Applications of Deep Learning in Biomedicine

Johannes Lutzeyer

Advanced Deep Learning: Lecture 9

March 06, 2025

Overview of Today's Lecture

1) Geometric Graph Neural Networks;

Overview of Today's Lecture

1) Geometric Graph Neural Networks;
2) Protein-Ligand Docking;

Overview of Today's Lecture

1) Geometric Graph Neural Networks;
2) Protein-Ligand Docking;
3) Protein-Property Prediction: Case Study of Antibiotic Resistance Classification.

Geometric Graphs (Duval et al., 2023)

Geometric Graphs (Duval et al., 2023)

Definition: Geometric Graphs

A geometric graph $G = (A, S, V, X)$ consists of an adjacency matrix A, node features split into two categories, scalar features S and vector features V, and embeddings of the nodes in d-dimensional Euclidean space X. In most biomedical applications $d = 3$.

Geometric Graphs (Duval et al., 2023)

Definition: Geometric Graphs

A geometric graph $G = (A, S, V, X)$ consists of an adjacency matrix A, node features split into two categories, scalar features S and vector features V, and embeddings of the nodes in d-dimensional Euclidean space X. In most biomedical applications $d = 3$.

Relevant Symmetries: Permutation and Euclidean symmetries.

Geometric Graphs (Duval et al., 2023)

Definition: Geometric Graphs

A geometric graph $G = (A, S, V, X)$ consists of an adjacency matrix A, node features split into two categories, scalar features S and vector features V, and embeddings of the nodes in d-dimensional Euclidean space X. In most biomedical applications $d = 3$.

Relevant Symmetries: Permutation and Euclidean symmetries.

- permutation P of the node labels have the following action on a geometric graph (PAP^T, PS, PV, PX).

Geometric Graphs (Duval et al., 2023)

Definition: Geometric Graphs

A geometric graph $G = (A, S, V, X)$ consists of an adjacency matrix A, node features split into two categories, scalar features S and vector features V, and embeddings of the nodes in d-dimensional Euclidean space X. In most biomedical applications $d = 3$.

Relevant Symmetries: Permutation and Euclidean symmetries.

- permutation P of the node labels have the following action on a geometric graph (PAP^T, PS, PV, PX).
- rotations $R \in SO(d)$ of the d-dimensional embeddings act on geometric graph as (A, S, VR, XR).

Geometric Graphs (Duval et al., 2023)

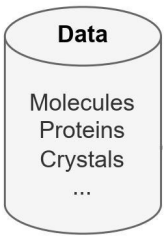
Definition: Geometric Graphs

A geometric graph $G = (A, S, V, X)$ consists of an adjacency matrix A, node features split into two categories, scalar features S and vector features V, and embeddings of the nodes in d-dimensional Euclidean space X. In most biomedical applications $d = 3$.

Relevant Symmetries: Permutation and Euclidean symmetries.

- permutation P of the node labels have the following action on a geometric graph (PAP^T, PS, PV, PX).
- rotations $R \in SO(d)$ of the d-dimensional embeddings act on geometric graph as (A, S, VR, XR).
- translations $T \in \mathbb{T}$ of the d-dimensional embeddings act on geometric graph as $(A, S, V, X + T)$.

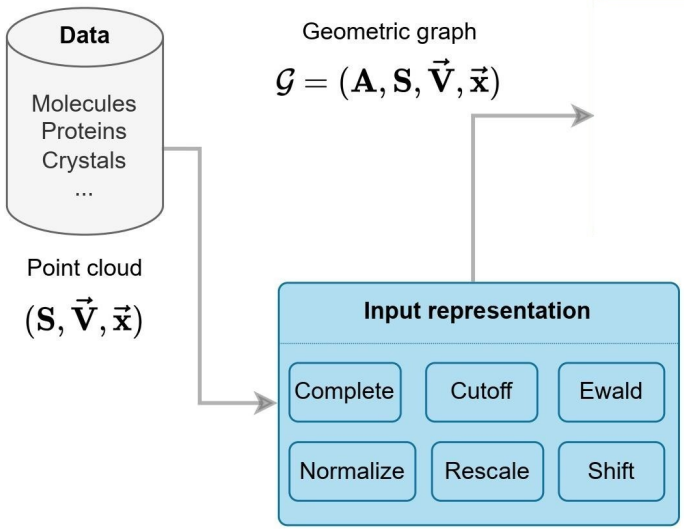
Geometric Graph Neural Networks (Duval et al., 2023)



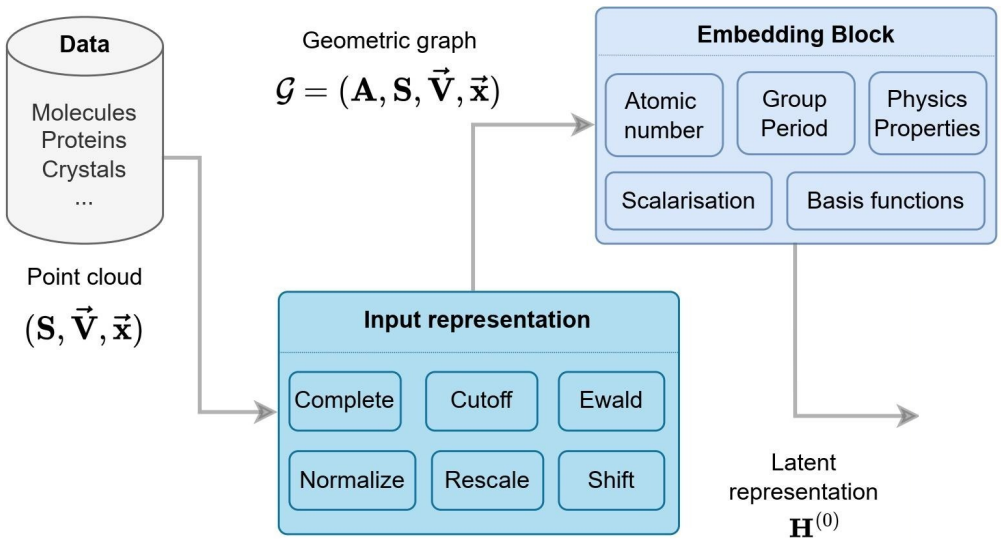
Point cloud

$(\mathbf{S}, \vec{\mathbf{V}}, \vec{\mathbf{x}})$

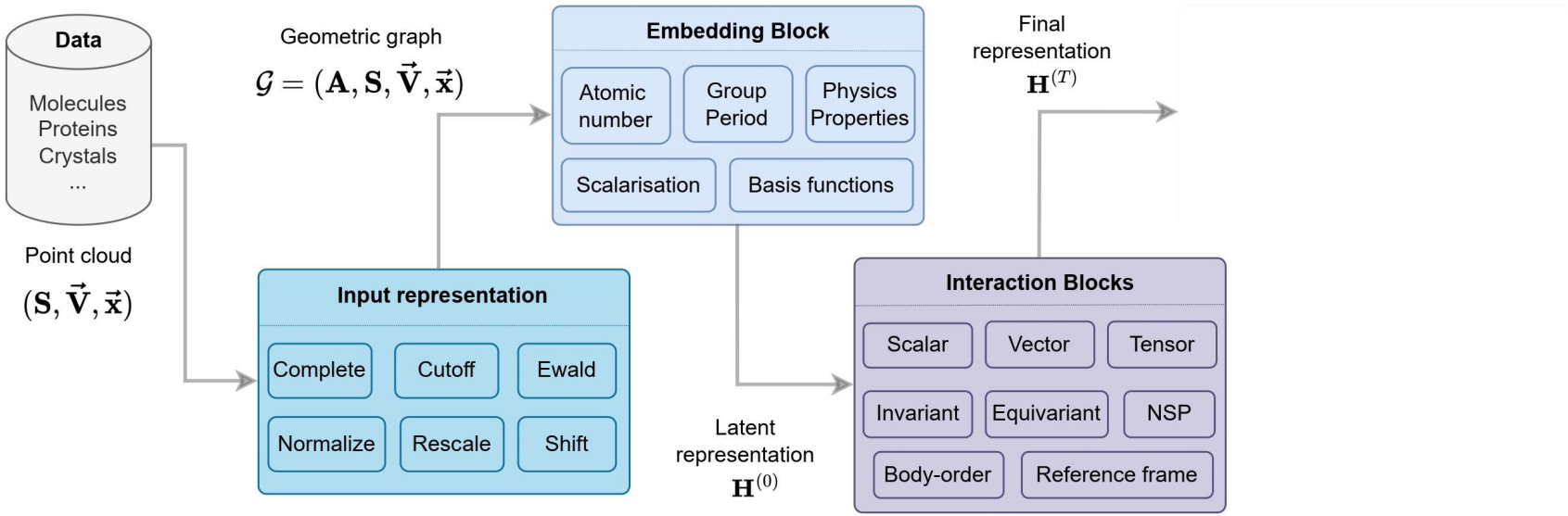
Geometric Graph Neural Networks (Duval et al., 2023)



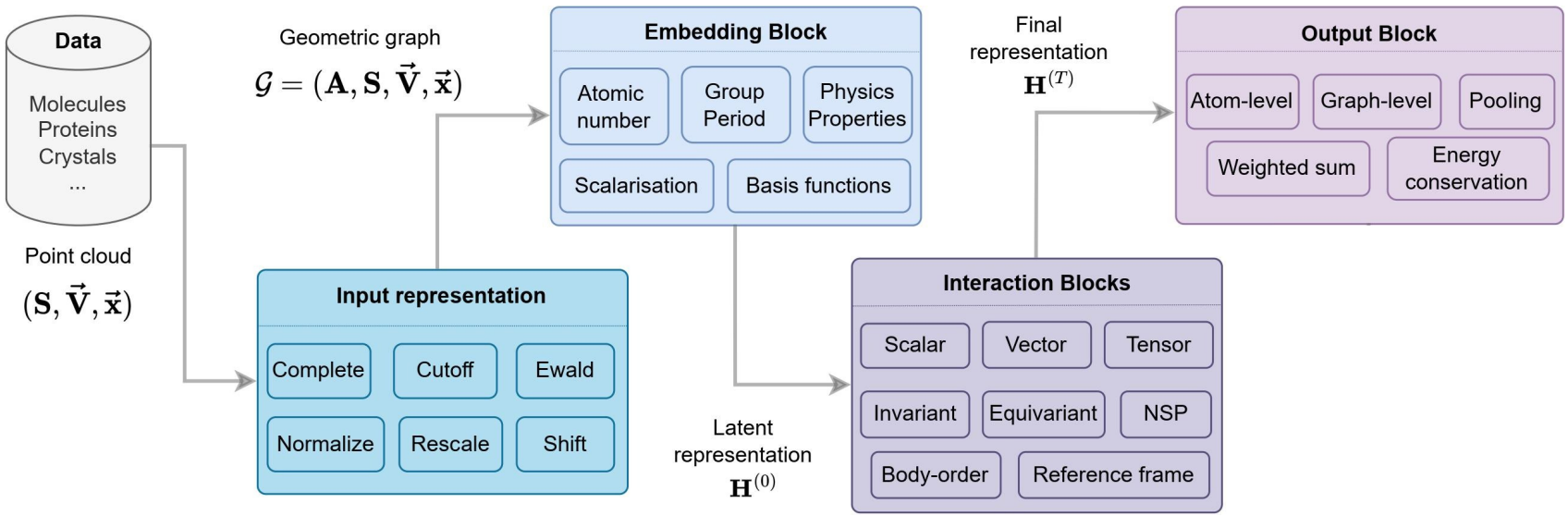
Geometric Graph Neural Networks (Duval et al., 2023)



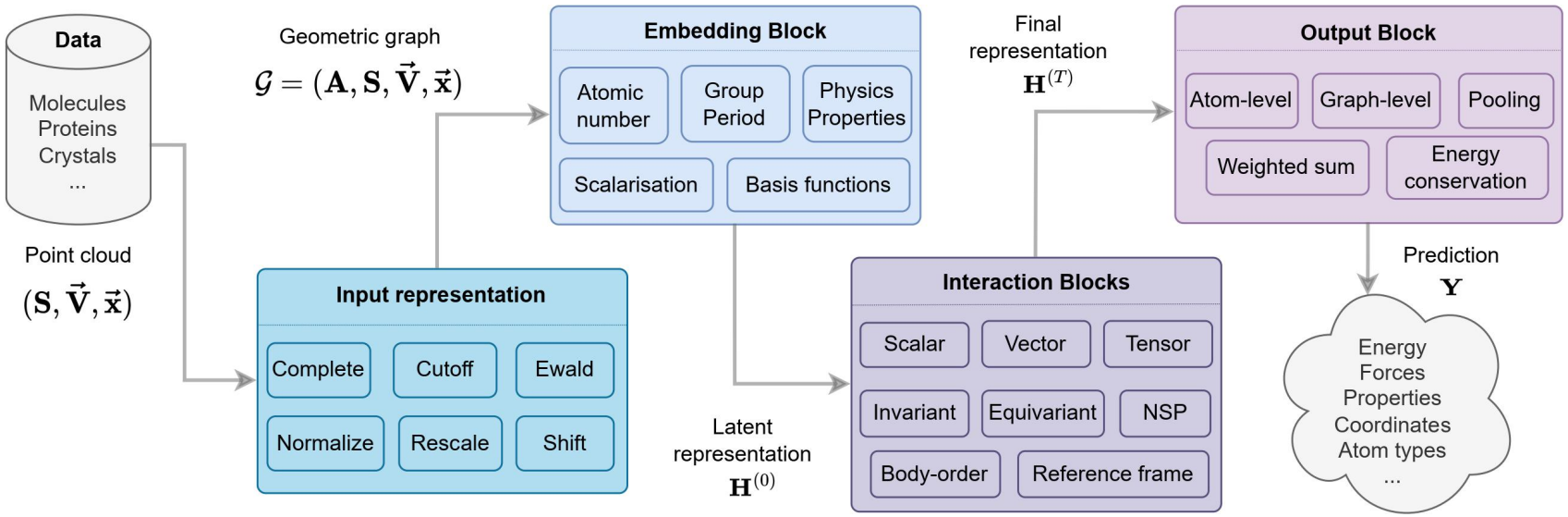
Geometric Graph Neural Networks (Duval et al., 2023)



Geometric Graph Neural Networks (Duval et al., 2023)



Geometric Graph Neural Networks (Duval et al., 2023)



Invariant Geometric Graph Neural Networks

General Idea (Duval et al., 2023)

Invariant GNNs leverage 3D geometric information by **pre-computing informative scalar quantities** between atoms, such as pairwise distances, triplet-wise angles, and quadruplet-wise torsion angles, and using learned latent representations of these quantities during message passing. Since these input scalar quantities are invariant to Euclidean transformations, the intermediate representations and predictions of these models are guaranteed to be invariant.

Invariant Geometric Graph Neural Networks

General Idea (Duval et al., 2023)

Invariant GNNs leverage 3D geometric information by **pre-computing informative scalar quantities** between atoms, such as pairwise distances, triplet-wise angles, and quadruplet-wise torsion angles, and using learned latent representations of these quantities during message passing. Since these input scalar quantities are invariant to Euclidean transformations, the intermediate representations and predictions of these models are guaranteed to be invariant.

Example: the interaction layer of SchNet (Schütt et al., 2018) takes the following form,

$$s_i^{(t+1)} = s_i^{(t)} + \sum_{j \in \mathcal{N}_i} s_j^{(t)} \odot W^{(t)}(\|x_i - x\|),$$

where $\odot$ denotes the element-wise product and $W^{(t)}$ is a trainable weight matrix.

Invariant Geometric Graph Neural Networks

General Idea (Duval et al., 2023)

Invariant GNNs leverage 3D geometric information by **pre-computing informative scalar quantities** between atoms, such as pairwise distances, triplet-wise angles, and quadruplet-wise torsion angles, and using learned latent representations of these quantities during message passing. Since these input scalar quantities are invariant to Euclidean transformations, the intermediate representations and predictions of these models are guaranteed to be invariant.

Example: the interaction layer of SchNet (Schütt et al., 2018) takes the following form,

$$s_i^{(t+1)} = s_i^{(t)} + \sum_{j \in \mathcal{N}_i} s_j^{(t)} \odot W^{(t)}(\|x_i - x\|),$$

where $\odot$ denotes the element-wise product and $W^{(t)}$ is a trainable weight matrix.

- DimeNet (Gasteiger et al., 2020) extends this concept by considering bond angles in addition to distance,

Invariant Geometric Graph Neural Networks

General Idea (Duval et al., 2023)

Invariant GNNs leverage 3D geometric information by **pre-computing informative scalar quantities** between atoms, such as pairwise distances, triplet-wise angles, and quadruplet-wise torsion angles, and using learned latent representations of these quantities during message passing. Since these input scalar quantities are invariant to Euclidean transformations, the intermediate representations and predictions of these models are guaranteed to be invariant.

Example: the interaction layer of SchNet (Schütt et al., 2018) takes the following form,

$$s_i^{(t+1)} = s_i^{(t)} + \sum_{j \in \mathcal{N}_i} s_j^{(t)} \odot W^{(t)}(\|x_i - x\|),$$

where $\odot$ denotes the element-wise product and $W^{(t)}$ is a trainable weight matrix.

- DimeNet (Gasteiger et al., 2020) extends this concept by considering bond angles in addition to distance,
- GemNet (Gasteiger et al., 2021) presents a further extension by also considering torsion angles of bonds.

Background: Proteins and Molecules

Definition: Proteins & Molecules

A **protein** is a sequence of amino acids, where each amino acid consists of several atoms. In a given environment a protein folds into a given three-dimensional shape that strongly impacts, if not fully determines, the protein's chemical function.

Background: Proteins and Molecules

Definition: Proteins & Molecules

A **protein** is a sequence of amino acids, where each amino acid consists of several atoms. In a given environment a protein folds into a given three-dimensional shape that strongly impacts, if not fully determines, the protein's chemical function.
A **molecule** is a graph of atoms that enter different chemical bonds. Molecules can be encoded as sequences using the "Simplified Molecular Input Line Entry System (SMILES)" encoding.

Background: Proteins and Molecules

Definition: Proteins & Molecules

A **protein** is a sequence of amino acids, where each amino acid consists of several atoms. In a given environment a protein folds into a given three-dimensional shape that strongly impacts, if not fully determines, the protein's chemical function.
A **molecule** is a graph of atoms that enter different chemical bonds. Molecules can be encoded as sequences using the "Simplified Molecular Input Line Entry System (SMILES)" encoding.

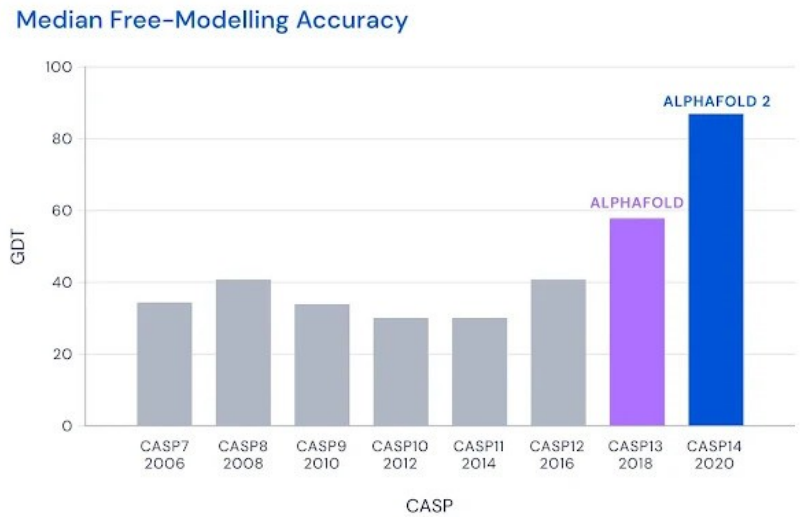
The optimal representation of protein and molecule data (sequences, point clouds, graphs or something is else) is tasks and context-dependent and still subject to active research.

Background: AlphaFold2

Deepmind's AlphaFold2 model (Jumper et al., 2021) *predicts the folded representation, i.e., 3-D positions of all aminoacids*, for any protein sequence.

Background: AlphaFold2

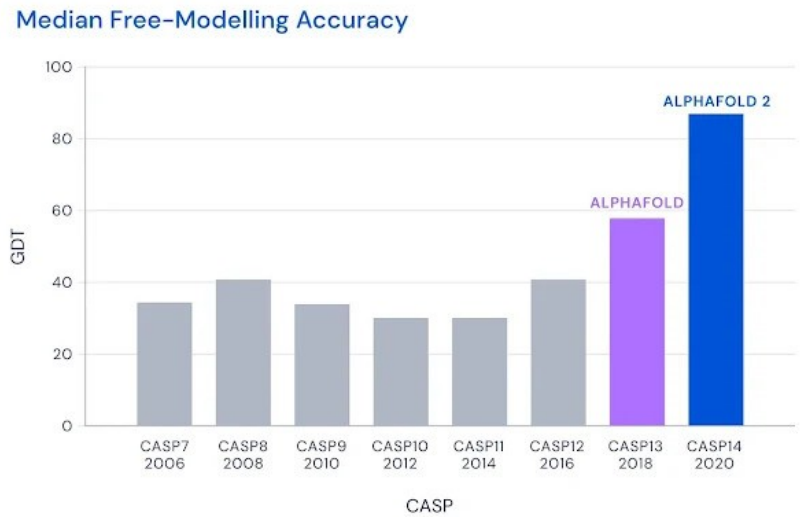
Deepmind's AlphaFold2 model (Jumper et al., 2021) *predicts the folded representation, i.e., 3-D positions of all aminoacids*, for any protein sequence. The scale of this breakthrough became apparent in the biennial Critical Assessment of protein Structure Prediction (CASP) challenge.



$\Rightarrow$ much excitement and follow-up work (close to 30000 citations in the past 4 years) & 2024 Chemistry Nobel Prize.

Background: AlphaFold2

Deepmind's AlphaFold2 model (Jumper et al., 2021) *predicts the folded representation, i.e., 3-D positions of all aminoacids*, for any protein sequence. The scale of this breakthrough became apparent in the biennial Critical Assessment of protein Structure Prediction (CASP) challenge.



$\Rightarrow$ much excitement and follow-up work (close to 30000 citations in the past 4 years) & 2024 Chemistry Nobel Prize.

Fun Fact: "The key principle of the building block of the network —named Evoformer [...]—is to view the prediction of protein structures as a graph inference problem in 3D space in which the edges of the graph are defined by residues in proximity."

Background: ESM2

FAIR's ESM2 model (Lin et al., 2022) is a pre-trained large protein language model pre-trained on amino acid sequences.

Background: ESM2

FAIR's ESM2 model (Lin et al., 2022) is a pre-trained large protein language model pre-trained on amino acid sequences.

- there exists variants of this model ranging from 8M to 15B parameters.

Background: ESM2

FAIR's ESM2 model (Lin et al., 2022) is a pre-trained large protein language model pre-trained on amino acid sequences.

- there exists variants of this model ranging from 8M to 15B parameters.
- ESMFold is less accurate than Alphafold2.

Background: ESM2

FAIR's ESM2 model (Lin et al., 2022) is a pre-trained large protein language model pre-trained on amino acid sequences.

- there exists variants of this model ranging from 8M to 15B parameters.
- ESMFold is less accurate than Alphafold2.
- the learned ESM2 embeddings are of very high quality and can also be used function or protein property prediction. The learned embeddings are often used as node features in graph-based modelling approaches.

Protein-Ligand Docking

ML Task: Protein-Ligand Docking

The task of protein-ligand docking consists of predicting the likelihood of binding and binding pose of a relatively small molecule, called a ligand, binding on a relatively large protein structure.

Protein-Ligand Docking

ML Task: Protein-Ligand Docking

The task of protein-ligand docking consists of predicting the likelihood of binding and binding pose of a relatively small molecule, called a ligand, binding on a relatively large protein structure.

- Initially protein-ligand docking was posed as a **regression task**, in which docking poses are regressed (Stärk et al., 2022).

Protein-Ligand Docking

ML Task: Protein-Ligand Docking

The task of protein-ligand docking consists of predicting the likelihood of binding and binding pose of a relatively small molecule, called a ligand, binding on a relatively large protein structure.

- Initially protein-ligand docking was posed as a **regression task**, in which docking poses are regressed (Stärk et al., 2022).
- Posing protein-ligand docking as a **generative task**, in which docking poses are generated has shown great success (Corso et al., 2023).

Protein-Ligand Docking

ML Task: Protein-Ligand Docking

The task of protein-ligand docking consists of predicting the likelihood of binding and binding pose of a relatively small molecule, called a ligand, binding on a relatively large protein structure.

- Initially protein-ligand docking was posed as a **regression task**, in which docking poses are regressed (Stärk et al., 2022).
- Posing protein-ligand docking as a **generative task**, in which docking poses are generated has shown great success (Corso et al., 2023).

Important Distinction: Apo- and Holo-structure

When a protein docks to a small ligand, the protein three dimensional structure often changes as a result of the docking process.

Protein-Ligand Docking

ML Task: Protein-Ligand Docking

The task of protein-ligand docking consists of predicting the likelihood of binding and binding pose of a relatively small molecule, called a ligand, binding on a relatively large protein structure.

- Initially protein-ligand docking was posed as a **regression task**, in which docking poses are regressed (Stärk et al., 2022).
- Posing protein-ligand docking as a **generative task**, in which docking poses are generated has shown great success (Corso et al., 2023).

Important Distinction: Apo- and Holo-structure

When a protein docks to a small ligand, the protein three dimensional structure often changes as a result of the docking process. Consequently, when docking it is very important to distinguish the apo- and holo- structure of a protein.

- The holo-structure of a protein refers to the protein structure *after binding* to a ligand.

Protein-Ligand Docking

ML Task: Protein-Ligand Docking

The task of protein-ligand docking consists of predicting the likelihood of binding and binding pose of a relatively small molecule, called a ligand, binding on a relatively large protein structure.

- Initially protein-ligand docking was posed as a **regression task**, in which docking poses are regressed (Stärk et al., 2022).
- Posing protein-ligand docking as a **generative task**, in which docking poses are generated has shown great success (Corso et al., 2023).

Important Distinction: Apo- and Holo-structure

When a protein docks to a small ligand, the protein three dimensional structure often changes as a result of the docking process. Consequently, when docking it is very important to distinguish the apo- and holo- structure of a protein.

- The holo-structure of a protein refers to the protein structure *after binding* to a ligand.
- The apo-structure of a protein refers to the protein structure *in absence of* ligands.

Protein-Ligand Docking

ML Task: Protein-Ligand Docking

The task of protein-ligand docking consists of predicting the likelihood of binding and binding pose of a relatively small molecule, called a ligand, binding on a relatively large protein structure.

- Initially protein-ligand docking was posed as a **regression task**, in which docking poses are regressed (Stärk et al., 2022).
- Posing protein-ligand docking as a **generative task**, in which docking poses are generated has shown great success (Corso et al., 2023).

Important Distinction: Apo- and Holo-structure

When a protein docks to a small ligand, the protein three dimensional structure often changes as a result of the docking process. Consequently, when docking it is very important to distinguish the apo- and holo- structure of a protein.

- The holo-structure of a protein refers to the protein structure *after binding* to a ligand.
- The apo-structure of a protein refers to the protein structure *in absence of* ligands.

Many current state-of-the-art docking models are trained on the task of docking to the holo-structure (Corso et al., 2023).

DiffDock

DiffDock (Corso et al., 2023) is a recent deep learning based protein-ligand docking model. It has two components.

DiffDock

DiffDock (Corso et al., 2023) is a recent deep learning based protein-ligand docking model. It has two components.

- a score model, that generates docking poses of a ligand;

DiffDock

DiffDock (Corso et al., 2023) is a recent deep learning based protein-ligand docking model. It has two components.

- a score model, that generates docking poses of a ligand;
- a confidence model that ranks the generated docking poses from most to least likely to be correct.

DiffDock

DiffDock (Corso et al., 2023) is a recent deep learning based protein-ligand docking model. It has two components.

- a score model, that generates docking poses of a ligand;
- a confidence model that ranks the generated docking poses from most to least likely to be correct.

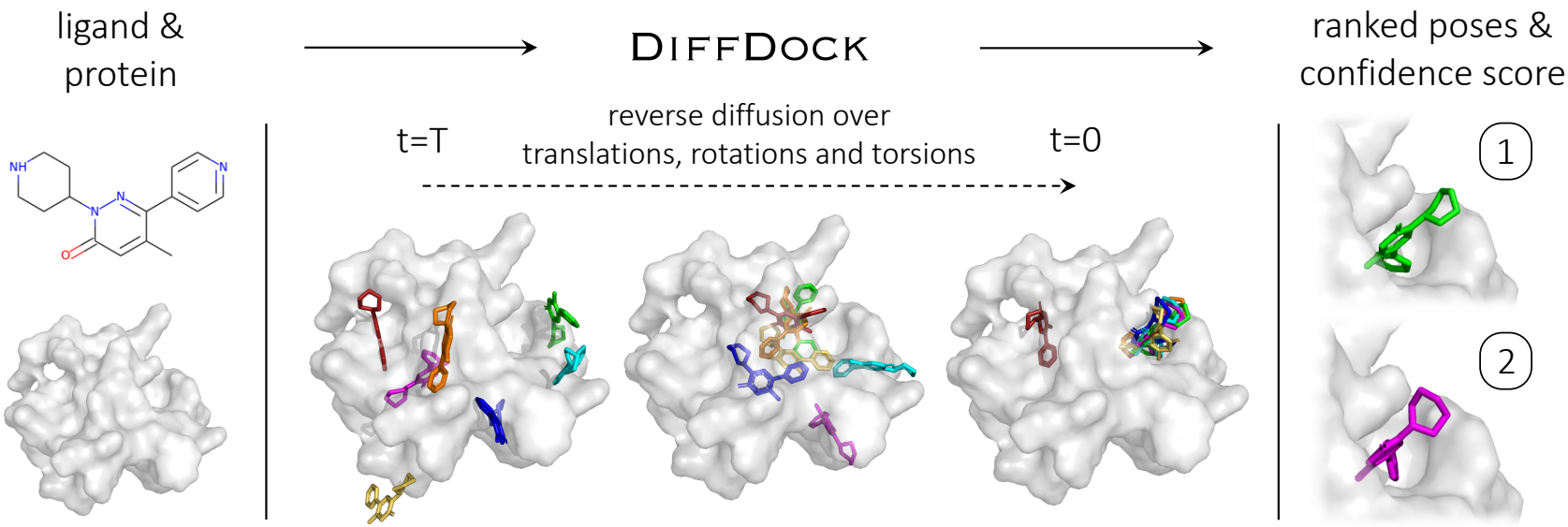
They do not diffuse over $\mathbb{R}^3$ instead they diffuse over the product space of ligand translations $\mathbb{T}(3)$, ligand rotations $SO(3)$ and changes to the torsion angles $SO(2)$.

DiffDock

DiffDock (Corso et al., 2023) is a recent deep learning based protein-ligand docking model. It has two components.

- a score model, that generates docking poses of a ligand;
- a confidence model that ranks the generated docking poses from most to least likely to be correct.

They do not diffuse over $\mathbb{R}^3$ instead they diffuse over the product space of ligand translations $\mathbb{T}(3)$, ligand rotations $SO(3)$ and changes to the torsion angles $SO(2)$.



Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model).

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.
- protein atoms-protein atoms: distance threshold of 5Å and maximal number of neighbours of 8.

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.
- protein atoms-protein atoms: distance threshold of 5Å and maximal number of neighbours of 8.
- amino acids-amino acids: distance threshold of 15 Å and maximal number of neighbours of 24.

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.
- protein atoms-protein atoms: distance threshold of 5Å and maximal number of neighbours of 8.
- amino acids-amino acids: distance threshold of 15 Å and maximal number of neighbours of 24.
- ligand atoms-protein atoms: distance threshold of 5Å.

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.
- protein atoms-protein atoms: distance threshold of 5Å and maximal number of neighbours of 8.
- amino acids-amino acids: distance threshold of 15 Å and maximal number of neighbours of 24.
- ligand atoms-protein atoms: distance threshold of 5Å.
- ligand atoms-amino acids: distance threshold of $20 + 3\sigma_{tr}$Å, where σ_{tr} denotes the current standard deviation of the diffusion translational noise.

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.
- protein atoms-protein atoms: distance threshold of 5Å and maximal number of neighbours of 8.
- amino acids-amino acids: distance threshold of 15 Å and maximal number of neighbours of 24.
- ligand atoms-protein atoms: distance threshold of 5Å.
- ligand atoms-amino acids: distance threshold of $20 + 3\sigma_{tr}$Å, where σ_{tr} denotes the current standard deviation of the diffusion translational noise.
- amino acids-protein atoms edges are drawn between all protein atoms that form a given amino acid.

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.
- protein atoms-protein atoms: distance threshold of 5Å and maximal number of neighbours of 8.
- amino acids-amino acids: distance threshold of 15 Å and maximal number of neighbours of 24.
- ligand atoms-protein atoms: distance threshold of 5Å.
- ligand atoms-amino acids: distance threshold of $20 + 3\sigma_{tr}$Å, where σ_{tr} denotes the current standard deviation of the diffusion translational noise.
- amino acids-protein atoms edges are drawn between all protein atoms that form a given amino acid.

Node and edge features are initialised as follows and then processed by two-layer MLPs.

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.
- protein atoms-protein atoms: distance threshold of 5Å and maximal number of neighbours of 8.
- amino acids-amino acids: distance threshold of 15 Å and maximal number of neighbours of 24.
- ligand atoms-protein atoms: distance threshold of 5Å.
- ligand atoms-amino acids: distance threshold of $20 + 3\sigma_{tr}$Å, where σ_{tr} denotes the current standard deviation of the diffusion translational noise.
- amino acids-protein atoms edges are drawn between all protein atoms that form a given amino acid.

Node and edge features are initialised as follows and then processed by two-layer MLPs.

- amio acid nodes: the residue type and ESM2 embeddings.

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.
- protein atoms-protein atoms: distance threshold of 5Å and maximal number of neighbours of 8.
- amino acids-amino acids: distance threshold of 15 Å and maximal number of neighbours of 24.
- ligand atoms-protein atoms: distance threshold of 5Å.
- ligand atoms-amino acids: distance threshold of $20 + 3\sigma_{tr}$Å, where σ_{tr} denotes the current standard deviation of the diffusion translational noise.
- amino acids-protein atoms edges are drawn between all protein atoms that form a given amino acid.

Node and edge features are initialised as follows and then processed by two-layer MLPs.

- amio acid nodes: the residue type and ESM2 embeddings.
- Ligand atom nodes: several chemical properties of the atom.

Detailed Functioning of DiffDock: Embedding Layer

Construct a *geometric heterogeneous graph* containing nodes representing ligand atoms, amino acids, and protein atoms (only used in the confidence model). Then, edges are drawn based as follows.

- Ligand atoms-ligand atoms: distance threshold of 5Å. In addition, covalent bonds in the ligand give rise to edges.
- protein atoms-protein atoms: distance threshold of 5Å and maximal number of neighbours of 8.
- amino acids-amino acids: distance threshold of 15 Å and maximal number of neighbours of 24.
- ligand atoms-protein atoms: distance threshold of 5Å.
- ligand atoms-amino acids: distance threshold of $20 + 3\sigma_{tr}$Å, where σ_{tr} denotes the current standard deviation of the diffusion translational noise.
- amino acids-protein atoms edges are drawn between all protein atoms that form a given amino acid.

Node and edge features are initialised as follows and then processed by two-layer MLPs.

- amio acid nodes: the residue type and ESM2 embeddings.
- Ligand atom nodes: several chemical properties of the atom.
- edges: radial basis embeddings of edge length (Schütt et al., 2018) and bond type for covalent bonds.

Detailed Functioning of DiffDock: Interaction Layer

1) The interaction layer transforms each hidden state using a Tensor Field Network Layer (TFN) (Thomas et al., 2018). TFNs generalise CNNs to learn rotation-equivariant functions on spheres. We learn separate set of weights for node and edge types.

Detailed Functioning of DiffDock: Interaction Layer

1) The interaction layer transforms each hidden state using a Tensor Field Network Layer (TFN) (Thomas et al., 2018). TFNs generalise CNNs to learn rotation-equivariant functions on spheres. We learn separate set of weights for node and edge types.
2) We then average the updated representations of all hidden states of a given node type in the neighbourhood of our given node.

Detailed Functioning of DiffDock: Interaction Layer

1) The interaction layer transforms each hidden state using a Tensor Field Network Layer (TFN) (Thomas et al., 2018). TFNs generalise CNNs to learn rotation-equivariant functions on spheres. We learn separate set of weights for node and edge types.
2) We then average the updated representations of all hidden states of a given node type in the neighbourhood of our given node.
3) We apply Batch Norm to the averaged representations.

Detailed Functioning of DiffDock: Interaction Layer

1) The interaction layer transforms each hidden state using a Tensor Field Network Layer (TFN) (Thomas et al., 2018). TFNs generalise CNNs to learn rotation-equivariant functions on spheres. We learn separate set of weights for node and edge types.
2) We then average the updated representations of all hidden states of a given node type in the neighbourhood of our given node.
3) We apply Batch Norm to the averaged representations.
4) Finally, we sum over node types.

Detailed Functioning of DiffDock: Output Layer

The output layer is where the main difference arises in the design of the score and confidence model.

Detailed Functioning of DiffDock: Output Layer

The output layer is where the main difference arises in the design of the score and confidence model.

- The score model's output is in the tangent space $T_{\mathbf{r}}\mathbb{T}_3 \oplus T_R SO(3) \oplus T_{\boldsymbol{\theta}} SO(2)^m$. This corresponds to having two $SE(3)$-equivariant output vectors representing the translational and rotational score predictions and m $SE(3)$-invariant output scalars representing the torsional score. For each of these, the DiffDock authors designed final tensor-product convolutions.

Detailed Functioning of DiffDock: Output Layer

The output layer is where the main difference arises in the design of the score and confidence model.

- The score model's output is in the tangent space $T_{\mathbf{r}}\mathbb{T}_3 \oplus T_R SO(3) \oplus T_{\boldsymbol{\theta}} SO(2)^m$. This corresponds to having two $SE(3)$-equivariant output vectors representing the translational and rotational score predictions and m $SE(3)$-invariant output scalars representing the torsional score. For each of these, the DiffDock authors designed final tensor-product convolutions.
- The confidence model outputs a single $SE(3)$-invariant scalar representing the confidence score. This is done by averaging ligand atom representations and applying a 3-layer MLP.

DiffDock Results

	Holo crystal proteins				Apo ESMFold proteins				
	Top-1 RMSD		Top-5 RMSD		Top-1 RMSD		Top-5 RMSD		Average
Method	%<2	Med.	%<2	Med.	%<2	Med.	%<2	Med.	Runtime (s)
GNINA	22.9	7.7	32.9	4.5	2.0	22.3	4.0	14.22	127
SMINA	18.7	7.1	29.3	4.6	3.4	15.4	6.9	10.0	126*
GLIDE	21.8	9.3							1405*
EQUIBIND	5.5	6.2	-	-	1.7	7.1	-	-	0.04
TANKBIND	20.4	4.0	24.5	3.4	10.4	5.4	14.7	4.3	0.7/2.5
P2RANK+SMINA	20.4	6.9	33.2	4.4	4.6	10.0	10.3	7.0	126*
P2RANK+GNINA	28.8	5.5	38.3	3.4	8.6	11.2	12.8	7.2	127
EQUIBIND+SMINA	23.2	6.5	38.6	3.4	4.3	8.3	11.7	5.8	126*
EQUIBIND+GNINA	28.8	4.9	39.1	3.1	10.2	8.8	18.6	5.6	127
DiffDock (10)	35.0	3.6	40.7	2.65	**21.7**	**5.0**	**31.9**	**3.3**	10
DiffDock (40)	**38.2**	**3.3**	**44.7**	**2.40**	20.3	5.1	31.3	**3.3**	40

$\Rightarrow$ DiffDock convincingly outperforms a variety of baselines in the blind docking task.

Protein-Docking: Recent Updates I

At ICLR 2024, DiffDock-L (Corso et al., 2024) was published

Protein-Docking: Recent Updates I

At ICLR 2024, DiffDock-L (Corso et al., 2024) was published

- the training data and model size (parameters) were scaled up;

Protein-Docking: Recent Updates I

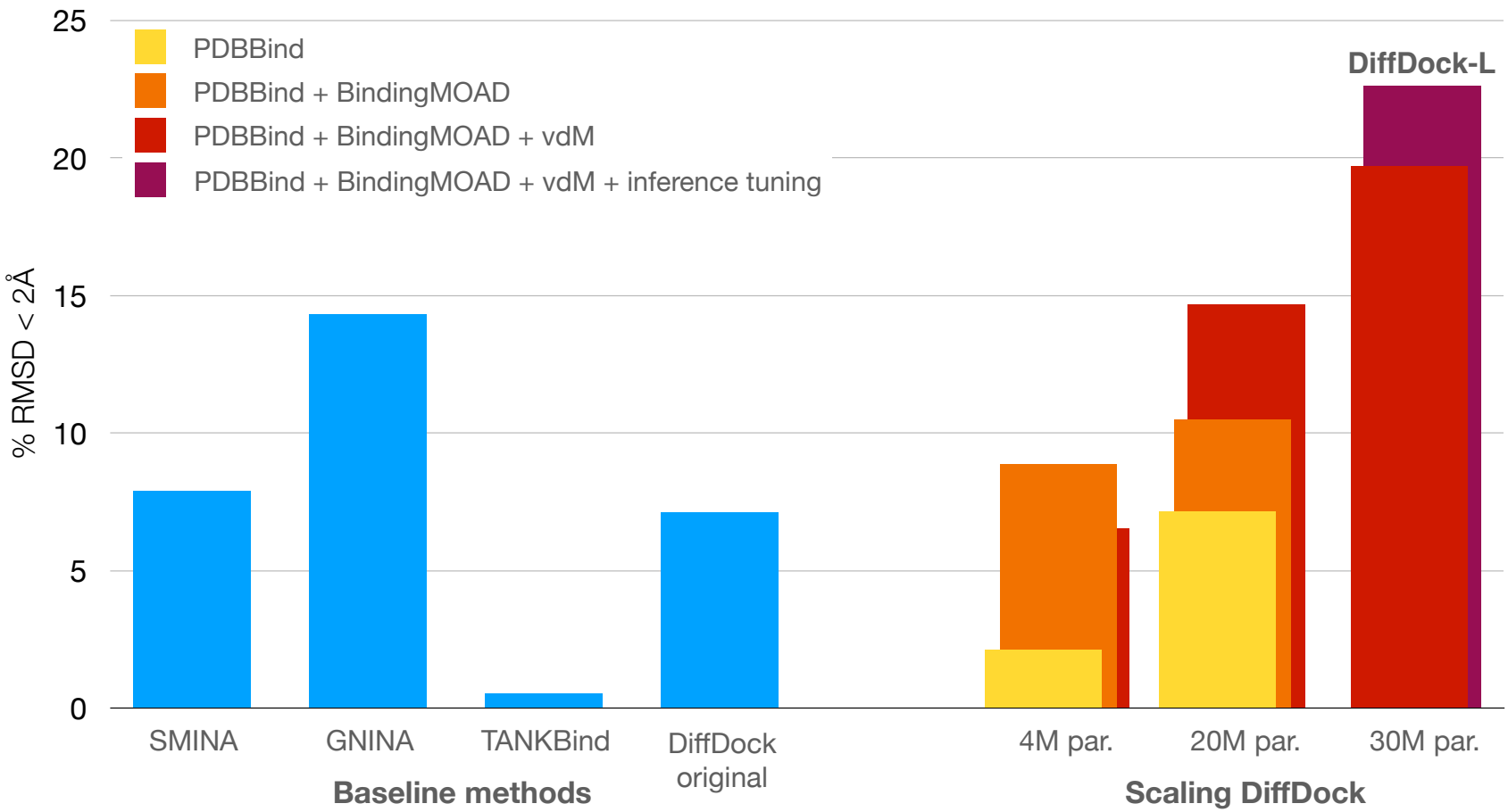
At ICLR 2024, DiffDock-L (Corso et al., 2024) was published

- the training data and model size (parameters) were scaled up;
- a new RL-like self-training method using feedback from the confidence model was incorporated.

Protein-Docking: Recent Updates I

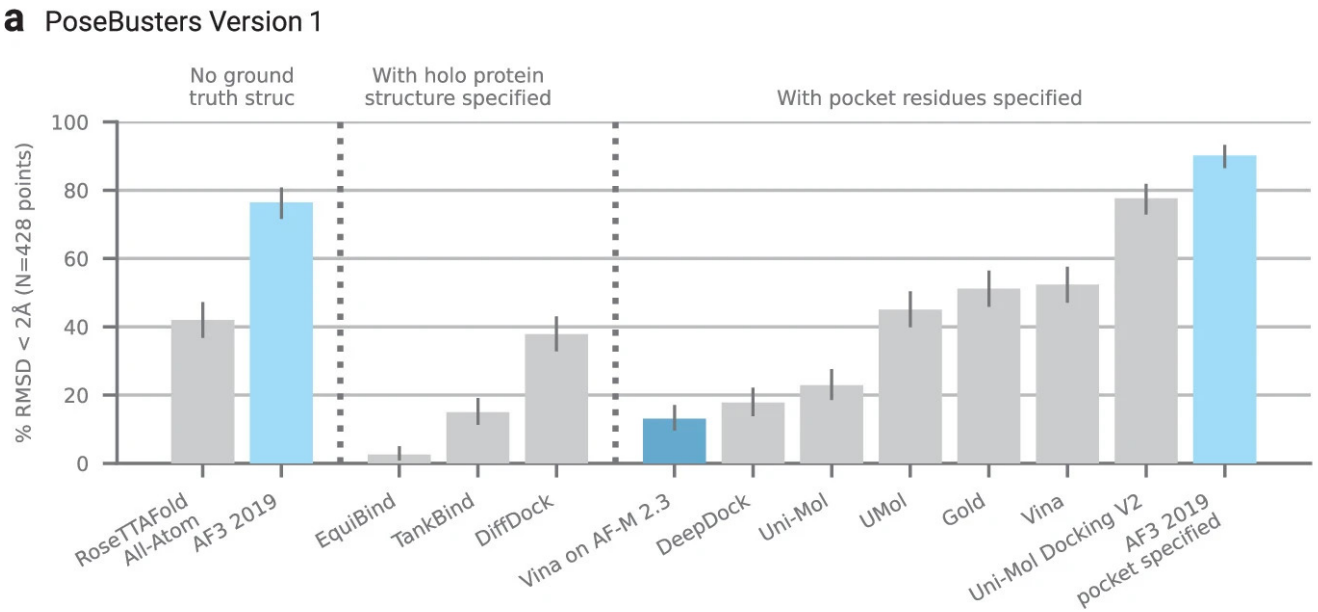
At ICLR 2024, DiffDock-L (Corso et al., 2024) was published

- the training data and model size (parameters) were scaled up;
- a new RL-like self-training method using feedback from the confidence model was incorporated.



Protein-Docking: Recent Updates II

Alphafold 3 has been published by DeepMind a few months ago (Abramson et al., 2024) and they claim state-of-the-art protein-ligand docking performance on apo-strcutures.



Antibiotic Resistance Classification (Qabel et al., 2022): Problem Set-Up

Problem: **Antimicrobial resistance** describes the property of germs, e.g., bacteria and fungi, to develop the ability to defeat drugs designed to kill them.

Antibiotic Resistance Classification (Qabel et al., 2022): Problem Set-Up

Problem: **Antimicrobial resistance** describes the property of germs, e.g., bacteria and fungi, to develop the ability to defeat drugs designed to kill them.

Relevant Data: **Amioacid Sequences**, i.e., proteins, together with their resistance classes in the COALA dataset (Hamid & Friedberg, 2020) (germ DNA $\rightarrow$ RNA $\rightarrow$ protein).

Antibiotic Resistance Classification (Qabel et al., 2022): Problem Set-Up

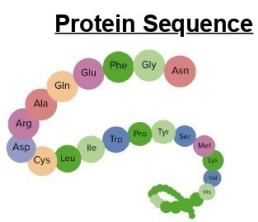
Problem: **Antimicrobial resistance** describes the property of germs, e.g., bacteria and fungi, to develop the ability to defeat drugs designed to kill them.

Relevant Data: **Amioacid Sequences**, i.e., proteins, together with their resistance classes in the COALA dataset (Hamid & Friedberg, 2020) (germ DNA $\rightarrow$ RNA $\rightarrow$ protein).

Machine Learning Task: **Antibiotic Resistance Classification**.

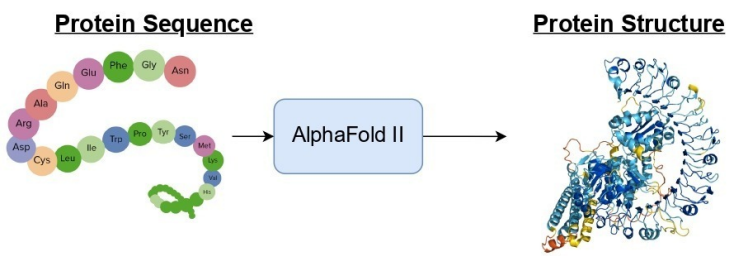
Methodology

1) Given a protein, i.e., a sequence of aminoacids we



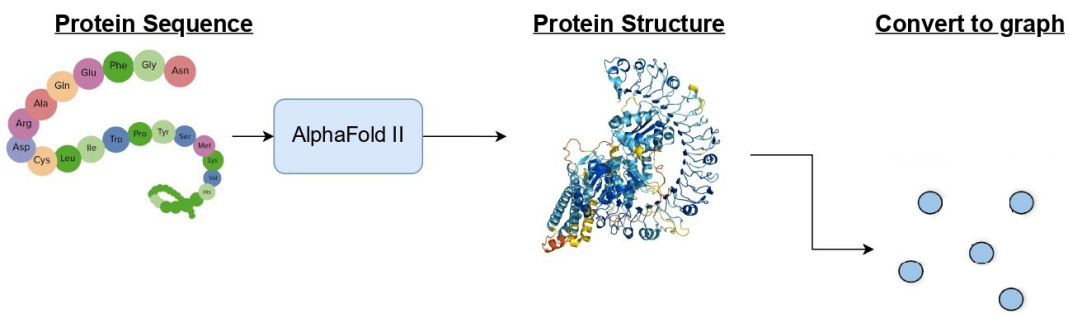
Methodology

1) Given a protein, i.e., a sequence of aminoacids we
   1) use the AlphaFoldII model (Jumper et al., 2021) to infer its 3D structure.



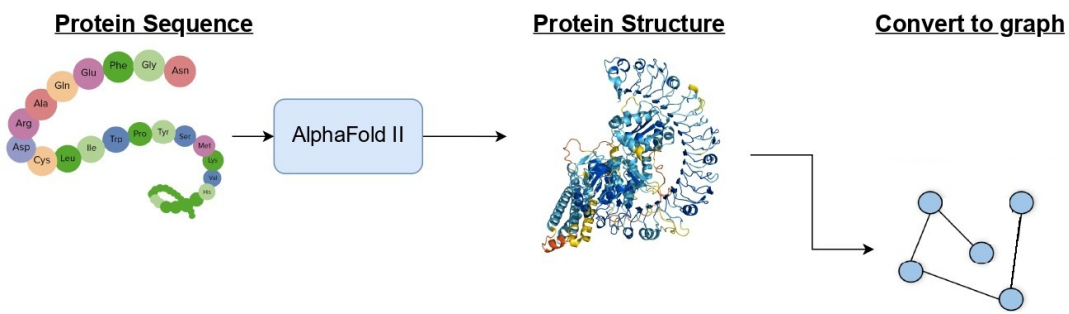
Methodology

1) Given a protein, i.e., a sequence of aminoacids we
   1) use the AlphaFoldII model (Jumper et al., 2021) to infer its 3D structure. We then convert this 3D structure into a **graph**, in which every aminoacid corresponds to a node and edges are drawn if



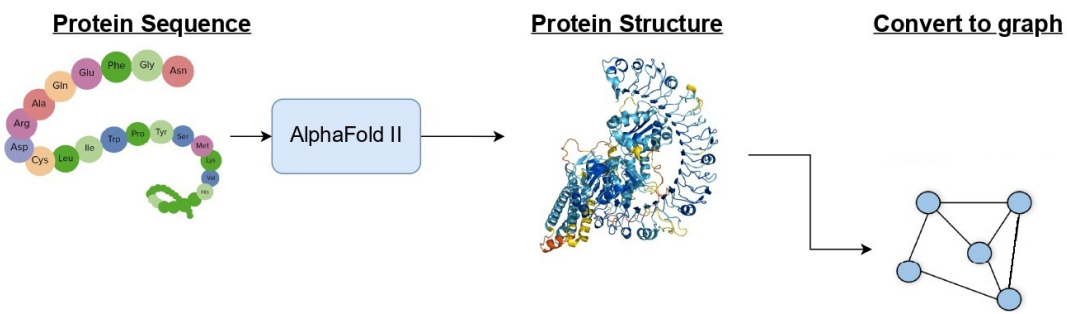
Methodology

1) Given a protein, i.e., a sequence of aminoacids we
   1) use the AlphaFoldII model (Jumper et al., 2021) to infer its 3D structure. We then convert this 3D structure into a **graph**, in which every aminoacid corresponds to a node and edges are drawn if
      - aminoacids are adjacent in the sequence,



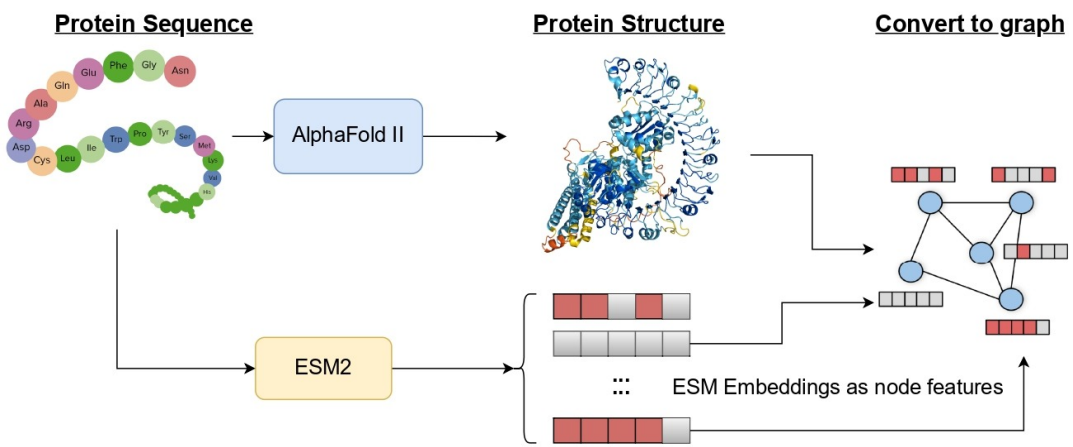
Methodology

1) Given a protein, i.e., a sequence of aminoacids we
   1) use the AlphaFoldII model (Jumper et al., 2021) to infer its 3D structure. We then convert this 3D structure into a **graph**, in which every aminoacid corresponds to a node and edges are drawn if
      - aminoacids are adjacent in the sequence,
      - aminoacids have a distance less than 8 Å in the 3D structure.



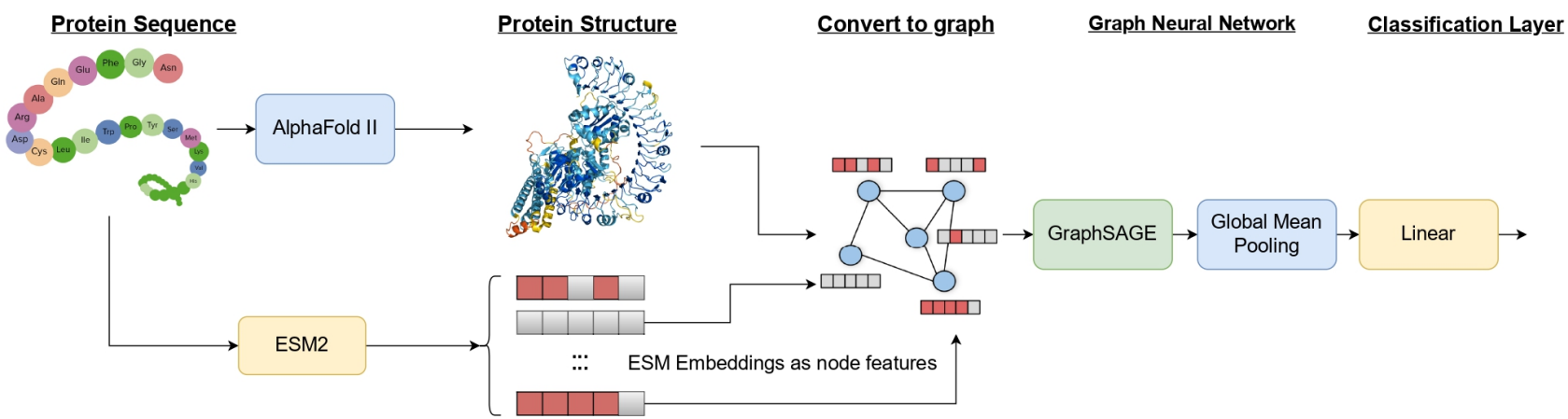
Methodology

1) Given a protein, i.e., a sequence of aminoacids we
   1) use the AlphaFoldII model (Jumper et al., 2021) to infer its 3D structure. We then convert this 3D structure into a **graph**, in which every aminoacid corresponds to a node and edges are drawn if
      - aminoacids are adjacent in the sequence,
      - aminoacids have a distance less than 8 Å in the 3D structure.
   2) use the ESM2 model (Lin et al., 2022) to obtain vectorial aminoacid representations to be used as **node features** in our graph.



Methodology

1) Given a protein, i.e., a sequence of aminoacids we
   1) use the AlphaFoldII model (Jumper et al., 2021) to infer its 3D structure. We then convert this 3D structure into a **graph**, in which every aminoacid corresponds to a node and edges are drawn if
      - aminoacids are adjacent in the sequence,
      - aminoacids have a distance less than 8 Å in the 3D structure.
   2) use the ESM2 model (Lin et al., 2022) to obtain vectorial aminoacid representations to be used as **node features** in our graph.
2) We apply a GNN to this graph structured data to perform the task of Antibiotic Resistance Classification.



Results

Table: Classification results (± standard deviation) of the different approaches on the COALA dataset. The best performance in the different groups is typeset in **bold**. Note that N/A refers to "not applicable".

Method	Accuracy	Accuracy (<50% id)	Accuracy (>50% id)	Accuracy (homolog not found)
BLAST	65.42% (± 0.57%)	75.11% (± 1.78%)	**95.29% (± 1.18%)**	N/A
Diamond	58.86% (± 0.62%)	64.69% (± 1.72%)	95.11% (± 1.53%)	N/A
TF-IDF	54.19% (± 1.62%)	55.18% (± 1.72%)	84.05% (± 2.07%)	35.43% (± 1.25%)
TRAC	69.80% (± 0.66%)	72.78% (± 1.77%)	90.92% (± 2.16%)	36.83% (± 4.36%)
ARG-SHINE	68.34% (± 1.27%)	69.60% (± 1.78%)	92.88% (± 0.70%)	38.00% (± 3.88%)
ARGGNN	**72.90% (± 0.65%)**	**76.49% (± 1.30%)**	93.00% (± 1.10%)	**38.90% (± 3.98%)**

- Our ARGGNN outperforms the state-of-the-art baseline models in the task of Antibiotic Resistance Classification.

Results

Table: Classification results (± standard deviation) of the different approaches on the COALA dataset. The best performance in the different groups is typeset in **bold**. Note that N/A refers to "not applicable".

Method	Accuracy	Accuracy (<50% id)	Accuracy (>50% id)	Accuracy (homolog not found)
BLAST	65.42% (± 0.57%)	75.11% (± 1.78%)	**95.29% (± 1.18%)**	N/A
Diamond	58.86% (± 0.62%)	64.69% (± 1.72%)	95.11% (± 1.53%)	N/A
TF-IDF	54.19% (± 1.62%)	55.18% (± 1.72%)	84.05% (± 2.07%)	35.43% (± 1.25%)
TRAC	69.80% (± 0.66%)	72.78% (± 1.77%)	90.92% (± 2.16%)	36.83% (± 4.36%)
ARG-SHINE	68.34% (± 1.27%)	69.60% (± 1.78%)	92.88% (± 0.70%)	38.00% (± 3.88%)
ARGGNN	**72.90% (± 0.65%)**	**76.49% (± 1.30%)**	93.00% (± 1.10%)	**38.90% (± 3.98%)**

- Our ARGGNN outperforms the state-of-the-art baseline models in the task of Antibiotic Resistance Classification.
- It is unclear how these results would look if the 15 billion parameter ESM2 model was used.

What's Next For Us?

- Deep Learning Models for Molecular Dynamic simulations
  Dr. Masoud Ramuz and Siyun Wang collaborating with LOB at IPP

What's Next For Us?

- Deep Learning Models for Molecular Dynamic simulations
  Dr. Masoud Ramuz and Siyun Wang collaborating with LOB at IPP
- GNNs in Histopathology
  Niklas Kormann quantifying kidney health

Conclusions

- There are many applications of Deep Learning in Biomedicine. Currently this field is growing at an immense speed.

Conclusions

- There are many applications of Deep Learning in Biomedicine. Currently this field is growing at an immense speed.
- Geometric GNNs gracefully handle graphs in which nodes are located in some Euclidean space.

Conclusions

- There are many applications of Deep Learning in Biomedicine. Currently this field is growing at an immense speed.
- Geometric GNNs gracefully handle graphs in which nodes are located in some Euclidean space.
- Protein-Ligand docking is an important learning task and there is much recent progress in the development of deep learning models for this task.

Conclusions

- There are many applications of Deep Learning in Biomedicine. Currently this field is growing at an immense speed.
- Geometric GNNs gracefully handle graphs in which nodes are located in some Euclidean space.
- Protein-Ligand docking is an important learning task and there is much recent progress in the development of deep learning models for this task.
- One can predict many properties of interest from protein data, one such property is antibiotic resistance.

Thank you for your attention!

johanneslutzeyer.com

References

J. Abramson, et al., "Accurate structure prediction of biomolecular interactions with AlphaFold 3," *Nature*, pp. 493-500, 2024.

I. Batatia, D. P. Kovacs, G. Simm, C. Ortner & G. Csányi, "MACE: Higher order equivariant message passing neural networks for fast and accurate force fields," *Advances in Neural Information Processing Systems*, pp. 11423-11436, 2022.

G. Corso, H. Stärk, B. Jing, R. Barzilay & T. Jaakkola, "DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking," *International Conference on Learning Representations (ICLR)*, 2023.

G. Corso, A. Deng, B. Fry, N. Polizzi, R. Barzilay & T. Jaakkola, "Deep Confident Steps To New Pockets: Strategies for Docking Generalization," *International Conference on Learning Representations (ICLR)*, 2024.

A. Duval, S. V. Mathis, C. K. Joshi, V. Schmidt, S. Miret, F. D. Malliaros, T. Cohen, P. Liò, Y. Bengio & M. Bronstein, "A Hitchhiker's Guide to Geometric GNNs for 3D Atomic Systems," *arXiv:2312.07511*, 2023.

J. Gasteiger, J. Groß & S. Günnemann, "Directional message passing for molecular graphs," *International Conference on Learning Representations (ICLR)*, 2020.

J. Gasteiger, F. Becker & S. Günnemann, "Gemnet: Universal directional graph neural networks for molecules," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

M. N. Hamid & I. Friedberg, "Transfer Learning Improves Antibiotic Resistance Class Prediction," *biorxiv:10.1101/2020.04.17.047316*, 2020.

C. K. Joshi, C. Bodnar, S. V. Mathis, T. Cohen & P. Lio, "On the expressive power of geometric graph neural networks," *ICML*, 2023.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli & D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, pp. 583–589, 2021.

Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, A. dos Santos, Costa, M. Fazel-Zarandi, R. Sercu, S. Candido & A. Rives, "Language Models of Protein Sequences at the Scale of Evolution Enable Accurate Structure Prediction," *biorxiv:10.1101/10.1101/2022.07.20.500902v1*, 2022.

A. Qabel, S. Ennadir, G. Nikolentzos, J. F. Lutzeyer, M. Chatzianastasis, H. Bostrom & M. Vazirgiannis, "Structure-Aware Antibiotic Resistance Classification Using Graph Neural Networks," *NeurIPS AI for Science Workshop*, 2022.

V. G. Satorras, E. Hoogeboom & M. Welling, " E (n) equivariant graph neural networks," *ICML*, 2021.

K. T. Schütt, H. E. Sauceda, P. J. Kindermans, A. Tkatchenko & K.R. Müller, "Schnet–a deep learning architecture for molecules and materials," *The Journal of Chemical Physics*, 2018.

H. Stärk, O. Ganea, L. Pattanaik, R. Barzilay & T. Jaakkola, "Equibind: Geometric deep learning for drug binding structure prediction," In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 20503-20521, 2022.

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