are categorized into protein folding (or protein structure prediction), protein inverse folding, and protein structure and sequence co-design.

Protein folding aims to generate folding structures given the amino acid sequences of the input protein. This task has significant implications in the field of drug design. The folding structure is generated by:

$$\vec{X} \sim p_\theta(\vec{X} \mid s), \tag{60}$$

where $s \in \mathbb{R}^N$ denotes the amino acid sequence based on the coordinates of all residues $\vec{X} \in \mathbb{R}^{N \times 3}$ (note that each row of $\vec{X}$ can include more than one 3D coordinate vector if full-atom coordinates are considered).

Symmetry preserved: This is an equivariant task, implying that $p_\theta(\vec{X} \mid s) = p_\theta(\vec{X}O + \vec{t} \mid s)$ for an arbitrary orthogonal transformation O and translation $\vec{t}$. Notably, some methods generate the distance matrix or other invariant forms of $\vec{X}$, reducing the task into a trivial generation problem without the equivariance constraint.

Methods: The AlphaFold series [33,183] and RoseTTAFold series [12,48] represent the forefront of contemporary techniques in protein folding. They employ a sophisticated multi-track architecture capable of processing multi-sequence alignments (MSA), amino acid pair-wise distance maps, and geometric structures, each with remarkable efficiency. Building upon these advancements, RoseTTAFold2 [48] extends the capabilities of both AlphaFold2 [183] and RoseTTAFold [12] by refining the attention mechanism and enhancing the three-track architecture, resulting in notable performance improvements. Moreover, RFAA [184] further extends RoseTTAFold's versatility to encompass the design of various biomolecules beyond proteins, including nucleic acids, small molecules, and metals. In contrast, ESMFold [336] and HelixFold-Single [187] represent a departure from traditional methods by eschewing the requirement for MSA. Instead, it learns to predict protein structures directly from primary sequence data, significantly enhancing inference efficiency. Additionally, EigenFold [185] introduces a novel harmonic diffusion process that projects protein structures onto eigenmodes, thereby preventing the disassembly of adjacent nodes.

Protein inverse folding aims to generate amino acid sequences conditional on the folding structures of the input protein. Using the same denotations as the task of protein folding, the model p_θ generates the amino acid sequence $s \in \mathbb{R}^N$ of interest:

$$s \sim p_\theta(s \mid \vec{X}). \tag{61}$$

Symmetry preserved: This is an invariant task, indicating

that $p_\theta(\boldsymbol{s} \mid \vec{\boldsymbol{X}}) = p_\theta(\boldsymbol{s} \mid \vec{\boldsymbol{X}}\boldsymbol{O} + \vec{t})$ for an arbitrary orthogonal transformation $\boldsymbol{O}$ and translation $\bar{t}$.

Methods: Typical methods such as [177] and [178] take the invariant features including distance and dihedral angles as input, to ensure invariance during generation. More recently, based on GVP [64] that is E(3)-equivariant, ESM-IF [85] further incorporates more structure information for the generation, while keeping the output sequence invariant. Similarly, LM-Design [181] integrates structural embedding into language models to improve the performance of inverse folding. ProteinMPNN [179] uses an invariant architecture to embed its backbone and predicts amino acid probabilities autoregressively while enforcing desired constraints. PiFold [180] additionally incorporate distance, angle, and direction features and proposes PiGNN to non-autoregressively generate the sequences. KW-Design [182] integrates knowledge from pretrained sequence and structure models to refine the sequences generated by the baselines with a memory retrieval mechanism.

Protein structure and sequence co-design aims to generate both the amino acid sequences and folding structures, which is formally derived as:

$$\vec{\boldsymbol{X}}, \boldsymbol{s} \sim p_\theta(\vec{\boldsymbol{X}}, \boldsymbol{s}). \tag{62}$$

Symmetry preserved: Clearly, this task is invariant with respect to $\boldsymbol{s}$, and equivariant with respect to $\vec{\boldsymbol{X}}$.

Methods: Based on RoseTTAFold [12], RFdiffusion [13] incorporates Gaussian noise into coordinates and Brownian motion noise into orientations, subsequently denoises the structure step-by-step and recovers sequence using ProteinMPNN [179]. Meanwhile, Chroma [14] introduces a revolutionary programmable diffusion framework, empowering diverse conditional generation and precise targeting of properties through constraints such as symmetry, shape, and semantics. Both Chroma and RFDiffusion begin with structure generation and then conduct the subsequent sampling of the corresponding sequence through another module. Unlike these two works, PROTSEED [188] designs the structure and sequence jointly by an encoder-decoder framework, where the encoder is trigonometry-aware to learn context features and the decoder is SE(3)-equivariant to express the sequence and structure.

Datasets: ATOM3D [269] constructs multiple widely-used datasets tailored for protein design tasks. CASP [293] stands out as a renowned contest dedicated to protein structure prediction. In this competition, participants submit predicted structures for evaluation, particularly when the experimental structures are not publicly available. The community then assesses the quality of these submissions. Additionally, AlphaFoldDB [286], SCOPe [282], and CATH [280] serve as valuable resources for protein design, providing datasets comprising protein structures alongside their corresponding sequences. SCOPe and CATH consist of segmented protein structure domains, while AlphaFoldDB boasts a repository of over 200 million complete structures predicted by AlphaFold2 [183]. Moreover, with predictions stemming from ESMFold [336], the ESM Metagenomic Atlas boasts a collection of about 772 million metagenomic protein structures.

5.3.3 Protein pretraining

Similar to molecule pretraining task, protein pretraining also aims to learn representations of protein, which can be used in downstream tasks.

Task definition: Generally, each input protein is modeled as a geometric graph $\vec{\mathcal{G}}$ and the pretraining purpose is to learn a parametric model ϕ_θ which can output high-quality representations $\boldsymbol{H} \in \mathbb{R}^{N \times d}$ of the input protein:

$$\boldsymbol{H} = \phi_\theta(\vec{\mathcal{G}}). \tag{63}$$

Symmetry preserved: It is equivariant for the output vectors in $\boldsymbol{H}$, and invariant for the output scalars in $\boldsymbol{H}$.

Datasets: For protein sequence pretraining methods, UniProt [288] functions as a central repository for both protein sequence and functional information. It is organized into clusters by UniRef [337], with pairwise sequence identity thresholds typically set at 50% and 100% (referred to as UniRef50 and UniRef100) to eliminate redundancy. BFD [290], on the other hand, represents a larger sequence dataset, formed by amalgamating UniProt with protein sequences sourced from metagenomic sequencing projects. Furthermore, NetSurfP-2.0 [291] furnishes labels for protein secondary structure prediction, delineated into 3-states and 8-states, offering valuable resources for supervised training. In the realm of protein structure pretraining and classification, SCOPe [282], CATH [280], and AlphaFoldDB [286] hold significant importance. They provide comprehensive repositories for protein structures, facilitating research and advancement in the field.

Methods: Previous protein pretraining methods such as ESM-1b [338], ESM2 [336], ProtTrans [190], xTrimoPGLM [191], and ProtGPT2 [192], are based on sequence masking and prediction, inspired by the success of NLP language models. Readers can refer to the survey by [339] for more introductions of protein language models. Recent attentions have been paid to pretrained models based on the 3D structure information. For instance, GearNet [168] built upon an invariant GNN with multi-type message passing leverages several pretraining objectives including contrastive learning between sequences and structures, distance/dihedral prediction, and residue type prediction. Other works like ProFSA [194] and DrugCLIP [196] also utilize contrastive learning to learn SE(3)-invariant features, but focusing more on pocket pretraining, where the pocket-ligand interaction knowledge is incorporated as well. Guo et al. [198] employs pretraining with the protein's tertiary structure, incorporating SE(3)-invariant features to ensure the efficient preservation of SE(3)-equivariance. PAAG [199] enables multi-level alignment between protein sequence and textual annotation to capture the fine-grained motif inside the protein and successfully designs proteins with functional domains.

5.4 Tasks on Mol+Mol

This subsection introduces the tasks with the input of "molecule+molecule", including liker design and chemical reaction prediction.

5.4.1 Linker design

Fragment-based molecule design requires to predict the linker,

a small molecule, so that two or more molecular components can be combined into novel molecules with desirable properties. Linkers are of great importance in maintaining the proper orientation, flexibility, and stability of multi-domain proteins or fusion proteins.

Task definition: The input consists of two or more unlinked molecular fragments, which are all represented as geometric graphs $\{\vec{\mathcal{G}}_i\}_{i=1}^k$, and the model needs to learn an equivariant function f_θ whose output is a small molecule $\vec{\mathcal{G}}_L$ used to link the fragments. Specifically,

$$\vec{X}_L, H_L = f_\theta(\vec{\mathcal{G}}_1, \vec{\mathcal{G}}_2, \dots, \vec{\mathcal{G}}_k). \tag{64}$$

Symmetry preserved: If we impose rotation or translation operations on the input fragments simultaneously, the output coordinates should transform correspondingly while the atom features keep invariant.

Datasets: The linkers connecting molecules in ZINC [295] can be computationally synthesized, similar to the methods employed by [340]. Conversely, CASF [296] offers experimentally validated molecules for linker design. In contrast to ZINC and CASF, which typically produce paired fragments, DiffLinker [200] generates a novel dataset comprising three or more fragments, drawing from GEOM [275].

Methods: DeLinker [201] and 3DLinker [28] employ VAE [322] to create the 3D structure of a linker. However, their capability is limited to linking only two fragments, rendering them ineffective when faced with an arbitrary number of fragments to link. In contrast, DiffLinker [200] has recently succeeded in addressing this challenge by harnessing an E(3)-equivariant diffusion model configured to handle multiple fragments.

5.4.2   Chemical reaction prediction

In chemical reactions, identifying and characterizing transition state (TS) structures is crucial for understanding reaction mechanisms. This process entails locating the TS structure that minimizes the system's potential energy (PE) while adhering to specific constraints, such as SE(3) invariance.

Task definition: Given a reactant $\vec{\mathcal{G}}_R$ and a product $\vec{\mathcal{G}}_P$, the objective is to generate the TS structure $\vec{\mathcal{G}}_{TS}$ that optimizes the following objective:

$$\vec{\mathcal{G}}_{TS}^* = \underset{\vec{\mathcal{G}}_{TS}}{\arg\min} \, \mathrm{PE}(\vec{\mathcal{G}}_{TS} | \vec{\mathcal{G}}_R, \vec{\mathcal{G}}_P), \tag{65}$$

where the function PE(·) returns the potential energy.

Symmetry preserved: In general, the output, namely, the TS structure is invariant to any independent transformation (e.g., rotation) imposed to each of the input structure. If the input and output are always fixed within the same 3D coordinate space, then this task is equivariant, namely, imposing the same transformation to the two input structures, the output TS is transformed in the same way.

Datasets: TSNet [203] has meticulously assembled a dataset called S_N2-TS, which contains structures of reactants, transition states (TS), and products pertinent to S_N2 reactions. Transition1x [297] provides a resource of 9.6 million density functional theory (DFT) calculations encompassing forces and

energies for molecular configurations across reaction pathways. This extensive dataset offers valuable information for training models for reaction prediction.

Methods: OA-ReactDiff [202] introduces a diffusion model to generate transition state (TS) structures. This model ensures SE(3)-equivariance of the score function by constructing local frames. Moreover, the equivariant backbone model is adapted to accommodate multiple objects. On the other hand, TSNet [203] employs the equivariant graph neural network (GNN) model TFN [7] to predict TS structures. Initially, TFN is pretrained on extensive chemical data, such as QM9 [21], to learn useful representations. It is then fine-tuned specifically for the task of predicting transition structures.

5.5   Tasks on mol+protein

The "molecule+protein" tasks are well explored, such as ligand binding affinity prediction, protein-ligand docking, and pocket-based molecule sampling.

5.5.1   Ligand binding affinity prediction

The task of predicting ligand binding affinity revolves around estimating the interaction strength between a protein (receptor) and a small molecule (ligand) [205]. Accurate predictions in this area offer significant advantages for designing and refining drug candidates. Additionally, they aid in prioritizing compounds for experimental evaluation, thereby streamlining the drug discovery process.

Task definition: With both the molecule and protein regarded as geometric graph $\vec{\mathcal{G}}_m, \vec{\mathcal{G}}_p$, the task aims to learn an efficient predictor ϕ_θ, which can predict the binding strength s accurately:

$$s = \phi_\theta(\vec{\mathcal{G}}_p, \vec{\mathcal{G}}_m). \tag{66}$$

Symmetry preserved: It is obvious that the binding affinity will not change under any transformation.

Datasets: CrossDocked2020 [298] contains over 22 million posed ligand-receptor complexes and the corresponding binding affinity values, which are generated by docking ligands into multiple receptor structures from the same binding pocket. PDBbind [22] provides accurate and reliable binding affinity data, allowing researchers to assess how well computational methods can predict the strength of binding between proteins and ligands.

Methods: MaSIF [204] utilizes geodesic space to represent the protein surface, assigns geometric and chemical features to patches, and employs rotation invariance to process these features, facilitating predictions of protein-ligand interactions. ProtNet [206] considers 3D protein presentations at various levels (e.g., amino-acid level, backbone level, and all-atoms level) to accomplish affinity prediction tasks. GET [205] extends this concept by unifying different levels universally for both molecule and protein representations. TargetDiff [29] introduces a diffusion process that gradually adds noise to coordinates and atom types. This process, guided by an SE(3)-equivariant graph neural network (GNN), incorporates binding free energy terms to steer generation towards high-affinity poses. HGIN [207] constructs a hierarchical invariant graph model to predict changes in binding affinity resulting from

protein mutations. BindNet [208] designs two pretraining tasks utilizing Uni-Mol [159] as the encoder to jointly learn protein and ligand interactions.

5.5.2 Protein-ligand docking

This task works towards predicting the transformation, e.g., rotation and translation, imposed on protein and molecules so that they can dock together with the minimum root-mean-square-deviation.

Task definition: Without loss of generality, we assume that the protein remains fixed while the position of the molecule transforms. By denoting the protein as $\vec{\mathcal{G}}_p := (\vec{X}_p, H_p)$ and the molecule as $\vec{\mathcal{G}}_m := (\vec{X}_m, H_m)$, respectively, the model needs to learn a prediction function ϕ_θ that outputs the rotation matrix and translation vector (i.e., $R, \vec{t}$) by

$$R, \vec{t} = \phi_\theta(\vec{X}_p, H_p; \vec{X}_m, H_m). \tag{67}$$

With the predicted rotation R and translation $\vec{t}$, we can dock the molecule towards the fixed protein.

Symmetry preserved: To make the final docked complex to be SE(3)-equivariant, the predictor ϕ is supposed to meet the following independent SE(3) constrains [211]:

$$R' = Q_m R Q_p^\top, \vec{t'} = Q_m \vec{t} - Q_m R Q_p^\top \vec{t}_p + \vec{t}_m,$$
$$\forall Q_p, Q_m \in \text{SO}(3), \vec{t}_p, \vec{t}_m \in \mathbb{R}^3, \tag{68}$$

where $R', \vec{t'}$ are the predicted rotation matrix and translation vector after transforming the protein and the molecule, namely, $R', \vec{t'} = \phi_\theta(\vec{X}_p Q_p + \vec{t}_p, H_p; \vec{X}_m Q_m + \vec{t}_m, H_m)$.

Datasets: PDBbind [22] stands out as the predominant dataset for Protein-Protein Docking, housing over 22 million poses resulting from the docking of ligands into their respective receptor structures. Typically, current methods segment the dataset based on chronological order, leveraging this organization for training and evaluation purposes.

Methods: EquiBind [211] and TankBind [212] have tackled the blind binding problem by leveraging equivariant graph neural networks. TankBind additionally introduces trigonometry constraints to enhance compound rationality. To further enhance performance, DiffDock [16] proposes a diffusion process operating across three groups (T(3), SO(3), and SO(2)). In contrast, DESERT [213] offers a unique approach by initially outlining pocket shapes and then generating molecule structures to bind these pockets. This method alleviates the scarcity of experimental binding data and is not reliant on predefined pocket-drug pairs. Recently, FABind [214] designs geometry-aware GNN layers and efficient interaction modules (e.g., interfacial message passing) to unify pocket prediction and the docking stage, which leads to fast and accurate prediction. Further, Re-Dock [215] explores flexible docking by considering the gap between apo and holo conformations of the target protein, which enhances the practical utility.

5.5.3 Pocket-based mol sampling

The technique of pocket-based molecular sampling aims at generating small molecules that have the potential to bind to a particular pocket on a protein or other biomolecular target.

Task definition: This target-aware design resorts to learn a generation model p_θ whose output is a new molecule $\vec{\mathcal{G}}_m$ that can bind to a specific pocket $\vec{\mathcal{G}}_p$:

$$\vec{\mathcal{G}}_m \sim p_\theta(\vec{\mathcal{G}}_m \mid \vec{\mathcal{G}}_p). \tag{69}$$

Symmetry preserved: It is an equivariant problem, implying that $p_\theta(\vec{\mathcal{G}}_m \mid \vec{\mathcal{G}}_p) = p_\theta(g \cdot \vec{\mathcal{G}}_m \mid g \cdot \vec{\mathcal{G}}_p)$ for any transformation g of interest.

Datasets: CrossDocked2020 [298] serves as a substantial resource for sampling molecules based on docking pockets, containing approximately 22.5 million docked protein-ligand pairs.

Methods: Pocket2Mol [216], GraphBP [219], SBDD [218], and FLAG [220] adopt an autoregressive approach to generate molecules conditioned on binding sites, operating at the granularity of atoms or motifs. In contrast, TargetDiff [29] with a series following diffusion-based methods [152,217,220,223,341] diverges from this method by utilizing 3D equivariant diffusion in a non-autoregressive fashion. This approach enables the generation of all atoms simultaneously, resulting in higher efficiency. DESERT [213] further explores to first sketch the shape of the molecule according to the pocket, and then generates a molecule fitting in the shape. D3FG [221] leverages a fragment-based diffusion to enhance the generative performance by decomposing molecules into functional groups and linkers.

5.6 Tasks on protein+protein

The "protein+protein" tasks include protein interface prediction, protein-protein binding affinity prediction, protein-protein docking, antibody design that considers specifically the interaction between antibodies and antigens, and peptide design that aims at generating target-specific peptide.

5.6.1 Protein interface prediction

Biological processes often depend on interactions between biomolecules. This creates a need for predicting protein-protein interfaces, which involves identifying the regions on a protein's surface that are likely to participate in interactions with other proteins.

Task definition: With the protein pair taken as two geometric graphs $\vec{\mathcal{G}}_1, \vec{\mathcal{G}}_2$, this task requires to learn a predictor ϕ_θ that determines if the atoms on the protein belong to the interface. The output are interpreted as the atomic probabilities $p \in \mathbb{R}^{N_1 + N_2}$ of being located on the interface:

$$p = \phi_\theta(\vec{\mathcal{G}}_1, \vec{\mathcal{G}}_2). \tag{70}$$

Symmetry preserved: Once the interaction proteins are selected, the atoms in the interface are deterministic no matter the rigid transformations on each partner, resulting in an invariant problem with respect to each protein:

$$\phi(\vec{\mathcal{G}}_1, \vec{\mathcal{G}}_2) = \phi(g_1 \cdot \vec{\mathcal{G}}_1, g_2 \cdot \vec{\mathcal{G}}_2), \forall g_1, g_2 \in \text{SE}(3). \tag{71}$$

Methods: The methods dMaSIF [225] and SASNet [226] operate via three-dimensional convolution on the protein 3D structures to keep rotation-invariance. Moreover, fed with more structure features such as distance, orientation and amide angle, DeepInteract [224] adopts geometric transformer and achieves competitive performance as well.

5.6.2   Binding affinity prediction

Protein-protein interactions are fundamental to bio-molecular activity and are crucial for many key functions in biological processes. Estimating the binding affinity between proteins not only aids in gaining a deeper understanding of protein mechanisms of action but also serves as the cornerstone for designing proteins with specific functions, such as highly specific antibodies and high-affinity ligands.

Task definition: Given a pair of proteins that can be considered as geometric graphs $\vec{\mathcal{G}}_1, \vec{\mathcal{G}}_2$, this task requires learning a predictive function ϕ_θ, which can efficiently and accurately predict the binding strength s between the pair of proteins:

$$s = \phi_\theta(\vec{\mathcal{G}}_1, \vec{\mathcal{G}}_2). \tag{72}$$

Symmetry preserved: This is an invariant task because the binding strength s remains unchanged under any translations or rotations applied to the pair of proteins.

Datasets: PDBbind [342] dataset constitutes an assembly of complex structures, meticulously sourced from the Protein Data Bank (PDB), accompanied by binding affinities that have been quantified through rigorous experimental methods. Protein-Protein Affinity Benchmark Version 2 [302,343] encompasses a repertoire of 176 variegated protein-protein complexes, each accompanied by detailed affinity annotations. SKEMPI (Structural database of Kinetics and Energetics of Mutant Protein Interactions) [344] constitutes a curated database that delineates alterations in binding affinities and kinetic parameters consequent to mutagenesis. SKEMPI 2.0 [303] represents the refined and augmented edition of the original SKEMPI database.

Methods: mmCSM-PPI [227] presents a binding affinity prediction method employing graph-based signatures that encapsulate protein structure's physico-chemical and geometric properties, augmented with complementary features to reflect various mechanisms. The Extra Trees model, trained with graph-based signatures and complementary features, yields promising results on the SKEMPI 2.0 dataset. GeoPPI [228] utilizes the 3D conformations to ascertain a geometric representation that embodies the topological features of the protein structure through a self-supervised learning approach. Subsequently, these representations serve as inputs for gradient-boosting trees, facilitating the prediction of the variations in protein-protein binding affinity due to mutations. GET [205] introduces a bilevel design that ensures equivariance while unifying representations across different levels. GET achieves state-of-the-art performance in PDB dataset.

5.6.3   Protein-protein docking

We have investigated docking pose prediction between protein and molecule in Section 5.5.2. Here, we study the similar problem between protein and protein.

Task definition: Assuming two proteins to be denoted as $\vec{\mathcal{G}}_1 = (\vec{X}_1, H_1), \vec{\mathcal{G}}_2 = (\vec{X}_2, H_2)$, respectively, the model needs to learn a prediction function ϕ_θ to output the rotation matrix and translation vector (i.e., $R, \vec{t}$) by

$$R, \vec{t} = \phi_\theta(\vec{X}_1, H_1; \vec{X}_2, H_2). \tag{73}$$

Symmetry preserved: This is identical to Eq. (68).

Methods: Equidock [229] uses SE(3)-equivariant graph neural networks and optimal transport techniques to predict the transformation by aligning key points. HMR [230] casts this task from 3D Euclidean space to 2D Riemannian manifold, keeping rotational invariant. DiffDock-PP [232] extends DiffDock [16], a diffusion generative model, to protein docking task and yields the state-of-the-art performance. Furthermore, in dMaSIF [235], an energy-based, SE(3)-equivariant model combined with physical priors is adopted to infer docking regions. Treating docking as an optimization problem, EBMDock [237] employs geometric deep learning to extract features from protein residues and learns distance distributions between the residues involved in interfaces. Multimetric protein docking can be tackled by AlphaFold-Multimer [234] and SyNDock [233]. Recently, ElliDock [236] predicts SE(3)-equivariant elliptic paraboloids as the binding interface for protein pairs, and transfers the rigid protein-protein docking task into surface fitting while ensuring the same degree of freedom. There are also several works targeting at antibody-antigen docking, a subfield of protein docking. For instance, HSRN [231] proposes a hierarchical framework to handle docking in an iterative manner. By harnessing the capabilities of tFold-Ab [244] and AlphaFold2 [183], tFold-Ag [244] generates antibody/antigen features and employs a docking module to predict complex structures with flexibility.

5.6.4   Antibody design

Antibodies are Y-shaped symmetric proteins produced by the immune system that recognize and bind to specific antigens. The design of antibodies mainly focuses on the variable domains consisting of a heavy chain and a light chain, with 3 Complementarity-Determining Regions (CDRs) and 4 framework regions interleaving on each chain. The 6 CDRs largely determine the binding specificity and affinity of the antibodies, especially CDR-H3 (i.e., the 3rd CDR on the heavy chain), which is the main scope of the design.

Task definition: Without loss of generality, we define the task as a conditional variant of structure and sequence co-design. More specifically, given the geometric graphs of the antigen $\vec{\mathcal{G}}_A$, the heavy chain $\vec{\mathcal{G}}_H$, and the light chain $\vec{\mathcal{G}}_L$ with the CDRs missing, the model ϕ_θ needs to fill in the geometric graph of the CDRs of interest $\vec{\mathcal{G}}_C$:

$$\vec{\mathcal{G}}_C = \phi_\theta(\vec{\mathcal{G}}_A, \vec{\mathcal{G}}_H, \vec{\mathcal{G}}_L). \tag{74}$$

Symmetry preserved: Apparently, the output CDRs $\vec{\mathcal{G}}_C$ should be SE(3)-equivariant with respect to the antigen:

$$g \cdot \vec{\mathcal{G}}_C = \phi_\theta(g \cdot \vec{\mathcal{G}}_A, g \cdot \vec{\mathcal{G}}_H, g \cdot \vec{\mathcal{G}}_L), \forall g \in \text{SE}(3). \tag{75}$$

Methods: Antibody is of great significance in the field of therapeutics and biology, thus many works have dedicated to designing antibodies with desired binding specificity and affinity ([17,32,238–240,242,243]). RefineGNN [239] initiates the first attempt to design CDRs on the heavy chain only. Then MEAN [32] and DiffAb [238] extend to the complete setting where the entire complex (i.e., the antigen, the heavy chain and the light chain) without CDRs are given

as contexts. Notably, MEAN [32] adopts GMN-like [51] multi-channel architecture to encode the backbone atoms of the residues, and proposes an equivariant attention mechanism to capture interactions between different geometric components. Progressively, MEAN is upgraded to dyMEAN [17] which proposes a dynamic multi-channel encoder to capture the full-atom geometry of residues and tackles a more challenging setting where the entire structure and docking pose of the antibody needs to be generated instead of given as contexts. DiffAb [238] proposes a diffusion generative model for antibody design. Similarly, AbDiffuser [243] also adopts diffusion-based generative model, but steps forward to project each side chain into 4 pseudo-carbon atoms to capture the full-atom geometry and handles length change by placeholders in the sequence. ADesigner [241] proposes a cross-gate MLP to facilitate the integration of sequences and structures. Unlike the aforementioned approaches, AbODE [242] explores graph PDEs for antibody design. GeoAB [245] uses torsional prior knowledge with equivariant neural network focusing on bond lengths, bond angles and dihedrals. RADD [246] introduce more node features, edge features, and edge relations to include more contextual and geometric information for designing the CDRs. Further, [240] utilizes pretrained antibody language models to improve the quality of sequence-structure co-design, and tFold-Ab [244] also employs a pretrained language model (i.e., ESM-PPI), along with feature updating (i.e., Evoformer-Single) and structure modules, to enable efficient and accurate prediction of antibody structures directly from sequence.

5.6.5 Peptide design

Peptide, which consists of short sequences of amino acids, represents the intermediate modality between small molecules and proteins, and plays a critical role in various biological functions. Its unique position makes functional peptide design particularly appealing for both biological research and therapeutic applications [345,346].

Task definition: Similar to antibody design, peptide design typically involves generating binding peptides for a given binding area on the target protein. Denoting the target as $\vec{\mathcal{G}}_B$ and the peptide as $\vec{\mathcal{G}}_P$, we can formalize the task as follows:

$$\vec{\mathcal{G}}_P = \phi_\theta(\vec{\mathcal{G}}_B). \tag{76}$$

Symmetry preserved: Akin to antibody design, the output of the model requires to maintain invariance in the sequence distribution and equivariance in the structure distribution in terms of the E(3) group.

Datasets: PepBDB [305] collects 13K protein-peptide complexes with peptides containing fewer than 50 residues from the Protein Data Bank [294]. [307] curates a diverse and non-redundant dataset of 96 protein-peptide complexes, with peptides between 4 and 25 residues, which is referred to as the Long Non-Redundant (LNR) dataset. PepGLAD [35] further collects 6K non-redundant protein-peptide complexes, also featuring peptides between 4 and 25 residues, and partitions them based on the sequence identity of the receptors for training and validation, employing LNR as the test set.

Methods: While conventional approaches rely on empirical energy functions to sample and optimize sequences and structures at the residue or fragment level [347,348], recent advances in geometric molecular design shed light on deep generative models. HelixGAN [247] focuses on a sub-family of peptides with α-helices. RFDiffusion [13], which is originally designed for protein generation, also explores supervised finetuning for target-specific peptide design. PepGLAD [35] takes a step further by tackling sequence-structure co-design with a geometric latent diffusion model.

5.7 Tasks on other domains

We briefly review the applications on other domains such as crystals and RNAs.

5.7.1 Crystal property prediction

In the realm of material science, the prediction of crystalline properties stands as a cornerstone for the innovation of new materials. Unlike molecules or proteins, which consist of a finite number of atoms, crystals are characterized by their periodic repetition throughout infinite 3D space. One of the main challenges lies in capturing this unique periodicity using geometric graph neural networks.

Task definition: The infinite crystal structure is commonly simplified by its repeating unit, which is called a *unit cell*, which is represented as $\vec{\mathcal{G}} = (\vec{L}, \vec{X}, H)$, where $\vec{X}, H$ are coordinate matrix and feature matrix as defined before, and the additional matrix $\vec{L} = [\vec{l}_1, \vec{l}_2, \vec{l}_3]^\top \in \mathbb{R}^{3\times3}$ consists of three lattice vectors determining the periodicity of the crystal. The task is to predict the property $y \in \mathbb{R}$ of the entire structure via the predictor ϕ_θ.

$$y = \phi_\theta(\vec{L}, \vec{X}, H). \tag{77}$$

Symmetry preserved: The output of the predictor should be invariant with respect to several types of groups: 1) E(3)-invariance of both the coordinates $\vec{X}$ and the lattice $\vec{L}$; 2) Periodic translation invariance of $\vec{X}$; 3) Cell choice invariance owing to periodicity, with details referred to [259].

Datasets: Materials Project (MP) [308] and JARVIS-DFT [312] are two commonly-used datasets. In particular, MP is an open-access database containing more than 150K crystal structures with several properties collected by DFT calculation. JARVIS-DFT, part of the Joint Automated Repository for Various Integrated Simulations (JARVIS), is also calculated by DFT and provides more unique properties of materials like solar-efficiency and magnetic moment.

Methods: To take the periodicity into consideration, CGCNN [249] proposes the multi-edge graph construction to model the interactions across the periodic boundaries. MEGNet [250] additionally updates the global state attributes during the message-passing procedure. ALIGNN [251] composes two GNNs for both the atomic bond graph and its line graph to capture the interactions among atomic triplets. ECN [252] leverages space group symmetries into the GNNs for more powerful expressivity. Matformer [253] utilizes self-connecting edges to explicitly introduce the lattice matrix $\vec{L}$ into the transformer-based framework. To utilize the large amount of unlabeled data, Crystal Twins [254] applies two contrastive frameworks, Barlow Twins [349] and SimSiam

[350], to pre-train the CGCNN models, and MMPT [255] proposes a mutex mask strategy to enforce the model to learn representations from two disjoint parts of the crystal.

5.7.2   Crystal generation

Besides predicting the invariant properties of 3D crystals, the rapid progress of geometric graph neural networks has also paved the way to de novo material design, whose goal is to generate novel crystal structures beyond the existing databases.

Task definition: Crystal generation methods commonly integrate geometric graph neural networks into deep generative frameworks, which aims to learn the distribution from a given dataset, allowing to generate new crystals through sampling from the learned distribution:

$$\vec{L}, \vec{X}, H \sim p_\theta(\vec{L}, \vec{X}, H). \qquad (78)$$

Symmetry preserved: Similar to the property prediction task, the learned distribution is also required to be invariant in terms of E(3) group and periodicity.

Datasets: CDVAE [257] collects three datasets, named Perov-5 [309,310], Carbon-24 [311], and MP-20 [308] to evaluate the generative models on different crystal distributions.

Methods: CDVAE [257] incorporates a diffusion-based decoder into a VAE-based framework, by first predicting the lattice parameters from the latent space, and updating the atom types and coordinates according to the predicted lattice. SyMat [49] refines this approach by generating atom types as permutation invariant sets and employing coordinate score-matching for the edges. DiffCSP [50], originally aiming at predicting crystal structures from given composition, also excels in generating structures from scratch. DiffCSP adopts the fractional coordinates $F = \vec{L}^{-1}\vec{X}$ instead of the Cartesian coordinates, and jointly generates the lattice matrix, atom types and coordinates via a diffusion-based framework. DiffCSP++ [258] extends DiffCSP with the conditions of lattice families and Wyckoff coordinates to maintain the space group constraints. Recently, MatterGen [259] further propels the joint diffusion method, and specializes the lattice diffusion process to be cubic-prior and rotation-fixed.

5.7.3   RNA 3D structure ranking

RNA, or ribonucleic acid, is a pivotal type of molecules that goes beyond its traditional role as a mere intermediary between DNA and protein synthesis. Its functionality heavily relies on its intricate three-dimensional structure, making the prediction and ranking of RNA's 3D conformation crucial. This structural complexity enables RNA to participate in gene regulation, cellular communication, and catalysis, underscoring its significance in fundamental life processes. As a result, RNA stands at the forefront of molecular biology and biotechnology research.

Task definition: Here, we refer the ranking of 3D RNA structures to the task of identifying which structure most accurately reflecting the RNA's actual shape from a pool of imprecise ones. In other words, the score model ϕ_θ is required to evaluate the root-mean-square deviation (RMSD) between each candidate 3D RNA structure represented by a geometric graph $\vec{\mathcal{G}}$, and the ground truth:

$$s = \phi_\theta(\vec{\mathcal{G}}). \qquad (79)$$

Symmetry preserved: This is obviously an invariant task because the RMSD value between the candidate structure and the ground truth remains impervious to any translations or rotations imposed on the candidate structure.

Methods: ARES [35] leverages e3nn [351] to model the 3D structure of RNA, ensuring equivariance and invariance during the update of atomic features. ARES then aggregates the features of all atoms to predict the RMSD value. In contrast, PaxNet [264] employs a two-layer multiplex graph to model the 3D structure of RNA. One layer captures local interactions, while the other focuses on non-local interactions. EquiRNA [265] introduces a hierarchical equivariant graph neural network with a size-insensitive K-nearest neighbor sampling strategy, aimed at solving the size generalization challenge through the reuse of nucleotide representations.

Datasets: ARES [35] uses a collection of 18K records from the FARFAR2-Classics dataset [352] as its training and validation sets. In addition, they have constructed two test sets: the first test set was selected from the FARFAR2-Puzzles dataset [352]; the second test set was curated based on certain criteria and built using the FARFAR2 rna denovo application. EquiRNA [265] introduces rRNAsolo, a new dataset for assessing size generalization in RNA structure evaluation. It covers a wider range of RNA sizes, more RNA types, and more recent RNA structures than existing datasets.

6   Discussion and future prospect

Whilst much progress has been made in this field, there are still a broad range of open research directions. We discuss several examples as follows.

Geometric graph foundation model. Recent advancements in AI research, exemplified by the remarkable progress of models like the GPT series [353–355] and Gato [356], have brought about substantial advantages by employing a unified foundational model across various tasks and domains. Foundation models diminish the necessity of manually crafting inductive biases for individual domains, amplifies the volume and variety of training data, and holds promise for further enhancement with increased data, computational resources, and model complexity. It is natural to mimic such success to geometric domain. However, it remains an interesting open question, especially considering the following design spaces. **1. Task space:** How to pretrain a large scale model that is generally beneficial to various downstream tasks? **2. Data space:** How to build a foundation model that can simultaneously extract rich information that spans across different types or scales of the geometric data? **3. Model space:** How to truly scale the model in terms of capacity and expressivity, such that more knowledge can be captured and stored in the model? Although some initial works (such as EPT [90]) manage to pretrain a unified model on small molecules and proteins, it still lacks a universal model that can tackle more kinds of input data and tasks.

Effective loop between model training and real-world experimental verification. Unlike typical applications in

vision and NLP, tasks in science usually require expensive labor, computational resources, and instruments to produce data, conduct verification, and record results. Existing research often adopts an open-loop style, where datasets are collected beforehand and proposed models are evaluated offline on these datasets. However, this approach presents two significant issues. Firstly, the constructed datasets are often small and insufficient for training geometric GNNs, especially for data-hungry foundational models equipped with large-scale parameters. Secondly, evaluating models solely on standalone datasets may fail to reflect feedback from the real world, resulting in less reliable evaluation of the model's true ability. These issues can be effectively addressed by training and testing geometric GNNs within a closed loop between model prediction and experimental verification. A notable example is provided by GNoME [357], which integrates an end-to-end pipeline consisting of graph network training, DFT computations, and autonomous laboratories for materials discovery and synthesis. It is expected that such a research paradigm will become increasingly important in future studies related to scientific applications.

Integration with large language models. Large Language Models (LLMs) have been extensively shown to possess a wealth of knowledge, spanning various domains. Moreover, there has been a development of domain-specific Language Model Agents (LMAs) that exhibit high levels of expertise in specific areas [358,359]. Given that many of the tasks under discussion are intricately linked with the natural sciences, such as physics, biochemistry, and material science, which often require a deep understanding of domain-specific knowledge, it becomes compelling to enhance the existing knowledge base by integrating LLM agents into the training and evaluation pipeline of geometric Graph Neural Networks (GNNs). This integration holds promise for augmenting the capabilities of GNNs by leveraging the comprehensive knowledge representations offered by LLMs, thereby potentially improving the performance and robustness of these models in scientific applications. While there have been works leveraging LLMs for certain tasks such as molecule property prediction and drug design, they only operate on motifs [360,361] or molecule graphs [362]. It still remains challenging to bridge them with geometric graph neural networks, enabling the pipeline to process 3D structural information and perform prediction and/or generation over 3D structures.

Relaxation of equivariance. While equivariance is undeniably pivotal for bolstering data efficiency and promoting generalization across diverse datasets, it is noteworthy that rigidly adhering to equivariance principles can sometimes overly constrain the model, potentially compromising its performance. Thus, delving into methodologies that offer a degree of flexibility in relaxing equivariance constraints holds considerable significance. By exploring approaches that strike a balance between maintaining equivariance and accommodating adaptability, researchers can unlock avenues for enhancing the practical utility of models. Several pioneer studies [363,364] try to relax the equivariance to a certain discrete point group and

achieves a remarkable improvement on various dynamic physical systems, ranging from particle to vehicle dynamics. This exploration may not only enrich our understanding of model behavior but also pave the way for the development of more robust and versatile solutions with broader applicability.

7 Conclusion

In this survey, we conduct a systematic investigation of the progress in geometric Graph Neural Networks (GNNs), through the lens of data structures, models, and their applications. We specify geometric graph as the data structure, which generalizes the concept of graph in the presence of geometric information and permits the vital symmetry under certain transformations. We present geometric GNNs as the models, which consist of invariant GNNs, scalarization-based/high-degree steerable equivariant GNNs, and geometric graph transformers. We exhaustively discuss their applications through the taxonomy on the data and tasks, including both single instance and multi-instance tasks over domains in physics, biochemistry, and others like materials and RNAs. We also discuss the challenges and the future potential directions of geometric GNNs.

References

1. Bronstein M M, Bruna J, Cohen T, Veličković P. Geometric deep learning: grids, groups, graphs, geodesics, and gauges. 2021, arXiv preprint arXiv: 2104.13478

2. Schütt K T, Arbabzadah F, Chmiela S, Müller K R, Tkatchenko A. Quantum-chemical insights from deep tensor neural networks. Nature Communications, 2017, 8: 13890

3. Klicpera J, Groß J, Gunnemann S. Directional message passing for molecular graphs. In: Proceedings of the 8th International Conference on Learning Representations. 2020

4. Klicpera J, Becker F, Günnemann S. GemNet: universal directional graph neural networks for molecules. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. 2021, 520

5. Satorras V G, Hoogeboom E, Welling M. E(n) equivariant graph

neural networks. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 9323−9332

6.   Schütt K, Unke O, Gastegger M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 9377−9388

7.   Thomas N, Smidt T, Kearnes S, Yang L, Li L, Kohlhoff K, Riley P. Tensor field networks: rotation- and translation-equivariant neural networks for 3D point clouds. 2018, arXiv preprint arXiv: 1802.08219

8.   Fuchs F B, Worrall D E, Fischer V, Welling M. SE(3)-Transformers: 3D roto-translation equivariant attention networks. In: Proceedings of the 34th Conference on Neural Information Processing Systems. 2020

9.   Brandstetter J, Hesselink R, van der Pol E, Bekkers E J, Welling M. Geometric and physical quantities improve E(3) equivariant message passing. In: Proceedings of the 10th International Conference on Learning Representations. 2022

10. Batzner S, Musaelian A, Sun L, Geiger M, Mailoa J P, Kornbluth M, Molinari N, Smidt T E, Kozinsky B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. Nature Communications, 2022, 13(1): 2453

11. Liao Y L, Smidt T E. Equiformer: equivariant graph attention transformer for 3D atomistic graphs. In: Proceedings of the 11th International Conference on Learning Representations. 2023

12. Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, et al. Accurate prediction of protein structures and interactions using a three-track neural network. Science, 2021, 373(6557): 871−876

13. Watson J L, Juergens D, Bennett N R, Trippe B L, Yim J, et al. De novo design of protein structure and function with RFdiffusion. Nature, 2023, 620(7976): 1089−1100

14. Ingraham J B, Baranov M, Costello Z, Barber K W, Wang W, et al. Illuminating protein space with a programmable generative model. Nature, 2023, 623(7989): 1070−1078

15. Townshend R J L, Eismann S, Watkins A M, Rangan R, Karelina M, Das R, Dror R O. Geometric deep learning of RNA structure. Science, 2021, 373(6558): 1047−1051

16. Corso G, Stärk H, Jing B, Barzilay R, Jaakkola T S. DiffDock: diffusion steps, twists, and turns for molecular docking. In: Proceedings of the 11th International Conference on Learning Representations. 2023

17. Kong X, Huang W, Liu Y. End-to-end full-atom antibody design. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 718

18. Gilmer J, Schoenholz S S, Riley P F, Vinyals O, Dahl G E. Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning. 2017, 1263−1272

19. McNutt A T, Francoeur P, Aggarwal R, Masuda T, Meli R, Ragoza M, Sunseri J, Koes D R. GNINA 1.0: molecular docking with deep learning.Journal of Cheminformatics, 2021, 13(1): 43

20. Adolf-Bryfogle J, Kalyuzhniy O, Kubitz M, Weitzner B D, Hu X, Adachi Y, Schief W R, Dunbrack Jr R L. RosettaAntibodyDesign (RAbD): a general framework for computational antibody design. PLoS Computational Biology, 2018, 14(4): e1006112

21. Ramakrishnan R, Dral P O, Rupp M, von Lilienfeld O A. Quantum chemistry structures and properties of 134 kilo molecules. Scientific Data, 2014, 1: 140022

22. Liu Z, Su M, Han L, Liu J, Yang Q, Li Y, Wang R. Forging the basis for developing protein–ligand interaction scoring functions. Accounts of Chemical Research, 2017, 50(2): 302−309

23. Dunbar J, Krawczyk K, Leem J, Baker T, Fuchs A, Georges G, Shi J, Deane C M. SAbDab: the structural antibody database. Nucleic Acids Research, 2014, 42(D1): D1140−D1146

24. Han J, Rong Y, Xu T, Huang W. Geometrically equivariant graph neural networks: a survey. 2022, arXiv preprint arXiv: 2202.07230

25. Han J, Huang W, Ma H, Li J, Tenenbaum J B, Gan C. Learning physical dynamics with subequivariant graph neural networks. In: Proceedings of the 36th Conference on Neural Information Processing Systems. 2022

26. Sanchez-Gonzalez A, Godwin J, Pfaff T, Ying R, Leskovec J, Battaglia P. Learning to simulate complex physics with graph networks. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 8459−8468

27. Kipf T, Fetaya E, Wang K C, Welling M, Zemel R. Neural relational inference for interacting systems. In: Proceedings of the 35th International Conference on Machine Learning. 2018, 2688−2697

28. Huang Y, Peng X, Ma J, Zhang M. 3DLinker: an E(3) equivariant variational autoencoder for molecular linker design. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 9280−9294

29. Guan J, Qian W W, Peng X, Su Y, Peng J, Ma J. 3D equivariant diffusion for target-aware molecule generation and affinity prediction. In: Proceedings of the 11th International Conference on Learning Representations. 2023

30. Jing B, Corso G, Chang J, Barzilay R, Jaakkola T. Torsional diffusion for molecular conformer generation. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1760

31. Wu L, Hou Z, Yuan J, Rong Y, Huang W. Equivariant spatio-temporal attentive graph networks to simulate physical dynamics. In: Proceedings of the 37th International Conference on Neural Information Processing System. 2023, 1965

32. Kong X, Huang W, Liu Y. Conditional antibody design as 3D equivariant graph translation. In: Proceedings of the 11th International Conference on Learning Representations. 2023

33. Senior A W, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, Qin C, Žídek A, Nelson A W R, Bridgland A, Penedones H, Petersen S, Simonyan K, Crossan S, Kohli P, Jones D T, Silver D, Kavukcuoglu K, Hassabis D. Improved protein structure prediction using potentials from deep learning. Nature, 2020, 577(7792): 706−710

34. Chanussot L, Das A, Goyal S, Lavril T, Shuaibi M, Riviere M, Tran K, Heras-Domingo J, Ho C, Hu W, Palizhati A, Sriram A, Wood B, Yoon J, Parikh D, Zitnick C L, Ulissi Z. Open catalyst 2020 (OC20) dataset and community challenges. ACS Catalysis, 2021, 11(10): 6059−6072

35. Kong X, Jia Y, Huang W, Liu Y. Full-atom peptide design with geometric latent diffusion. In: Proceedings of the 38th Conference on Neural Information Processing Systems. 2024

36. Duval A, Mathis S V, Joshi C K, Schmidt V, Miret S, Malliaros F D, Cohen T, Liò P, Bengio Y, Bronstein M. A hitchhiker's guide to geometric GNNs for 3D atomic systems. 2024, arXiv preprint arXiv: 2312.07511

37. Xia J, Zhu Y, Du Y, Li S Z. A systematic survey of chemical pre-trained models. In: Proceedings of the 32nd International Joint Conference on Artificial Intelligence. 2023, 6787−6795

38. Guo Z, Guo K, Nan B, Tian Y, Iyer R G, Ma Y, Wiest O, Zhang X, Wang W, Zhang C, Chawla N V. Graph-based molecular representation learning. In: Proceedings of the 32nd International Joint Conference on Artificial Intelligence. 2023, 6638−6646.

39. Atz K, Grisoni F, Schneider G. Geometric deep learning on molecular representations. Nature Machine Intelligence, 2021, 3(12): 1023−1032

40. Zhang X, Wang L, Helwig J, Luo Y, Fu C, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. 2025, arXiv preprint arXiv: 2307.08423

41. Esteves C. Theoretical aspects of group equivariant neural networks.

2020, arXiv preprint arXiv: 2004.05154

42. Cederberg J. A course in modern geometries. Springer Science & Business Media, 2004

43. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip S Y. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(1): 4−24

44. Yuan Z, Wei Z, Lv F, Wen J R. Index-free triangle-based graph local clustering. Frontiers of Computer Science, 2024, 18(3): 183404

45. Wu Z, Ramsundar B, Feinberg E N, Gomes J, Geniesse C, Pappu A S, Leswing K, Pande V. MoleculeNet: a benchmark for molecular machine learning. Chemical Science, 2018, 9(2): 513−530

46. Villar S, Hogg D W, Storey-Fisher K, Yao W, Blum-Smith B. Scalars are universal: equivariant machine learning, structured like classical physics. In: Proceedings of the 35th Conference on Neural Information Processing Systems. 2021

47. Schutt K T, Sauceda H E, Kindermans P J, Tkatchenko A, Müller K R. SchNet–a deep learning architecture for molecules and materials. The Journal of Chemical Physics, 2018, 148(24): 241722

48. Baek M, Anishchenko I, Humphreys I R, Cong Q, Baker D, DiMaio F. Efficient and accurate prediction of protein structure using RoseTTAFold2. bioRxiv, 2023

49. Luo Y, Liu C, Ji S. Towards symmetry-aware generation of periodic materials. In: Proceedings of the 37th Conference on Neural Information Processing Systems. 2023, 36

50. Jiao R, Huang W, Lin P, Han J, Chen P, Lu Y, Liu Y. Crystal structure prediction by joint equivariant diffusion. In: Proceedings of the 37th International Conference on Neural Information Processing System. 2023, 767

51. Huang W, Han J, Rong Y, Xu T, Sun F, Huang J. Equivariant graph mechanics networks with constraints. In: Proceedings of the 10th International Conference on Learning Representations. 2022

52. Gasteiger J, Giri S, Margraf J T, Günnemann S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. 2022, arXiv preprint arXiv: 2011.14115

53. Zhu F, Futrega M, Bao H, Eryilmaz S B, Kong F, Duan K, Zheng X, Angel N, Jouanneaux M, Stadler M, Marcinkiewicz M, Xie F, Yang J, Andersch M. FastDimeNet++: training DimeNet++ in 22 minutes. In: Proceedings of the 52nd International Conference on Parallel Processing. 2023, 274−284

54. Finzi M, Stanton S, Izmailov P, Wilson A G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 3165−3176

55. Liu Y, Wang L, Liu M, Lin Y, Zhang X, Oztekin B, Ji S. Spherical message passing for 3D molecular graphs. In: Proceedings of the 10th International Conference on Learning Representations. 2022

56. Wang L, Liu Y, Lin Y, Liu H, Ji S. ComENet: towards complete and efficient message passing for 3D molecular graphs. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 47

57. Li Z, Wang X, Huang Y, Zhang M. Is distance matrix enough for geometric deep learning? In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 1627

58. Li Z, Wang X, Kang S, Zhang M. On the completeness of invariant geometric deep learning models. 2024, arXiv preprint arXiv: 2402.04836

59. Yue A, Luo D, Xu H. A plug-and-play quaternion message-passing module for molecular conformation representation. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 16633−16641

60. Du W, Zhang H, Du Y, Meng Q, Chen W, Zheng N, Shao B, Liu T Y. SE(3) equivariant graph neural networks with complete local frames. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 5583−5608

61. Kofinas M, Nagaraja N S, Gavves E. Roto-translated local coordinate frames for interacting dynamical systems. In: Proceedings of the 35th Conference on Neural Information Processing Systems. 2021

62. Kofinas M, Bekkers E J, Nagaraja N S, Gavves E. Latent field discovery in interacting dynamical systems with neural fields. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 1379

63. Kohler J, Klein L, Noé F. Equivariant flows: sampling configurations for multi-body systems with symmetric energies. 2019, arXiv preprint arXiv: 1910.00753

64. Jing B, Eismann S, Suriana P, Townshend R J L, Dror R O. Learning from protein structure with geometric vector perceptrons. In: Proceedings of the 9th International Conference on Learning Representations. 2021

65. Han J, Huang W, Xu T, Rong Y. Equivariant graph hierarchy-based neural networks. In: Proceedings of the 36th Conference on Neural Information Processing Systems. 2022

66. Zhang Y, Cen J, Han J, Zhang Z, Zhou J, Huang W. Improving equivariant graph neural networks on large geometric graphs via virtual nodes learning. In: Proceedings of the 41st International Conference on Machine Learning. 2024

67. Puny O, Atzmon M, Smith E J, Misra I, Grover A, Ben-Hamu H, Lipman Y. Frame averaging for invariant and equivariant network design. In: Proceedings of the 10th International Conference on Learning Representations. 2022

68. Duval A A, Schmidt V, Hernández-Garcıa A, Miret S, Malliaros F D, Bengio Y, Rolnick D. FAENet: frame averaging equivariant GNN for materials modeling. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 9013−9033

69. Du W, Du Y, Wang L, Feng D, Wang G, Ji S, Gomes C P, Ma Z M. A new perspective on building efficient and expressive 3D equivariant graph neural networks. In: Proceedings of the 37th International Conference on Neural Information Processing System. 2023, 2910

70. Aykent S, Xia T. SaVeNet: a scalable vector network for enhanced molecular representation learning. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 1860

71. Wang Y, Wang T, Li S, He X, Li M, Wang Z, Zheng N, Shao B, Liu T Y. Enhancing geometric representations for molecules with equivariant vector-scalar interactive message passing. Nature Communications, 2024, 15(1): 313

72. Wang Z, Liu G, Zhou Y, Wang T, Shao B. QuinNet: efficiently incorporating quintuple interactions into geometric deep learning force fields. In: Proceedings of the 37th Conference on Neural Information Processing Systems. 2023, 3368

73. Cen J, Li A, Lin N, Ren Y, Wang Z, Huang W. Are high-degree representations really unnecessary in equivariant graph neural networks? In: Proceedings of the 38th Conference on Neural Information Processing Systems. 2024

74. Battiloro C, Karaismailoglu E, Tec M, Da-soulas G, Audirac M, Dominici F. E(n) equivariant topological neural networks. In: Proceedings of the Thirteenth International Conference on Learning Representations. 2025

75. Li Z, Cen J, Su B, Huang W, Xu T, Rong Y, Zhao D. Large language-geometry model: when LLM meets equivariance. 2025, arXiv preprint

arXiv: 2502.11149

76.   Anderson B, Hy T S, Kondor R. Cormorant: covariant molecular neural networks. In: Proceedings of the 33rd Conference on Neural Information Processing Systems. 2019

77.   Musaelian A, Batzner S, Johansson A, Sun L, Owen C J, Kornbluth M, Kozinsky B. Learning local equivariant representations for large-scale atomistic dynamics. Nature Communications, 2023, 14(1): 579

78.   Zitnick C L, Das A, Kolluru A, Lan J, Shuaibi M, Sriram A, Ulissi Z, Wood B. Spherical channels for modeling atomic interactions. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 585

79.   Passaro S, Zitnick C L. Reducing SO(3) convolutions to SO(2) for efficient equivariant GNNs. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 1140

80.   Batatia I, Kovács D P, Simm G N C, Ortner C, Csányi G. MACE: higher order equivariant message passing neural networks for fast and accurate force fields. In: Proceedings of the 36th Conference on Neural Information Processing Systems. 2022, 11423−11436

81.   Ying C, Cai T, Luo S, Zheng S, Ke G, He D, Shen Y, Liu T Y. Do transformers really perform bad for graph representation? In: Proceedings of the 35th Conference on Neural Information Processing Systems. 2021

82.   Shi Y, Zheng S, Ke G, Shen Y, You J, He J, Luo S, Liu C, He D, Liu T Y. Benchmarking graphormer on large-scale molecular modeling datasets. 2023, arXiv preprint arXiv: 2203.04810

83.   Thölke P, de Fabritiis G. Equivariant transformers for neural network based molecular potentials. In: Proceedings of the 10th International Conference on Learning Representations. 2022

84.   Hutchinson M J, Le Lan C, Zaidi S, Dupont E, Teh Y W, Kim H. Lietransformer: equivariant self-attention for Lie groups. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 4533−4543

85.   Hsu C, Verkuil R, Liu J, Lin Z, Hie B, Sercu T, Lerer A, Rives A. Learning inverse folding from millions of predicted structures. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 8946−8970

86.   Liao Y L, Wood B M, Das A, Smidt T E. EquiformerV2: improved equivariant transformer for scaling to higher-degree representations. In: Proceedings of the 12th International Conference on Learning Representations. 2024

87.   Wang Y, Li S, Wang T, Shao B, Zheng N, Liu T Y. Geometric transformer with interatomic positional encoding. In: Proceedings of the 37th Conference on Neural Information Processing Systems. 2023, 36

88.   Frank J T, Unke O T, Müller K R, Chmiela S. A Euclidean transformer for fast and stable machine learned force fields. Nature Communications, 2024, 15(1): 6539

89.   Aykent S, Xia T. GotenNet: rethinking efficient 3D equivariant graph neural networks. In: Proceedings of the 13th International Conference on Learning Representations. 2025

90.   Jiao R, Kong X, Yu Z, Huang W, Liu Y. Equivariant pretrained transformer for unified geometric learning on multi-domain 3D molecules. 2025, arXiv preprint arXiv: 2402.12714v1

91.   Ma H, Bian Y, Rong Y, Huang W, Xu T, Xie W, Ye G, Huang J. Cross-dependent graph neural networks for molecular property prediction. Bioinformatics, 2022, 38(7): 2003−2009

92.   Zhang M, Li P. Nested graph neural networks. In: Proceedings of the 35th Conference on Neural Information Processing Systems. 2021, 15734−15747

93.   Qin S, Zhang X, Xu H, Xu Y. Fast quaternion product units for learning disentangled representations in SO(3). IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(4): 4504−4520

94.   Zhu X, Xu Y, Xu H, Chen C. Quaternion convolutional neural networks. In: Proceedings of the 15th European Conference on Computer Vision. 2018, 645−661

95.   Zhang X, Qin S, Xu Y, Xu H. Quaternion product units for deep learning on 3D rotation groups. In: Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, 7302−7311

96.   Joshi C K, Bodnar C, Mathis S V, Cohen T, Liò P. On the expressive power of geometric graph neural networks. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 625

97.   Gilmore R. Lie Groups, Physics, and Geometry: An Introduction for Physicists, Engineers and Chemists. Cambridge: Cambridge University Press, 2008

98.   Müller C. Spherical Harmonics. Berlin: Springer, 2006

99.   Griffiths D J, Schroeter D F. Introduction to Quantum Mechanics. Cambridge: Cambridge University Press, 2018

100.   Weiler M, Geiger M, Welling M, Boomsma W, Cohen T. 3D steerable CNNs: learning rotationally equivariant features in volumetric data. In: Proceedings of the 32nd Conference on Neural Information Processing Systems. 2018, 31

101.   Ramachandran P, Zoph B, Le Q V. Searching for activation functions. In: Proceedings of the 6th International Conference on Learning Representations. 2018

102.   Drautz R. Atomic cluster expansion for accurate and transferable interatomic potentials. Physical Review B, 2019, 99(1): 014104

103.   Dusson G, Bachmayr M, Csányi G, Drautz R, Etter S, van der Oord C, Ortner C. Atomic cluster expansion: completeness, efficiency and stability. Journal of Computational Physics, 2022, 454: 110946

104.   Bochkarev A, Lysogorskiy Y, Menon S, Qamar M, Mrovec M, Drautz R. Efficient parametrization of the atomic cluster expansion. Physical Review Materials, 2022, 6(1): 013804

105.   Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017, 6000−6010

106.   Yuan C, Zhao K, Kuruoglu E E, Wang L, Xu T, Huang W, Zhao D, Cheng H, Rong Y. A survey of graph transformers: Architectures, theories and applications. arXiv preprint arXiv: 2502.16533, 2025

107.   Hu W, Fey M, Ren H, Nakata M, Dong Y, Leskovec J. OGB-LSC: a large-scale challenge for machine learning on graphs. In: Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks. 2021

108.   Shuaibi M, Kolluru A, Das A, Grover A, Sriram A, Ulissi Z, Zitnick C L. Rotation invariant graph neural networks using spin convolutions. 2021, arXiv preprint arXiv: 2106.09575

109.   Dym N, Maron H. On the universality of rotation equivariant point cloud networks. In: Proceedings of the 9th International Conference on Learning Representations. 2021

110.   Weisfeiler B, Leman A. The reduction of a graph to canonical form and the algebra which appears therein. Nauchno-Technicheskaya Informatsia, 1968, 2(9): 12−16

111.   Lawrence H, Portilheiro V, Zhang Y, Kaba S O. Improving equivariant networks with probabilistic symmetry breaking. In: Proceedings of the Geometry-Grounded Representation Learning and Generative Modeling at 41st International Conference on Machine Learning. 2024

112.   Battaglia P, Pascanu R, Lai M, Jimenez Rezende D, Kavukcuoglu K. Interaction networks for learning about objects, relations and physics.

In: Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016, 4509−4517

113. Sanchez-Gonzalez A, Bapst V, Cranmer K, Battaglia P. Hamiltonian graph networks with ode integrators. 2019, arXiv preprint arXiv: 1909.12790

114. Guo L, Wang W, Chen Z, Zhang N, Sun Z, Lai Y, Zhang Q, Chen H. Newton–cotes graph neural networks: on the time evolution of dynamic systems. In: Proceedings of the 37th Conference on Neural Information Processing Systems. 2023, 36

115. Allen K R, Guevara T L, Rubanova Y, Stachenfeld K, Sanchez-Gonzalez A, Battaglia P, Pfaff T. Graph network simulators can learn discontinuous, rigid contact dynamics. In: Proceedings of the 6th Conference on Robot Learning. 2023, 1157−1167

116. Rubanova Y, Sanchez-Gonzalez A, Pfaff T, Battaglia P. Constraint-based graph network simulator. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 18844−18870

117. Wu T, Wang Q, Zhang Y, Ying R, Cao K, Sosic R, Jalali R, Hamam H, Maucec M, Leskovec J. Learning large-scale subsurface simulations with a hybrid graph network simulator. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2022, 4184−4194

118. Li Y, Wu J, Tedrake R, Tenenbaum J B, Torralba A. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In: Proceedings of the 7th International Conference on Learning Representations. 2019

119. Mrowca D, Zhuang C, Wang E, Haber N, Fei-Fei L, Tenenbaum J B, Yamins D L K. Flexible neural representation for physics prediction. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. 2018, 8813−8824

120. Allen K R, Rubanova Y, Lopez-Guevara T, Whitney W, Sanchez-Gonzalez A, Battaglia P W, Pfaff T. Learning rigid dynamics with face interaction graph networks. In: Proceedings of the 11th International Conference on Learning Representations. 2023

121. Xu C, Tan R T, Tan Y, Chen S, Wang Y G, Wang X, Wang Y. EqMotion: equivariant multi-agent motion prediction with invariant interaction reasoning. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 1410−1420

122. Liu Y, Cheng J, Zhao H, Xu T, Zhao P, Tsung F G, Li J, Rong Y. Improving generalization in equivariant graph neural networks with physical inductive biases. In: Proceedings of the 12th International Conference on Learning Representations. 2024

123. Coors B, Condurache A P, Geiger A. SphereNet: learning spherical representations for detection and classification in omnidirectional images. In: Proceedings of the 15th European Conference on Computer Vision. 2018, 525−541

124. Wang X, Zhang M. Graph neural network with local frame for molecular potential energy surface. In: Proceedings of the 1st Learning on Graphs Conference. 2022, 19

125. Luo S, Chen T, Krishnapriyan A S. Enabling efficient equivariant operations in the Fourier basis via gaunt tensor products. In: Proceedings of the 12th International Conference on Learning Representations. 2024

126. Köhler J, Klein L, Noe F. Equivariant flows: exact likelihood generative learning for symmetric densities. In: Proceedings of the 37th International Conference on Machine Learning. 2020, 5361−5370

127. Xu M, Han J, Lou A, Kossaifi J, Ramanathan A, Azizzadenesheli K, Leskovec J, Ermon S, Anandkumar A. Equivariant graph neural operator for modeling 3D dynamics. In: Proceedings of the 41st International Conference on Machine Learning. 2024

128. Schreiner M, Winther O, Olsson S. Implicit transfer operator learning: multiple time-resolution surrogates for molecular dynamics. In:

Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 1582

129. Midgley L I, Stimper V, Antorán J, Mathieu E, Schölkopf B, Hernández-Lobato J M. SE(3) equivariant augmented coupling flows. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 3466

130. Han J, Xu M, Lou A, Ye H, Ermon S. Geometric trajectory diffusion models. In: Proceedings of the 38th Conference on Neural Information Processing Systems. 2024

131. Raja S, Amin I, Pedregosa F, Krishnapriyan A S. Stability-aware training of neural network interatomic potentials with differentiable Boltzmann estimators. 2025, arXiv preprint arXiv: 2402.13984v1

132. Amin I, Raja, Krishnapriyan A S. Towards fast, specialized machine learning force fields: distilling foundation models via energy hessians. In: Proceedings of the 13th International Conference on Learning Representations. 2025

133. Xu M, Yu L, Song Y, Shi C, Ermon S, Tang J. GeoDiff: a geometric diffusion model for molecular conformation generation. In: Proceedings of the 10th International Conference on Learning Representations. 2022

134. Xu M, Powers A S, Dror R O, Ermon S, Leskovec J. Geometric latent diffusion models for 3D molecule generation. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 38592−38610

135. Xu M, Wang W, Luo S, Shi C, Bengio Y, Gomez-Bombarelli R, Tang J. An end-to-end framework for molecular conformation generation via bilevel programming. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 11537−11547

136. Shi C, Luo S, Xu M, Tang J. Learning gradient fields for molecular conformation generation. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 9558−9568

137. Gebauer N W A, Gastegger M, Schutt K T. Symmetry-adapted generation of 3D point sets for the targeted discovery of molecules. In: Proceedings of the 33rd Conference on Neural Information Processing Systems. 2019, 32

138. Gebauer N W A, Gastegger M, Hessmann S S P, Müller K R, Schütt K T. Inverse design of 3D molecular structures with conditional generative neural networks. Nature Communications, 2022, 13(1): 973

139. Huang L, Zhang H, Xu T, Wong K C. MDM: molecular diffusion model for 3D molecule generation. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence. 2023, 5105−5112

140. Peng X, Guan J, Liu Q, Ma J. MolDiff: addressing the atom-bond inconsistency problem in 3D molecule diffusion generation. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 27611−27629

141. Luo S, Shi C, Xu M, Tang J. Predicting molecular conformation via dynamic graph score matching. In: Proceedings of the 35th Conference on Neural Information Processing Systems. 2021

142. Satorras V G, Hoogeboom E, Fuchs F B, Posner I, Welling M. E(n) equivariant normalizing flows. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. 2021, 320

143. Hoogeboom E, Satorras V G, Vignac C, Welling M. Equivariant diffusion for molecule generation in 3D. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 8867−8887

144. Ganea O E, Pattanaik L, Coley C W, Barzilay R, Jensen K F, Green W H, Jaakkola T S. GEOMOL: torsional geometric generation of molecular 3D conformer ensembles. In: Proceedings of the 35th Conference on Neural Information Processing Systems. 2021

145. Wang F, Xu H, Chen X, Lu S, Deng Y, Huang W. MPerformer: an SE(3) transformer-based molecular perceptron. In: Proceedings of the

32nd ACM International Conference on Information and Knowledge Management. 2023, 2512−2522

146. Bao F, Zhao M, Hao Z, Li P, Li C, Zhu J. Equivariant energy-guided SDE for inverse molecular design. In: Proceedings of the 11th International Conference on Learning Representations. 2023

147. Zhu J, Xia Y, Liu C, Wu L, Xie S, Wang Y, Wang T, Qin T, Zhou W, Li H, Liu H, Liu T Y. Direct molecular conformation generation. Transactions on Machine Learning Research, 2022, See openreview. net/forum?id=lCPOHiztuw website, 2022

148. Qiang B, Song Y, Xu M, Gong J, Gao B, Zhou H, Ma W Y, Lan Y. Coarse-to-fine: a hierarchical diffusion model for molecule generation in 3D. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 28277–28299

149. Song Y, Gong J, Xu M, Cao Z, Lan Y, Ermon S, Zhou H, Ma W Y. Equivariant flow matching with hybrid probability transport for 3D molecule generation. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 26

150. Reidenbach D, Krishnapriyan A S. Coarsenconf: equivariant coarsening with aggregated attention for molecular conformer generation. Journal of Chemical Information and Modeling, 2025, 65(1): 22−30

151. Song Y, Gong J, Zhou H, Zheng M, Liu J, Ma W Y. Unified generative modeling of 3D molecules with Bayesian flow networks. In: Proceedings of the 12th International Conference on Learning Representations. 2024

152. Qu Y, Qiu K, Song Y, Gong J, Han J, Zheng M, Zhou H, Ma W Y. MolCRAFT: structure-based drug design in continuous parameter space. In: Proceedings of the 41st International Conference on Machine Learning. 2024

153. Jiao R, Han J, Huang W, Rong Y, Liu Y. Energy-motivated equivariant pretraining for 3D molecular graphs. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence. 2023, 8096−8104

154. Liu S, Guo H, Tang J. Molecular geometry pretraining with SE(3)-invariant denoising distance matching. In: Proceedings of the 11th International Conference on Learning Representations. 2023

155. Liu S, Wang H, Liu W, Lasenby J, Guo H, Tang J. Pre-training molecular graph representation with 3D geometry. In: Proceedings of the 10th International Conference on Learning Representations. 2022

156. Zaidi S, Schaarschmidt M, Martens J, Kim H, Teh Y W, Sanchez-Gonzalez A, Battaglia P W, Pascanu R, Godwin J. Pre-training via denoising for molecular property prediction. In: Proceedings of the 11th International Conference on Learning Representations. 2023

157. Feng J, Wang Z, Li Y, Ding B, Wei Z, Xu H. MGMAE: molecular representation learning by reconstructing heterogeneous graphs with a high mask ratio. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. 2022, 509−519

158. Stärk H, Beaini D, Corso G, Tossou P, Dallago C, Gunnemann S, Lió P. 3D infomax improves GNNs for molecular property prediction. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 20479−20502

159. Zhou G, Gao Z, Ding Q, Zheng H, Xu H, Wei Z, Zhang L, Ke G. Uni-Mol: a universal 3D molecular representation learning framework. In: Proceedings of the 11th International Conference on Learning Representations. 2023

160. Luo S, Chen T, Xu Y, Zheng S, Liu T Y, Wang L, He D. One transformer can understand both 2D & 3D molecular data. In: Proceedings of the 11th International Conference on Learning Representations. 2023

161. Liu S, Du W, Ma Z M, Guo H, Tang J. A group symmetric stochastic differential equation model for molecule multi-modal pretraining. In:

Proceedings of the 40th International Conference on Machine Learning. 2023, 21497–21526

162. Ni Y, Feng S, Ma W Y, Ma Z M, Lan Y. Sliced denoising: a physics-informed molecular pre-training method. In: Proceedings of the 12th International Conference on Learning Representations. 2024

163. Feng S, Ni Y, Lan Y, Ma Z M, Ma W Y. Fractional denoising for 3D molecular pre-training. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 9938−9961

164. Liu Y, Chen J, Jiao R, Li J, Huang W, Su B. DenoiseVAE: learning molecule-adaptive noise distributions for denoising-based 3D molecular pre-training. In: Proceedings of the 13th International Conference on Learning Representations. 2025

165. Liu S, Rong Y, Zhao D, Liu Q, Wu S, Wang L. MolSpectra: pre-training 3D molecular representation with multi-modal energy spectra. In: Proceedings of the 13th International Conference on Learning Representations. 2025

166. Wang Z, Combs S A, Brand R, Calvo M R, Xu P, Price G, Golovach N, Salawu E O, Wise C J, Ponnapalli S P, Clark P M. LM-GVP: an extensible sequence and structure informed deep learning framework for protein property prediction. Scientific Reports, 2022, 12(1): 6832

167. Gligorijević V, Renfrew P D, Kosciolek T, Leman J K, Berenberg D, Vatanen T, Chandler C, Taylor B C, Fisk I M, Vlamakis H, Xavier R J, Knight R, Cho K, Bonneau R. Structure-based protein function prediction using graph convolutional networks. Nature Communications, 2021, 12(1): 3168

168. Zhang Z, Xu M, Jamasb A R, Chenthamarakshan V, Lozano A C, Das P, Tang J. Protein representation learning by geometric structure pretraining. In: Proceedings of the 11th International Conference on Learning Representations. 2023

169. Torng W, Altman R B. 3D deep convolutional neural networks for amino acid environment similarity analysis. BMC Bioinformatics, 2017, 18(1): 302

170. Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the tm-score. Nucleic Acids Research, 2005, 33(7): 2302−2309

171. Eismann S, Townshend R J L, Thomas N, Jagota M, Jing B, Dror R O. Hierarchical, rotation-equivariant neural networks to select structural models of protein complexes. Proteins: Structure, Function, and Bioinformatics, 2021, 89(5): 493−501

172. Eismann S, Suriana P, Jing B, Townshend R J L, Dror R O. Protein model quality assessment using rotation-equivariant transformations on point clouds. Proteins: Structure, Function, and Bioinformatics, 2023, 91(8): 1089−1096

173. Chen C, Chen X, Morehead A, Wu T, Cheng J. 3D-equivariant graph neural networks for protein model quality assessment. Bioinformatics, 2023, 39(1): btad030

174. Tubiana J, Schneidman-Duhovny D, Wolfson H J. ScanNet: an interpretable geometric deep learning model for structure-based protein binding site prediction. Nature Methods, 2022, 19(6): 730−739

175. Zhang Y, Wei Z, Yuan Y, Ding Z, Huang W. EquiPocket: an E(3)-equivariant geometric graph neural network for ligand binding site prediction. In: Proceedings of the 41st International Conference on Machine Learning. 2024

176. Meller A, Ward M D, Borowsky J H, Lotthammer J M, Kshirsagar M, Oviedo F, Ferres J L, Bowman G. Predicting the locations of cryptic pockets from single protein structures using the pocketminer graph neural network. Biophysical Journal, 2023, 122(3Suppl): 445A

177. Ingraham J, Garg V K, Barzilay R, Jaakkola T. Generative models for graph-based protein design. In: Proceedings of the 33rd Conference on Neural Information Processing Systems. 2019, 32

178. Tan C, Gao Z, Xia J, Hu B, Li S Z. Generative *de novo* protein design

with global context. 2023, arXiv preprint arXiv: 2204.10673

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

180. Gao Z, Tan C, Li S Z. PiFold: toward effective and efficient protein inverse folding. In: Proceedings of the 11th International Conference on Learning Representations. 2023

181. Zheng Z, Deng Y, Xue D, Zhou Y, Ye F, Gu Q. Structure-informed language models are protein designers. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 1781

182. Gao Z, Tan C, Chen X, Zhang Y, Xia J, Li S, Li S Z. KW-Design: pushing the limit of protein design via knowledge refinement. In: Proceedings of the 12th International Conference on Learning Representations. 2024

183. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, et al. Highly accurate protein structure prediction with AlphaFold. Nature, 2021, 596(7873): 583−589

184. Krishna R, Wang J, Ahern W, Sturmfels P, Venkatesh P, et al. Generalized biomolecular modeling and design with RoseTTAFold All-Atom. Science, 2024, 384(6693): eadl2528

185. Jing B, Erives E, Pao-Huang P, Corso G, Berger B, Jaakkola T. EigenFold: generative protein structure prediction with diffusion models. In: Proceedings of the ICLR 2023-Machine Learning for Drug Discovery Workshop. 2023

186. Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, Smetanin N, Verkuil R, Kabeli O, Shmueli Y, Dos Santos Costa A, Fazel-Zarandi M, Sercu T, Candido S, Rives A. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science, 2023, 379(6637): 1123−1130

187. Fang X, Wang F, Liu L, He J, Lin D, Xiang Y, Zhu K, Zhang X, Wu H, Li H, Song L. A method for multiple-sequence-alignment-free protein structure prediction using a protein language model. Nature Machine Intelligence, 2023, 5(10): 1087−1096

188. Shi C, Wang C, Lu J, Zhong B, Tang J. Protein sequence and structure co-design with equivariant translation. In: Proceedings of the 11th International Conference on Learning Representations. 2023

189. Yue A, Wang Z, Xu H. ReQFlow: rectified quaternion flow for efficient and high-quality protein backbone generation. 2025, arXiv preprint arXiv: 2502.14637

190. Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, Gibbs T, Feher T, Angerer C, Steinegger M, Bhowmik D, Rost B. ProtTrans: toward understanding the language of life through self-supervised learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(10): 7112−7127

191. Chen B, Cheng X, Li P, Geng Y A, Gong J, Li S, Bei Z, Tan X, Wang B, Zeng X, Liu C, Zeng A, Dong Y, Tang J, Song L. xTrimoPGLM: unified 100B-scale pre-trained transformer for deciphering the language of protein. 2024, arXiv preprint arXiv: 2401.06199

192. Ferruz N, Schmidt S, Höcker B. ProtGPT2 is a deep unsupervised language model for protein design. Nature Communications, 2022, 13(1): 4348

193. Mansoor S, Baek M, Madan U, Horvitz E. Toward more general embeddings for protein design: harnessing joint representations of sequence and structure. bioRxiv, 2021

194. Gao B, Jia Y, Mo Y, Ni Y, Ma W Y, Ma Z M, Lan Y. Self-supervised pocket pretraining via protein fragment-surroundings alignment. In: Proceedings of the 12th International Conference on Learning Representations. 2024

195. Wang Z, Zhang Q, Hu S, Yu H, Jin X, Gong Z, Chen H. Multi-level protein structure pre-training via prompt learning. In: Proceedings of the 11th International Conference on Learning Representations. 2023

196. Gao B, Qiang B, Tan H, Ren M, Jia Y, Lu M, Liu J, Ma W Y, Lan Y. DrugCLIP: contrastive protein-molecule representation learning for virtual screening. In: Proceedings of the 37th Conference on Neural Information Processing Systems. 2023, 36

197. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, Guo D, Ott M, Zitnick C L, Ma J, Fergus R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proceedings of the National Academy of Sciences of the United States of America, 2021, 118(15): e2016239118

198. Guo Y, Wu J, Ma H, Huang J. Self-supervised pre-training for protein embeddings using tertiary structures. In: Proceedings of the 36th AAAI Conference on Artificial Intelligence. 2022, 6801−6809

199. Yuan C, Li S, Ye G, Zhang Y, Huang L K, Huang W, Liu W, Yao J, Rong Y. Annotation-guided protein design with multi-level domain alignment. 2024, arXiv preprint arXiv: 2404.16866

200. Igashov I, St¨ark H, Vignac C, Schneuing A, Satorras V G, Frossard P, Welling M, Bronstein M, Correia B. Equivariant 3d-conditional diffusion model for molecular linker design. Nature Machine Intelligence, 2024, 6(4): 417–427

201. Imrie F, Bradley A R, van der Schaar M, Deane C M. Deep generative models for 3D linker design. Journal of Chemical Information and Modeling, 2020, 60(4): 1983−1995

202. Duan C, Du Y, Jia H, Kulik H J. Accurate transition state generation with an object-aware equivariant elementary reaction diffusion model. Nature Computational Science, 2023, 3(12): 1045−1055

203. Jackson R, Zhang W, Pearson J. TSNet: predicting transition state structures with tensor field networks and transfer learning. Chemical Science, 2021, 12(29): 10022−10040

204. Gainza P, Sverrisson F, Monti F, Rodolà E, Boscaini D, Bronstein M M, Correia B E. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. Nature Methods, 2020, 17(2): 184−192

205. Kong X, Huang W, Liu Y. Generalist equivariant transformer towards 3D molecular interaction learning. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 25149−25175

206. Wang L, Liu H, Liu Y, Kurtin J, Ji S. Learning hierarchical protein representations via complete 3D graph networks. In: Proceedings of the 11th International Conference on Learning Representations. 2023

207. Zhao K, Rong Y, Jiang B, Tang J, Zhang H, Yu J X, Zhao P. Geometric graph learning for protein mutation effect prediction. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023, 3412−3422

208. Feng S, Li M, Jia Y, Ma W Y, Lan Y. Protein-ligand binding representation learning from fine-grained interactions. In: Proceedings of the 12th International Conference on Learning Representations. 2024

209. Jian Y, Wu C, Reidenbach D, Krishnapriyan A S. General binding affinity guidance for diffusion models in structure-based drug design. 2024, arXiv preprint arXiv: 2406.16821

210. Xue F, Zhang M, Li S, Gao X, Wohlschlegel J A, Huang W, Yang Y, Deng W. Se (3)-equivariant ternary complex prediction towards target protein degradation. arXiv preprint arXiv: 2502.18875, 2025

211. Stärk H, Ganea O, Pattanaik L, Barzilay D, Jaakkola T. EquiBind: geometric deep learning for drug binding structure prediction. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 20503−20521

212. Lu W, Wu Q, Zhang J, Rao J, Li C, Zheng S. TANKBind: trigonometry-aware neural networks for drug-protein binding structure

prediction. In: Proceedings of the 36th Conference on Neural Information Processing Systems. 2022

213. Long S, Zhou Y, Dai X, Zhou H. Zero-shot 3D drug design by sketching and generating. In: Proceedings of the 36th Conference on Neural Information Processing Systems. 2022, 23894−23907

214. Pei Q, Gao K, Wu L, Zhu J, Xia Y, Xie S, Qin T, He K, Liu T Y, Yan R. FABind: fast and accurate protein-ligand binding. In: Proceedings of the 37th Conference on Neural Information Processing Systems. 2023

215. Huang Y, Zhang O, Wu L, Tan C, Lin H, Gao Z, Li S, Li S Z. ReDock: towards flexible and realistic molecular docking with diffusion bridge. In: Proceedings of the 41st International Conference on Machine Learning. 2024

216. Peng X, Luo S, Guan J, Xie Q, Peng J, Ma J. Pocket2Mol: efficient molecular sampling based on 3D protein pockets. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 17644−17655

217. Lin H, Huang Y, Zhang O, Ma S, Liu M, Li X, Wu L, Wang J, Hou T, Li S Z. DiffBP: generative diffusion of 3D molecules for target protein binding. 2024, arXiv preprint arXiv: 2211.11214

218. Luo S, Guan J, Ma J, Peng J. A 3D generative model for structure-based drug design. In: Proceedings of the 35th Conference on Neural Information Processing Systems. 2021

219. Liu M, Luo Y, Uchino K, Maruhashi K, Ji S. Generating 3D molecules for target protein binding. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 13912−13924

220. Zhang Z, Min Y, Zheng S, Liu Q. Molecule generation for target protein binding with structural motifs. In: Proceedings of the 11th International Conference on Learning Representations. 2023

221. Lin H, Huang Y, Zhang O, Wu L, Li S, Chen Z, Li S Z. Functional-group-based diffusion for pocket-specific molecule generation and elaboration. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 36

222. Qiu K, Song Y, Yu J, Ma H, Cao Z, Zhang Z, Wu Y, Zheng M, Zhou H, Ma W Y. Structure-based molecule optimization via gradient-guided Bayesian update. 2024, arXiv preprint arXiv: 2411.13280

223. Pinheiro P O, Jamasb A, Mahmood O, Sresht V, Saremi S. Structure-based drug design by denoising voxel grids. In: Proceedings of the 41st International Conference on Machine Learning. 2024

224. Morehead A, Chen C, Cheng J. Geometric transformers for protein interface contact prediction. In: Proceedings of the 10th International Conference on Learning Representations. 2022

225. Sverrisson F, Feydy J, Correia B E, Bronstein M M. Fast end-to-end learning on protein surfaces. In: Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, 15267−15276

226. Townshend R J L, Bedi R, Suriana P A, Dror R O. End-to-end learning on 3D protein structure for interface prediction. In: Proceedings of the 33rd Conference on Neural Information Processing Systems. 2019, 32

227. Rodrigues C H M, Pires D E V, Ascher D B. mmCSM-PPI: predicting the effects of multiple point mutations on protein–protein interactions. Nucleic Acids Research, 2021, 49(W1): W417−W424

228. Liu X, Luo Y, Li P, Song S, Peng J. Deep geometric representations for modeling effects of mutations on protein-protein binding affinity. PLoS Computational Biology, 2021, 17(8): e1009284

229. Ganea O E, Huang X, Bunne C, Bian Y, Barzilay R, Jaakkola T S, Krause A. Independent SE(3)-equivariant models for end-to-end rigid protein docking. In: Proceedings of the 10th International Conference on Learning Representations. 2022

230. Wang Y, Shen Y, Chen S, Wang L, Fei Y, Zhou H. Learning harmonic molecular representations on riemannian manifold. In: Proceedings of the 11th International Conference on Learning Representations. 2023

231. Jin W, Barzilay R, Jaakkola T. Antibody-antigen docking and design via hierarchical structure refinement. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 10217−10227

232. Ketata M A, Laue C, Mammadov R, Stärk H, Wu M, Corso G, Marquet C, Barzilay R, Jaakkola T S. DiffDock-PP: rigid protein-protein docking with diffusion models. In: Proceedings of the ICLR 2023-Machine Learning for Drug Discovery Workshop. 2023

233. Ji Y, Bian Y, Fu G, Zhao P, Luo P. SyNDock: N rigid protein docking via learnable group synchronization. 2023, arXiv preprint arXiv: 2305.15156

234. Evans R, O'Neill M, Pritzel A, Antropova N, Senior A, et al. Protein complex prediction with alphafold-multimer. bioRxiv, 2021

235. Sverrisson F, Feydy J, Southern J, Bronstein M M, Correia B E. Physics-informed deep neural network for rigid-body protein docking. In: Proceedings of the MLDD 2022 - Machine Learning for Drug Discovery Workshop of ICLR 2022. 2022

236. Yu Z, Huang W, Liu Y. Rigid protein-protein docking via equivariant elliptic-paraboloid interface prediction. In: Proceedings of the 12th International Conference on Learning Representations. 2024

237. Wu H, Liu W, Bian Y, Wu J, Yang N, Yan J. EBMDock: neural probabilistic protein-protein docking via a differentiable energy model. In: Proceedings of the 12th International Conference on Learning Representations. 2024

238. Luo S, Su Y, Peng X, Wang S, Peng J, Ma J. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 709

239. Jin W, Wohlwend J, Barzilay R, Jaakkola T S. Iterative refinement graph neural network for antibody sequence-structure co-design. In: Proceedings of the 10th International Conference on Learning Representations. 2022

240. Gao K, Wu L, Zhu J, Peng T, Xia Y, He L, Xie S, Qin T, Liu H, He K, Liu T Y. Incorporating pre-training paradigm for antibody sequence-structure co-design. 2022, arXiv preprint arXiv: 2211.08406

241. Tan C, Gao Z, Wu L, XIA J, Zheng J, Yang X, Liu Y, Hu B, Li S Z. Cross-gate MLP with protein complex invariant embedding is a one-shot antibody designer. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 15222−15230

242. Verma Y, Heinonen M, Garg V. AbODE: ab initio antibody design using conjoined ODEs. In: Proceedings of the 40th International Conference on Machine Learning. 2023, 35037−35050

243. Martinkus K, Ludwiczak J, Cho K, Liang W C, Lafrance-Vanasse J, Hotzel I, Rajpal A, Wu Y, Bonneau R, Gligorijevic V, Loukas A. AbDiffuser: full-atom generation of in vitro functioning antibodies. In: Proceedings of the 37th Conference on Neural Information Processing Systems. 2023

244. Wu F, Zhao Y, Wu J, Jiang B, He B, Huang L, Qin C, Yang F, Huang N, Xiao Y, Wang R, Jia H, Rong Y, Liu Y, Lai H, Xu T, Liu W, Zhao P, Yao J. Fast and accurate modeling and design of antibody-antigen complex using tFold. bioRxiv, 2024

245. Lin H, Wu L, Huang Y, Liu Y, Zhang O, Zhou Y, Sun R, Li S Z. GeoAB: towards realistic antibody design and reliable affinity maturation. In: Proceedings of the 41st International Conference on Machine Learning. 2024

246. Wu L, Lin H, Huang Y, Gao Z, Tan C, Liu Y, Wu T, Li S Z. Relation-aware equivariant graph networks for epitope-unknown antibody design and specificity optimization. 2024, arXiv preprint arXiv: 2501.00013

247. Xie X, Valiente P A, Kim P M. HelixGAN a deep-learning methodology for conditional de novo design of α-helix structures.

Bioinformatics, 2023, 39(1): btad036

248. Lin H, Zhang O, Zhao H, Jiang D, Wu L, Liu Z, Huang Y, Li S Z. PPFLOW: target-aware peptide design with torsional flow matching. In: Proceedings of the 41st International Conference on Machine Learning. 2024

249. Xie T, Grossman J C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. Physical Review Letters, 2018, 120(14): 145301

250. Chen C, Ye W, Zuo Y, Zheng C, Ong S P. Graph networks as a universal machine learning framework for molecules and crystals. Chemistry of Materials, 2019, 31(9): 3564−3572

251. Choudhary K, DeCost B. Atomistic line graph neural network for improved materials property predictions. npj Computational Materials, 2021, 7(1): 185

252. Kaba S O, Ravanbakhsh S. Equivariant networks for crystal structures. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 300

253. Yan K, Liu Y, Lin Y, Ji S. Periodic graph transformers for crystal material property prediction. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1096

254. Magar R, Wang Y, Barati Farimani A. Crystal twins: self-supervised learning for crystalline material property prediction. npj Computational Materials, 2022, 8(1): 231

255. Yu H, Song Y, Hu J, Guo C, Yang B. A crystal-specific pre-training framework for crystal material property prediction. 2023, arXiv preprint arXiv: 2306.05344

256. Song Z, Meng Z, King I. A diffusion-based pre-training framework for crystal property prediction. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 8993−9001

257. Xie T, Fu X, Ganea O E, Barzilay R, Jaakkola T S. Crystal diffusion variational autoencoder for periodic material generation. In: Proceedings of the 10th International Conference on Learning Representations. 2022

258. Jiao R, Huang W, Liu Y, Zhao D, Liu Y. Space group constrained crystal generation. In: Proceedings of the 12th International Conference on Learning Representations. 2024

259. Zeni C, Pinsler R, Zügner D, Fowler A, Horton M, et al. MatterGen: a generative model for inorganic materials design. 2024, arXiv preprint arXiv: 2312.03687

260. Li Q, Jiao R, Wu L, Zhu T, Huang W, Jin S, Liu Y, Weng H, Chen X. Powder diffraction crystal structure determination using generative models. 2024, arXiv preprint arXiv: 2409.04727

261. Lin P, Chen P, Jiao R, Mo Q, Cen J, Huang W, Liu Y, Huang D, Lu Y. Equivariant diffusion for crystal structure prediction. In: Proceedings of the 41st International Conference on Machine Learning. 2024, 1204

262. Miller B K, Chen R T Q, Sriram A, Wood B M. FlowMM: generating materials with riemannian flow matching. In: Proceedings of the 41st International Conference on Machine Learning. 2024

263. Wu H, Song Y, Gong J, Cao Z, Ouyang Y, Zhang J, Zhou H, Ma W Y, Liu J. A periodic Bayesian flow for material generation. In: Proceedings of the 13th International Conference on Learning Representations. 2025

264. Zhang S, Liu Y, Xie L. Physics-aware graph neural network for accurate RNA 3D structure prediction. 2023, arXiv preprint arXiv: 2210.16392

265. Li Z, Cen J, Huang W, Wang T, Song L. Size-generalizable RNA structure evaluation by exploring hierarchical geometries. In: Proceedings of the 13th International Conference on Learning Representations. 2025

266. Greff K, Belletti F, Beyer L, Doersch C, Du Y, et al. Kubric: a scalable dataset generator. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 3739–3751

267. Bear D, Wang E, Mrowca D, Binder F J, Tung H Y, Pramod R T, Holdaway C, Tao S, Smith K A, Sun F Y, Li F F, Kanwisher N, Tenenbaum J, Yamins D, Fan J E. Physion: evaluating physical prediction from vision in humans and machines. In: Proceedings of the 1st Neural Information Processing Systems Track on Datasets and Benchmarks. 2021

268. Yu K T, Bauza M, Fazeli N, Rodriguez A. More than a million ways to be pushed. A high-fidelity experimental dataset of planar pushing. In: Proceedings of 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2016, 30−37

269. Townshend R J L, Vogele M, Suriana P, Derry A, Powers A S, Laloudakis Y, Balachandar S, Jing B, Anderson B M, Eismann S, Kondor R, Altman R B, Dror R O. ATOM3D: tasks on molecules in three dimensions. In: Proceedings of the 35th Conference on Neural Information Processing Systems. 2021

270. Xu M, Luo S, Bengio Y, Peng J, Tang J. Learning neural generative dynamics for molecular conformation generation. In: Proceedings of the 9th International Conference on Learning Representations. 2021

271. Chmiela S, Tkatchenko A, Sauceda H E, Poltavsky I, Schütt K T, Müller K R. Machine learning of accurate energy-conserving molecular force fields. Science Advances, 2017, 3(5): e1603015

272. Tran R, Lan J, Shuaibi M, Wood B M, Goyal S, Das A, Heras-Domingo J, Kolluru A, Rizvi A, Shoghi N, Sriram A, Therrien F, Abed J, Voznyy O, Sargent E H, Ulissi Z, Zitnick C L. The open catalyst 2022 (OC22) dataset and challenges for oxide electrocatalysts. ACS Catalysis, 2023, 13(5): 3066−3084

273. Seyler S, Beckstein O. Molecular dynamics trajectory for benchmarking MDanalysis. 2017

274. Lindorff-Larsen K, Piana S, Dror R O, Shaw D E. How fast-folding proteins fold. Science, 2011, 334(6055): 517−520

275. Axelrod S, Gómez-Bombarelli R. GEOM, energy-annotated molecular conformations for property prediction and molecular generation. Scientific Data, 2022, 9(1): 185

276. Wang X, Zhao H, Tu W W, Yao Q. Automated 3D pre-training for molecular property prediction. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 2419−2430

277. Isert C, Atz K, Jimenez-Luna J, Schneider G. QMugs, quantum mechanical properties of drug-like molecules. Scientific Data, 2022, 9(1): 273

278. Ashburner M, Ball C A, Blake J A, Botstein D, Butler H, Cherry J M, Davis A P, Dolinski K, Dwight S S, Eppig J T, Harris M A, Hill D P, Issel-Tarver L, Kasarskis A, Lewis S, Matese J C, Richardson J E, Ringwald M, Rubin G M, Sherlock G. Gene ontology: tool for the unification of biology. Nature Genetics, 2000, 25(1): 25−29

279. Bairoch A. The ENZYME database in 2000. Nucleic Acids Research, 2000, 28(1): 304−305

280. Orengo C A, Michie A D, Jones S, Jones D T, Swindells M B, Thornton J M. CATH–a hierarchic classification of protein domain structures. Structure, 1997, 5(8): 1093−1109

281. Xue Y, Liu Z, Fang X, Wang F. Multimodal pre-training model for sequence-based prediction of protein-protein interaction. In: Proceedings of the 16th Machine Learning in Computational Biology Meeting. 2022, 34−46

282. Chandonia J M, Fox N K, Brenner S E. SCOPe: classification of large macromolecular structures in the structural classification of proteins—extended database. Nucleic Acids Research, 2019, 47(D1): D475−D481

283. Heinzinger M, Weissenow K, Sanchez J G, Henkel A, Steinegger M, Rost B. ProstT5: bilingual language model for protein sequence and

structure. bioRxiv, 2023

284. Bepler T, Berger B. Learning the protein language: evolution, structure, and function. Cell Systems, 2021, 12(6): 654−669

285. Rao R, Bhattacharya N, Thomas N, Duan Y, Chen P, Canny J, Abbeel P, Song Y S. Evaluating protein transfer learning with TAPE. In: Proceedings of the 33rd Conference on Neural Information Processing Systems. 2019, 32

286. Varadi M, Anyango S, Deshpande M, Nair S, Natassia C, et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Research, 2022, 50(D1): D439−D444

287. Gao Z, Tan C, Li S Z. AlphaDesign: a graph protein design method and benchmark on AlphaFoldDB. 2022, arXiv preprint arXiv: 2202.01079

288. Consortium T U. UniProt: the universal protein knowledgebase in 2023. Nucleic Acids Research, 2023, 51(D1): D523−D531

289. Almagro Armenteros J J, Sønderby C K, Sønderby S K, Nielsen H, Winther O. DeepLoc: prediction of protein subcellular localization using deep learning. Bioinformatics, 2017, 33(21): 3387−3395

290. Steinegger M, Söding J. Clustering huge protein sequence sets in linear time. Nature Communications, 2018, 9(1): 2542

291. Klausen M S, Jespersen M C, Nielsen H, Jensen K K, Jurtz V I, Sønderby C K, Sommer M O A, Winther O, Nielsen M, Petersen B, Marcatili P. NetSurfP-2. 0: improved prediction of protein structural features by integrated deep learning. Proteins: Structure, Function, and Bioinformatics, 2019, 87(6): 520−527

292. Xu M, Zhang Z, Lu J, Zhu Z, Zhang Y, Chang M, Liu R, Tang J. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 2548

293. Kryshtafovych A, Schwede T, Topf M, Fidelis K, Moult J. Critical assessment of methods of protein structure prediction (CASP)—round XIII. Proteins: Structure, Function, and Bioinformatics, 2019, 87(12): 1011−1020

294. Berman H M, Westbrook J, Feng Z, Gilliland G, Bhat T N, Weissig H, Shindyalov I N, Bourne P E. The protein data bank. Nucleic Acids Research, 2000, 28(1): 235−242

295. Sterling T, Irwin J J. ZINC 15 – ligand discovery for everyone. Journal of Chemical Information and Modeling, 2015, 55(11): 2324−2337

296. Su M, Yang Q, Du Y, Feng G, Liu Z, Li Y, Wang R. Comparative assessment of scoring functions: the CASF-2016 update. Journal of Chemical Information and Modeling, 2019, 59(2): 895−913

297. Schreiner M, Bhowmik A, Vegge T, Busk J, Winther O. Transition1x-a dataset for building generalizable reactive machine learning potentials. Scientific Data, 2022, 9(1): 779

298. Francoeur P G, Masuda T, Sunseri J, Jia A, Iovanisci R B, Snyder I, Koes D R. Three-dimensional convolutional neural networks and a cross-docked data set for structure-based drug design. Journal of Chemical Information and Modeling, 2020, 60(9): 4200−4215

299. Morehead A, Chen C, Sedova A, Cheng J. Dips-plus: the enhanced database of interacting protein structures for interface prediction. Scientific Data, 2023, 10(1): 509

300. Stark C, Breitkreutz B J, Reguly T, Boucher L, Breitkreutz A, Tyers M. BioGRID: a general repository for interaction datasets. Nucleic Acids Research, 2006, 34(S1): D535−D539

301. Hallee L, Gleghorn J P. Protein-protein interaction prediction is achievable with large language models. bioRxiv, 2023

302. Vreven T, Moal I H, Vangone A, Pierce B G, Kastritis P L, Torchala M, Chaleil R, Jiménez-García B, Bates P A, Fernandez-Recio J, Bonvin A M J J, Weng Z. Updates to the integrated protein– protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. Journal of Molecular Biology, 2015, 427(19): 3031−3041

303. Jankauskaitė J, Jiménez-García B, Dapkūnas J, Fernández-Recio J, Moal I H. SKEMPI 2. 0: an updated benchmark of changes in protein–protein binding energy, kinetics and thermodynamics upon mutation. Bioinformatics, 2019, 35(3): 462−469

304. Raybould M I J, Kovaltsuk A, Marks C, Deane C M. CoV-AbDab: the coronavirus antibody database. Bioinformatics, 2021, 37(5): 734−735

305. Wen Z, He J, Tao H, Huang S Y. PepBDB: a comprehensive structural database of biological peptide–protein interactions. Bioinformatics, 2019, 35(1): 175−177

306. Lei Y, Li S, Liu Z, Wan F, Tian T, Li S, Zhao D, Zeng J. A deep-learning framework for multi-level peptide–protein interaction prediction. Nature Communications, 2021, 12(1): 5465

307. Tsaban T, Varga J K, Avraham O, Ben-Aharon Z, Khramushin A, Schueler-Furman O. Harnessing protein folding neural networks for peptide–protein docking. Nature Communications, 2022, 13(1): 176

308. Jain A, Ong S P, Hautier G, Chen W, Richards W D, Dacek S, Cholia S, Gunter D, Skinner D, Ceder G, Persson K A. Commentary: the materials project: a materials genome approach to accelerating materials innovation. APL Materials, 2013, 1(1): 011002

309. Castelli I E, Landis D D, Thygesen K S, Dahl S, Chorkendorff I, Jaramillo T F, Jacobsen K W. New cubic perovskites for one- and two-photonwater splitting using the computational materials repository. Energy and Environmental Science, 2012, 5(10): 9034−9043

310. Castelli I E, Olsen T, Datta S, Landis D D, Dahl S, Thygesen K S, Jacobsen K W. Computational screening of perovskite metal oxides for optimal solar light capture. Energy and Environmental Science, 2012, 5(2): 5814−5819

311. Pickard C J. AIRSS data for carbon at 10GPa and the C+N+H+O system at 1GPa. 2020

312. Choudhary K, Garrity K F, Reid A C E, DeCost B, Biacchi A J, et al. The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. npj Computational Materials, 2020, 6(1): 173

313. Choudhary K, DeCost B, Tavazza F. Machine learning with force-field-inspired descriptors for materials: fast screening and mapping energy landscape. Physical Review Materials, 2018, 2(8): 083801

314. Watkins A M, Rangan R, Das R. FARFAR2: improved *de novo* Rosetta prediction of complex global RNA folds. Structure, 2020, 28(8): 963−976.e6

315. Liu Y, Cheng J, Zhao H, Xu T, Zhao P, Tsung F, Li J, Rong Y. SEGNO: generalizing equivariant graph neural networks with physical inductive biases. In: Proceedings of the 12th International Conference on Learning Representations. 2024

316. Downs G M, Gillet V J, Holliday J D, Lynch M F. Review of ring perception algorithms for chemical graphs. Journal of Chemical Information and Computer Sciences, 1989, 29(3): 172−187

317. Lipinski C A, Lombardo F, Dominy B W, Feeney P J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Advanced Drug Delivery Reviews, 2012, 64 Suppl 1: 4−17

318. Gowers R J, Linke M, Barnoud J, Reddy T J E, Melo M N, Seyler S L, Domanski J J, Dotson D L, Buchoux S, Kenney I M, Beckstein O. MDAnalysis: a python package for the rapid analysis of molecular dynamics simulations. In: Proceedings of the 15th Python in Science Conference. 2016, 105

319. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention. 2015, 234−241

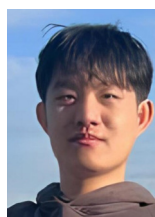320. Huang C W, Dinh L, Courville A. Augmented normalizing flows:

bridging the gap between generative flows and latent variable models. 2020, arXiv preprint arXiv: 2002.07101

321. Liberti L, Lavor C, Maculan N, Mucherino A. Euclidean distance geometry and applications. SIAM Review, 2014, 56(1): 3−69

322. Kingma D P, Welling M. Auto-encoding variational Bayes. In: Proceedings of the 2nd International Conference on Learning Representations. 2014, 1050

323. Wang L, Song C, Liu Z, Rong Y, Liu Q, Wu S, Wang L. Diffusion models for molecules: a survey of methods and tasks. 2025, arXiv preprint arXiv: 2502.09511

324. Wang S, Guo Y, Wang Y, Sun H, Huang J. SMILES-BERT: large scale unsupervised pre-training for molecular property prediction. In: Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. 2019, 429−436

325. Hu W, Liu B, Gomes J, Zitnik M, Liang P, Pande V S, Leskovec J. Strategies for pre-training graph neural networks. In: Proceedings of the 8th International Conference on Learning Representations. 2020

326. Rong Y, Bian Y, Xu T, Xie W, Wei Y, Huang W, Huang J. Self-supervised graph transformer on large-scale molecular data. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 1053

327. Hu W, Fey M, Zitnik M, Dong Y, Ren H, Liu B, Catasta M, Leskovec J. Open graph benchmark: datasets for machine learning on graphs. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 1855

328. Nakata M, Shimazaki T. PubChemQC project: a large-scale first-principles electronic structure database for data-driven chemistry. Journal of Chemical Information and Modeling, 2017, 57(6): 1300−1308

329. Pracht P, Bohle F, Grimme S. Automated exploration of the low-energy chemical space with fast quantum chemical methods. Physical Chemistry Chemical Physics, 2020, 22(14): 7169−7192

330. Hung M C, Link W. Protein localization in disease and therapy. Journal of Cell Science, 2011, 124(20): 3381−3392

331. Dallago C, Mou J, Johnston K E, Wittmann B J, Bhattacharya N, Goldman S, Madani A, Yang K K. FLIP: benchmark tasks in fitness landscape inference for proteins. bioRxiv, 2021

332. Krivák R, Hoksza D. Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features. Journal of Cheminformatics, 2015, 7: 12

333. Le Guilloux V, Schmidtke P, Tuffery P. Fpocket: an open source platform for ligand pocket detection. BMC Bioinformatics, 2009, 10: 168

334. Jiménez J, Doerr S, Martínez-Rosell G, Rose A S, De Fabritiis G. DeepSite: protein-binding site predictor using 3D-convolutional neural networks. Bioinformatics, 2017, 33(19): 3036−3042

335. Mylonas S K, Axenopoulos A, Daras P. DeepSurf: a surface-based deep learning approach for the prediction of ligand binding sites on proteins. Bioinformatics, 2021, 37(12): 1681−1690

336. Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, dos Santos Costa A, Fazel-Zarandi M, Sercu T, Candido S, Rives A. Language models of protein sequences at the scale of evolution enable accurate structure prediction. bioRxiv, 2022

337. Suzek B E, Wang Y, Huang H, McGarvey P B, Wu C H, Consortium U. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics, 2015, 31(6): 926−932

338. Rao R, Meier J, Sercu T, Ovchinnikov S, Rives A. Transformer protein language models are unsupervised structure learners. In: Proceedings of the 9th International Conference on Learning Representations. 2021

339. Wu L, Huang Y, Lin H, Li S Z. A survey on protein representation

340. Hussain J, Rea C. Computationally efficient algorithm to identify matched molecular pairs (MMPs) in large data sets. Journal of Chemical Information and Modeling, 2010, 50(3): 339−348

341. Lin H, Huang Y, Zhang O, Wu L, Li S, Chen Z, Li S Z. Functional-group-based diffusion for pocket-specific molecule generation and elaboration. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 1504

342. Wang R, Fang X, Lu Y, Wang S. The PDBbind database: collection of binding affinities for protein-ligand complexes with known three-dimensional structures. Journal of Medicinal Chemistry, 2004, 47(12): 2977−2980

343. Kastritis P L, Moal I H, Hwang H, Weng Z, Bates P A, Bonvin A M J J, Janin J. A structure-based benchmark for protein–protein binding affinity. Protein Science, 2011, 20(3): 482−491

344. Moal I H, Fernández-Recio J. SKEMPI: a structural kinetic and energetic database of mutant protein interactions and its use in empirical models. Bioinformatics, 2012, 28(20): 2600−2607

345. Fosgerau K, Hoffmann T. Peptide therapeutics: current status and future directions. Drug Discovery Today, 2015, 20(1): 122−128

346. Lee A C L, Harris J L, Khanna K K, Hong J H. A comprehensive review on current advances in peptide drug development and design. International Journal of Molecular Sciences, 2019, 20(10): 2383

347. Bhardwaj G, Mulligan V K, Bahl C D, Gilmore J M, Harvey P J, et al. Accurate *de novo* design of hyperstable constrained peptides. Nature, 2016, 538(7625): 329−335

348. Cao L, Coventry B, Goreshnik I, Huang B, Sheffler W, et al. Design of protein-binding proteins from the target structure alone. Nature, 2022, 605(7910): 551−560

349. Zbontar J, Jing L, Misra I, LeCun Y, Deny S. Barlow twins: self-supervised learning via redundancy reduction. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 12310−12320

350. Chen X, He K. Exploring simple Siamese representation learning. In: Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021, 15745−15753

351. Geiger M, Smidt T. e3nn: Euclidean neural networks. 2022, arXiv preprint arXiv: 2207.09453

352. Das R, Baker D. Automated *de novo* prediction of native-like RNA tertiary structures. Proceedings of the National Academy of Sciences of the United States of America, 2007, 104(37): 14664−14669

353. Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. 2018

354. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language models are unsupervised multitask learners. OpenAI Blog, 2019, 1(8): 9

355. Brown T B, Mann B, Ryder N, Subbiah M, Kaplan J, et al. Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. 2020, 159

356. Reed S, Zolna K, Parisotto E, Colmenarejo S G, Novikov A, Barth-Maron G, Giménez M, Sulsky Y, Kay J, Springenberg J T, Eccles T, Bruce J, Razavi A, Edwards A, Heess N, Chen Y, Hadsell R, Vinyals O, Bordbar M, de Freitas N. A generalist agent. Transactions on Machine Learning Research, 2022, See openreview.net/forum?id= 1ikK0kHjvj website, 2022

357. Merchant A, Batzner S, Schoenholz S S, Aykol M, Cheon G, Cubuk E D. Scaling deep learning for materials discovery. Nature, 2023, 624(7990): 80−85

358. Bran A M, Cox S, Schilter O, Baldassari C, White A D, Schwaller P. Augmenting large language models with chemistry tools. Nature Machine Intelligence, 2024, 6(5): 525−535

359. Liu X, Yu H, Zhang H, Xu Y, Lei X, et al. AgentBench: evaluating LLMs as agents. In: Proceedings of the 12th International Conference on Learning Representations. 2024

360. Janakarajan N, Erdmann T, Swaminathan S, Laino T, Born J. Language models in molecular discovery. In: Satoh H, Funatsu K, Yamamoto H, eds. Drug Development Supported by Informatics. Singapore: Springer, 2024, 121−141

361. Liu S, Wang J, Yang Y, Wang C, Liu L, Guo H, Xiao C. Conversational drug editing using retrieval and domain feedback. In: Proceedings of the 12th International Conference on Learning Representations. 2024

362. Zhang W, Wang X, Nie W, Eaton J, Rees B, Gu Q. MoleculeGPT: instruction following large language models for molecular property prediction. In: Proceedings of NeurIPS 2023 Workshop on New Frontiers of AI for Drug Discovery and Development. 2023

363. Zheng Z, Liu Y, Li J, Yao J, Rong Y. Relaxing continuous constraints of equivariant graph neural networks for physical dynamics learning. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024

364. Liu Y, Zheng Z, Rong Y, Li J. Equivariant graph learning for high-density crowd trajectories modeling. Transactions on Machine Learning Research, See openreview.net/forum?id=TeQRze2ZjO website, 2024

Jiaqi HAN is a PhD student in Computer Science at Stanford University, USA. Previously, he received his BE in Computer Science at Tsinghua University, China. His research interest involves developing principled machine learning methods for modeling geometric systems.

Jiacheng CEN is currently a PhD student in the Gaoling School of Artificial Intelligence, Renmin University of China. His research interests mainly concern geometric learning theory and its applications in scientific settings.

Liming WU is currently pursuing his PhD in the Gaoling School of Artificial Intelligence at Renmin University of China. His research interests lie in geometric deep learning and AI for Science.

Zongzhao LI is currently a PhD student in the Gaoling School of Artificial Intelligence, Renmin University of China, China. His research interests lie in geometric deep learning and its applications in AI for Science problems.
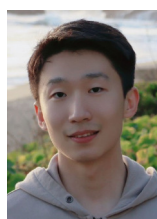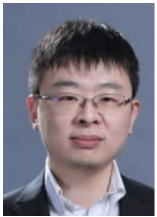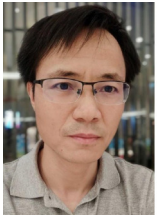

Xiangzhe KONG received his BE in Computer Science from Tsinghua University, China in 2022. He is currently a PhD student in the Department of Computer Science and Technology at Tsinghua University, China.

Rui JIAO is a PhD student in the Department of Computer Science and Technology at Tsinghua University, China. Prior to that, he received the BE degree in Computer Science at Tsinghua University, China. His research interest focuses on geometric machine learning and material design.

Ziyang YU received his BE in Computer Science from Tsinghua University, China in 2024. He is currently a DEng student in the Department of Computer Science and Technology at Tsinghua University, China.

Tingyang XU is a Senior Research Scientist at Alibaba DAMO Academy's Language & Science Lab. He earned his bachelor's degree from Shanghai Jiao Tong University, China and his master's and PhD in Computer Science from the University of Connecticut, USA under Professor Jinbo Bi. His research focuses on deep learning and its scientific applications, particularly geometric graph neural networks for molecular dynamics, drug design, and AI for science. He has developed SOTA methods for deep geometric graph learning, advancing tasks like molecular generation and molecular property prediction.

Fandi WU is a Senior Research Scientist at Tencent Life Science Lab. He earned his bachelor's degree from University of Science and Technology of China, China and his PhD in Computer Science from the Institute of Computing Technology, Chinese Academy of Sciences. His research focuses on deep learning and its scientific applications, particularly protein structure prediction and protein design.

Zihe WANG received his BE in Computer Science from Tsinghua University, China in 2011 and his PhD in Computer Science from Tsinghua University, China in 2016. He is currently an assistant professor at Renmin University of China. His research focuses on algorithms and mechanism design.

Hongteng XU is an associate professor in the Gaoling School of Artificial Intelligence, Renmin University of China, China. From 2018 to 2020, he was a senior research scientist in Infinia ML Inc. In the same time period, he is a visiting faculty member in the Department of Electrical and Computer Engineering, Duke University, USA. He received his Ph.D. from the School of Electrical and Computer Engineering at Georgia Institute of Technology, USA in 2017. His research interests include machine learning and its applications, especially optimal transport theory, sequential data modeling and analysis, deep learning techniques, and their applications in computer vision and data mining.

Zhewei WEI received his PhD of Computer Science and Engineering from Hong Kong University of Science and Technology, China. He did postdoctoral research in Aarhus University, USA from 2012 to 2014, and joined Renmin University of China, China in 2014.

Deli ZHAO is leading the AI team in Alibaba DAMO Academy. He has researched on computer vision and machine learning for nearly two decades, now mainly focusing on generative models, multi-modal learning, and foundation models.

Yang LIU is the GDS Professor in the Department of Computer Science and Technology at Tsinghua University, China. He is Executive Dean of Institute for AI Industry Research (AIR) and Associate Dean of the Department of Computer Science and Technology. His research interests include artificial intelligence, natural language processing, and Medical AI.
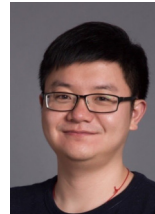
Yu RONG is an IEEE Senior Member and is recognized as a high-level overseas talent by Shenzhen. In June 2017, he joined Tencent AI Lab as a principal researcher and transitioned to Alibaba DAMO Academy in June 2024, focusing on large language models and AI for Science. His research interests lie in graph deep learning and large language models, particularly applied within the AI for Science domain.

Wenbing HUANG is now a tenure-track associate professor at Gaoling School of Artificial Intelligence (GSAI), Renmin University of China, China. Before joining GSAI, he worked as an assistant researcher at AIR, Tsinghua University and senior researcher at Tencent AI Lab. His research focuses on geometric deep learning, GNN and AI for Science.