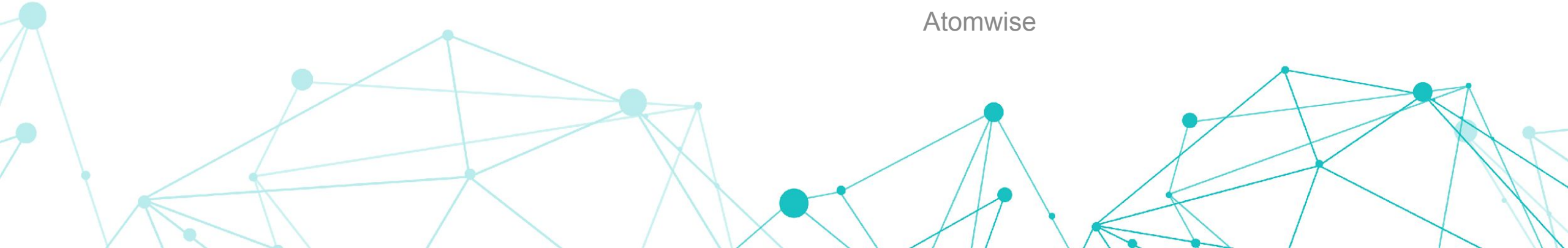
Atomwise

Protein-ligand ~~binding prediction~~ pose classification with SO(3)-equivariant neural networks

Brandon M. Anderson
Interim head of ml.research
Atomwise

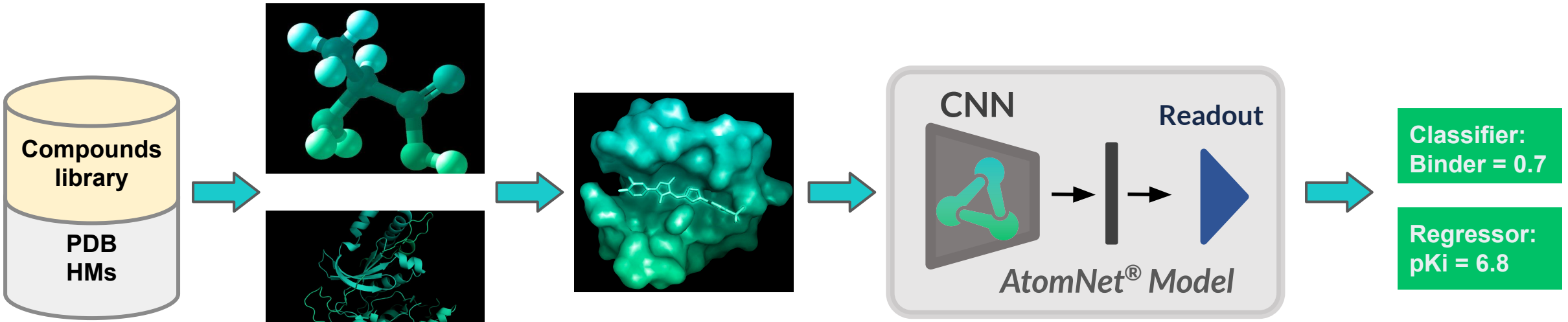


Agenda

- Structure-based neural networks
- Equivariant networks
- Pose classification task
- Results

Structure-based drug design

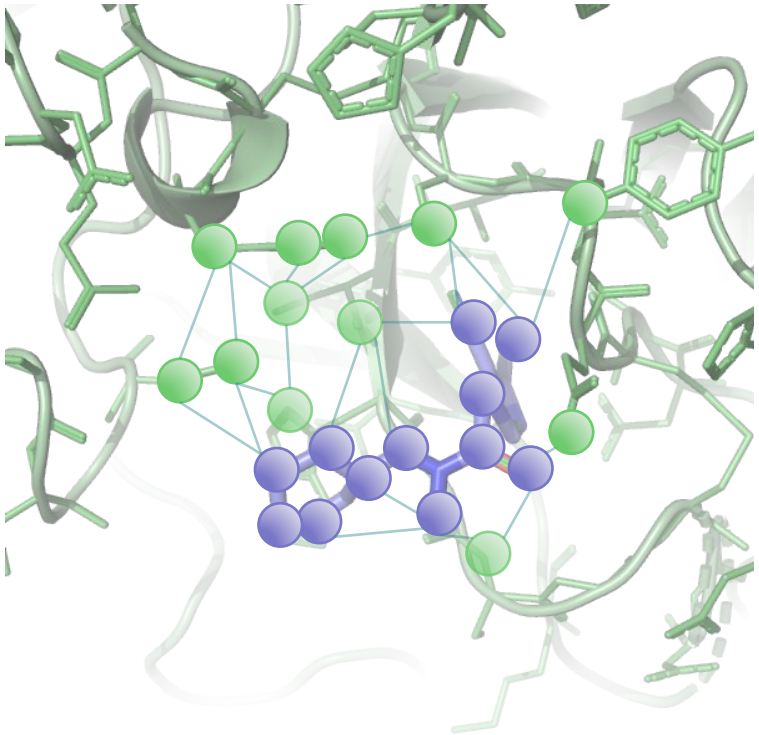
The Atomwise story

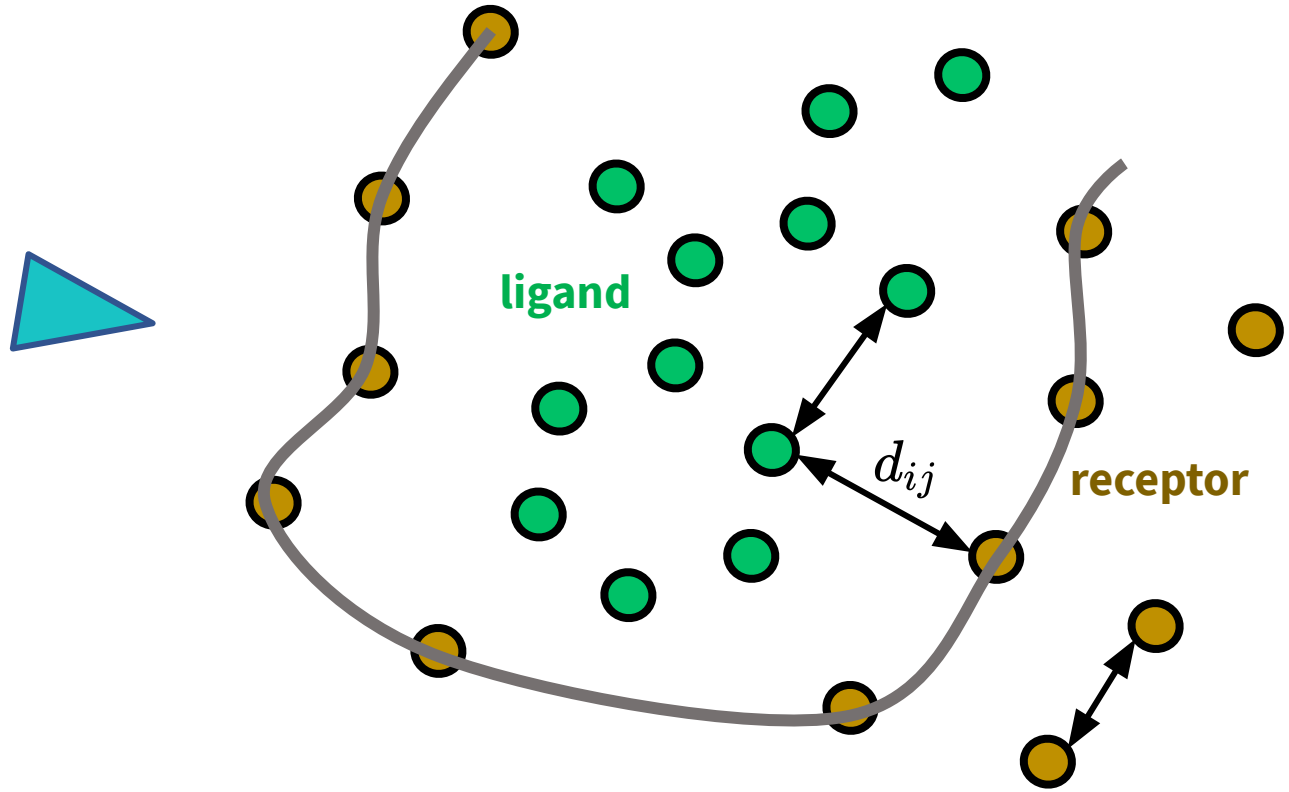


Data | Ligand Receptor | Docked Complex | ML | Prediction

Point-based networks: Atoms and Distances

Message passing neural networks (MPNN) on atomic graphs





Point-based Message Passing

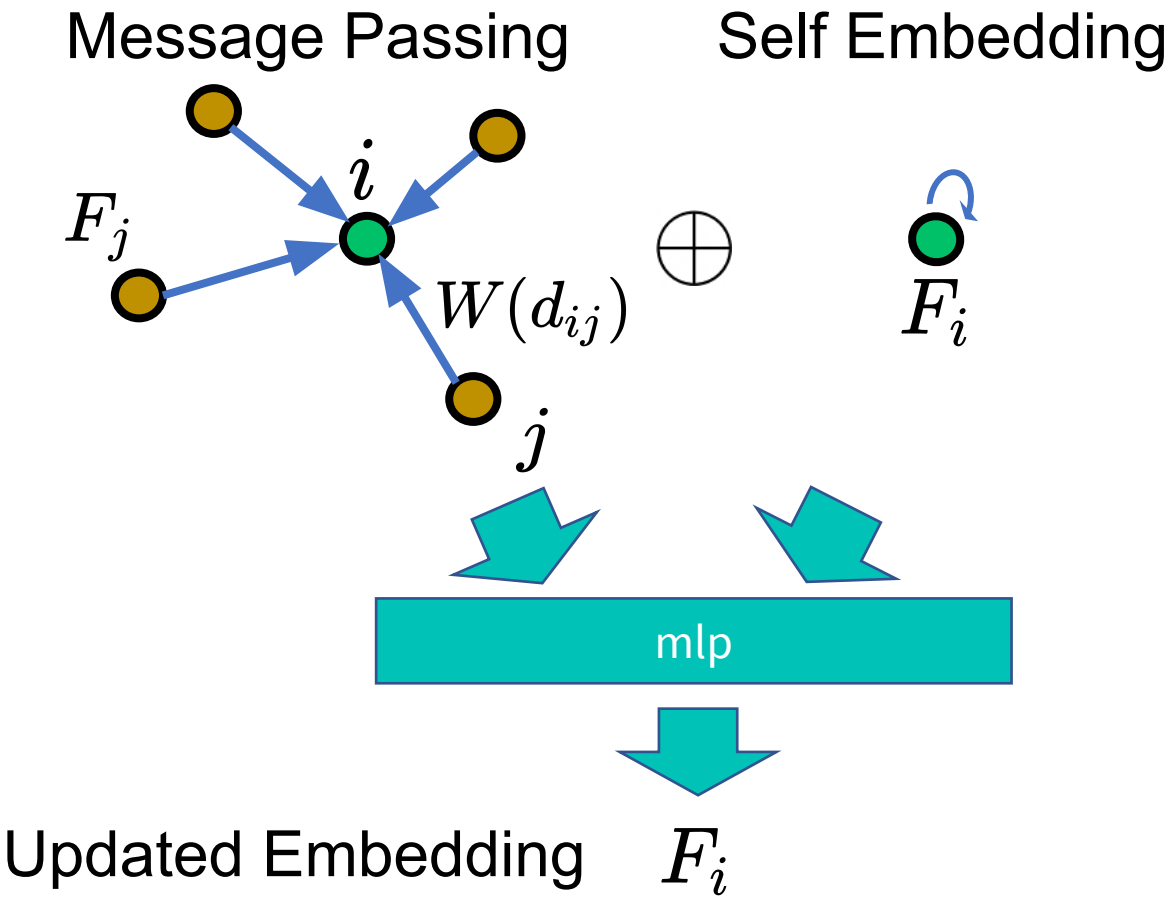
Aka a pairwise convolution

- Update embeddings depend on
  - Source atom embedding
  - Distance between atoms
- Message passing:

$$F_i \sim \sum_j W(d_{ij}) F_j$$

- Edges are functions of position with spherical bessel functions basis:

$$W(r) = \sum_n c_n j_0 \left(\frac{z_{0n} r}{R}\right)$$



Pairwise convolution introduced in: Schutt, et. al., NeurIPS 2017
Message passing neural networks: Gilmer, et. al., ICML 2017

SO(3)-Equivariant networks

Limitation: only pass pairwise distances
    How to capture local geometric information?

Promote scalar hidden states to "geometric" internal state (scalar, vector, tensor, ...)

How to do this and maintain symmetry of the problem?

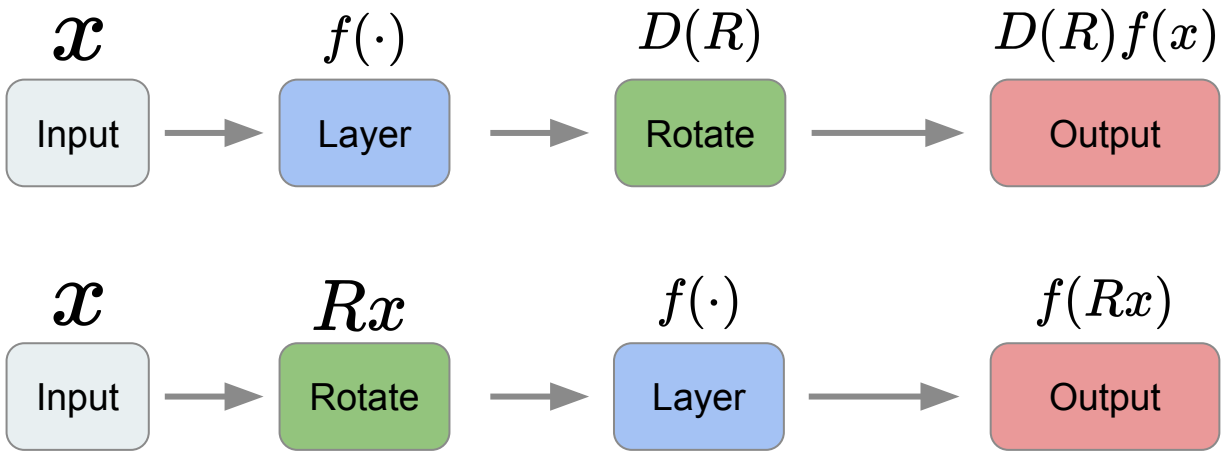
Thomas, et. al., https://arxiv.org/abs/1802.08219
BMA, Hy, Kondor, NeurIPS 2019

Equivariant layers

A rotation R of the input gives a fixed and predictable transformation of output

- Transformation is multiplication by a matrix $D(R)$

$$f(Rx) = D(R)f(x)$$



Group theory

A general framework for equivariant networks

Geometric tensors can be decomposed into "irreducible representations"

- labeled by integer *l*=0, 1, 2, ...
- *l*=0: scalar (monopole)
- *l*=1: vector (dipole)
- *l*=2: tensor (quadrupole)

Spherical tensors have size (2**l*+1)

- Standard basis: Spherical harmonics

Transformation properties by a rotation *R* pre-determined by "Wigner-D matrices" *D(R)* (2**l*+1, 2**l*+1)

Tensor products reduced using "Clebsch-Gordan decomposition" (linear transformation)

Spherical Harmonics $Y_m^\ell(\hat{\mathbf{r}})$

Basis functions for data on a sphere

Angular components of hydrogen wavefunctions

(2 * *l* + 1) components

Real spherical harmonics ("tesseral harmonics")

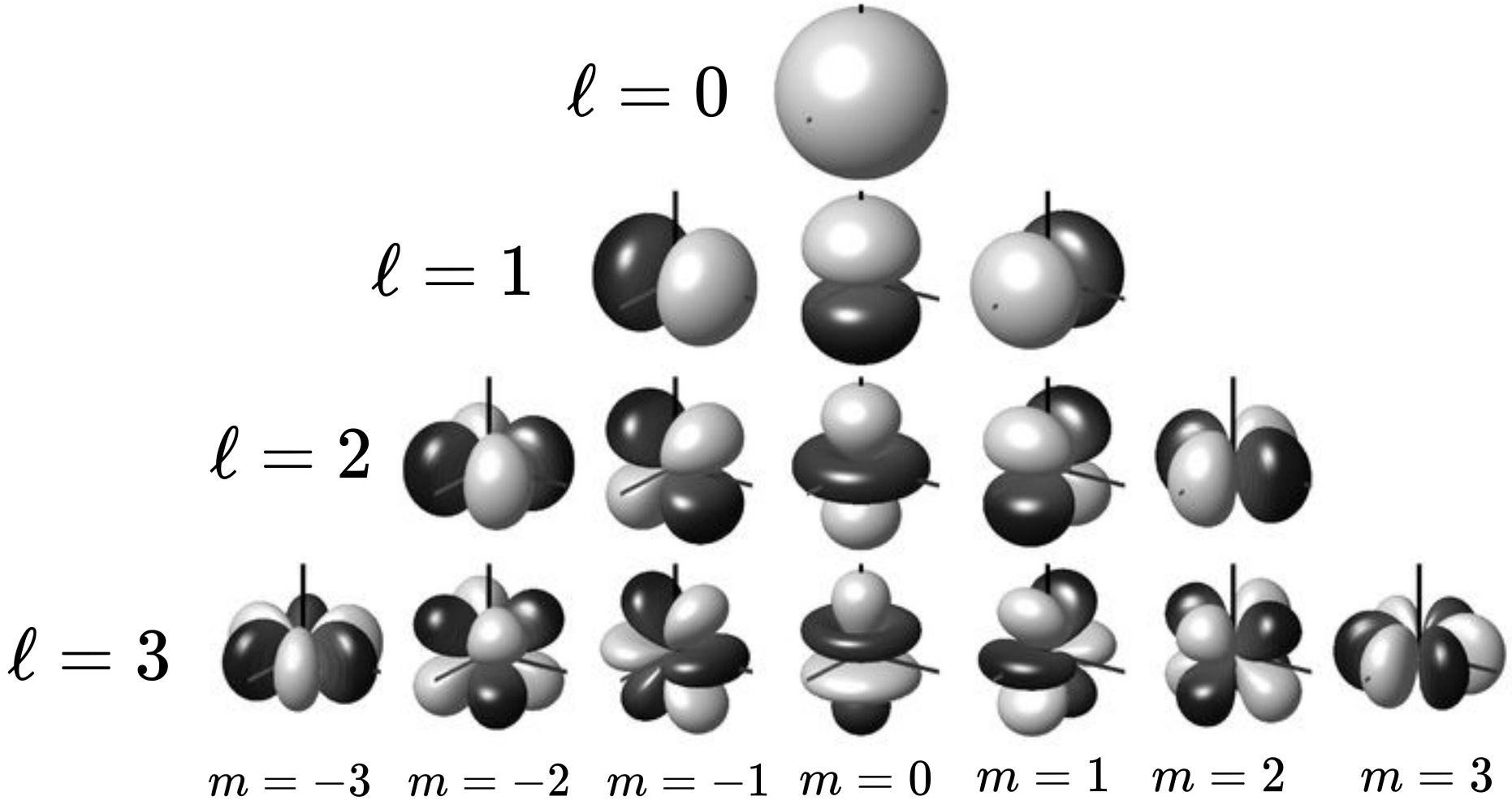


Image credit: https://en.wikipedia.org/wiki/File:Spherical_Harmonics_deg3.png

Clebsch-Gordan transformation

$$Y_{\ell_1}(x) \otimes Y_{\ell_2}(x) = C_{\ell_1 \ell_2} \left[\bigoplus_{\ell=|\ell_1-\ell_2|}^{\ell_1+\ell_2} Y_\ell(x) \right]$$

Example: l=1 with l=1: $3 \otimes 3 \to 1 \oplus 3 \oplus 5$

- Scalar product (l=0) $v_1 \cdot v_2$
- Vector product (l=1) $v_1 \times v_2$
- Anti-symmetric traceless product (l=2) $(v_1)_i (v_2)_j + (v_1)_j (v_2)_i - \frac{\delta_{ij}}{3}(v_1 \cdot v_2)$

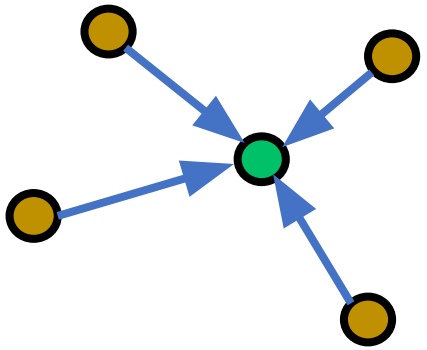
Tensor Field Networks (TFNs)

Key ingredients: Geometric tensors

Hidden states: Geometric tensors

$$F_i = \left(F_i^{(0)}, F_i^{(1)}, \ldots, F_i^{(\ell)}\right)$$

$$F_j = \left(F_j^{(0)}, F_j^{(1)}, \ldots, F_j^{(\ell)}\right)$$



Thomas, et. al., https://arxiv.org/abs/1802.08219

Tensor Field Networks (TFNs)

Key ingredients: Message passing (Pairwise convolution in the paper)

Message passing:

- Edge function is now a spherical tensor

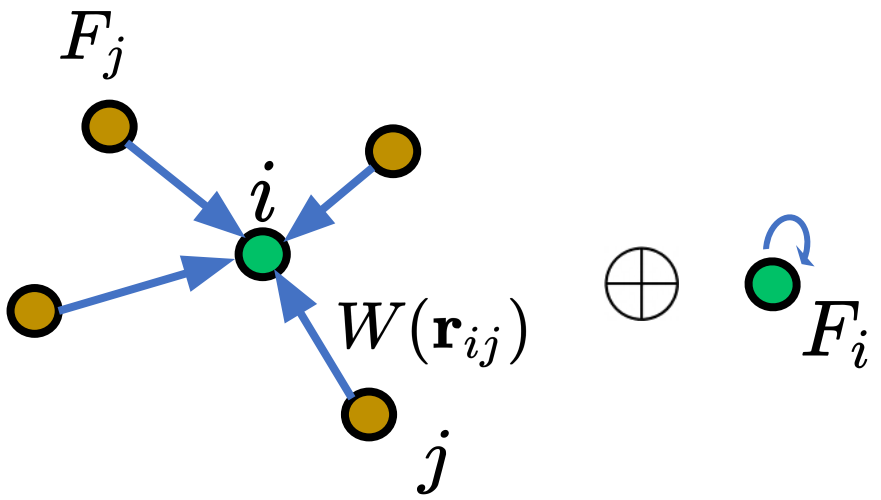
$$W(r) \rightarrow W(\mathbf{r}) = R^{(\ell)}(r) \cdot Y^{\ell}(\hat{\mathbf{r}})$$

- Convolution is Clebsch-Gordan operation

$$\sum_j W(r) F_j \rightarrow \sum_j W(\mathbf{r}_{ij}) \otimes F_j$$

Multiplication Clebsch-Gordan

- Self-interaction included



$$W(\mathbf{r}) = R^{(\ell)}(r) \cdot Y^{\ell}(\hat{\mathbf{r}})$$

Thomas, et. al., https://arxiv.org/abs/1802.08219

Tensor Field Networks (TFNs)

Key ingredients: Normalization non-linearity

Normalization is also a non-linearity:

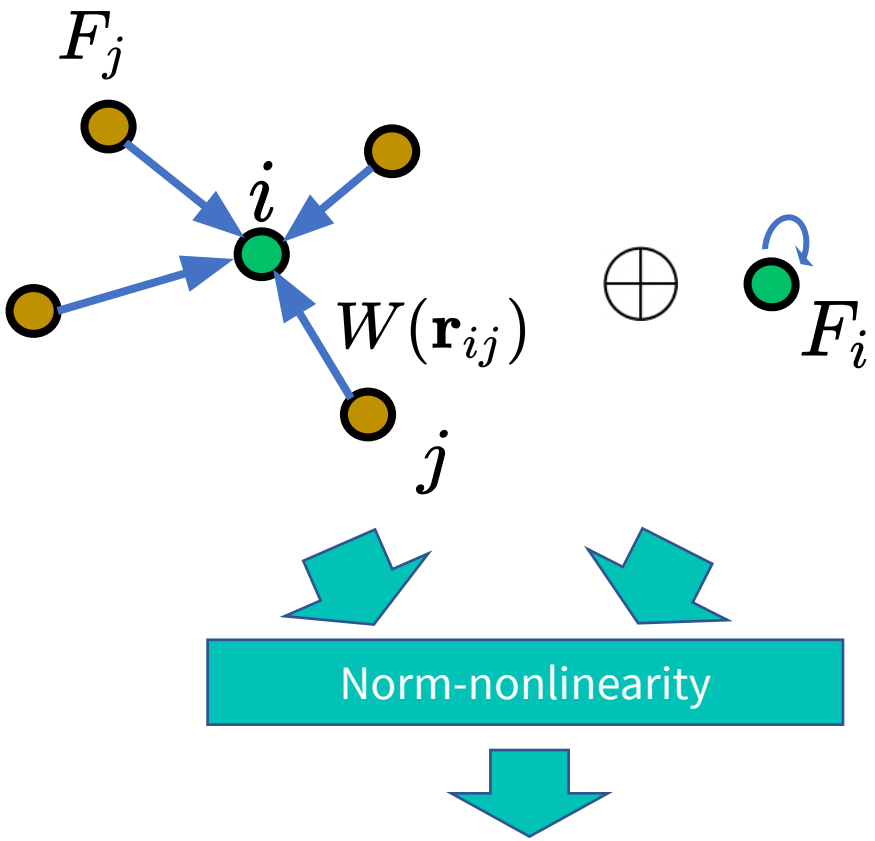
$$F_i^{(\ell)} \leftarrow \eta\left(|F_i|^{(\ell)}\right) \frac{F_i^{(\ell)}}{|F_i|^{(\ell)}}$$

Where the norm of a spherical tensor is:

$$|F|^{(\ell)} = \sqrt{\sum_{m=-\ell}^{\ell} \left|F_m^{(\ell)}\right|^2}$$

And we use a learnable function

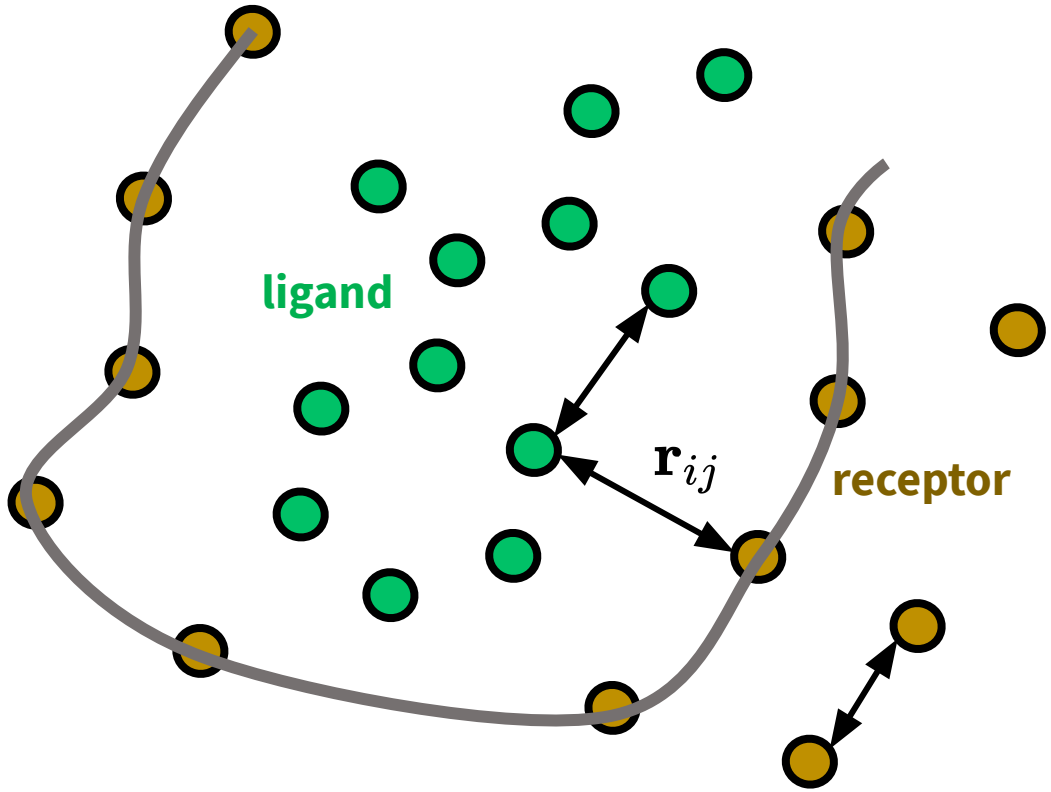
$$\eta(x)$$



Thomas, et. al., https://arxiv.org/abs/1802.08219

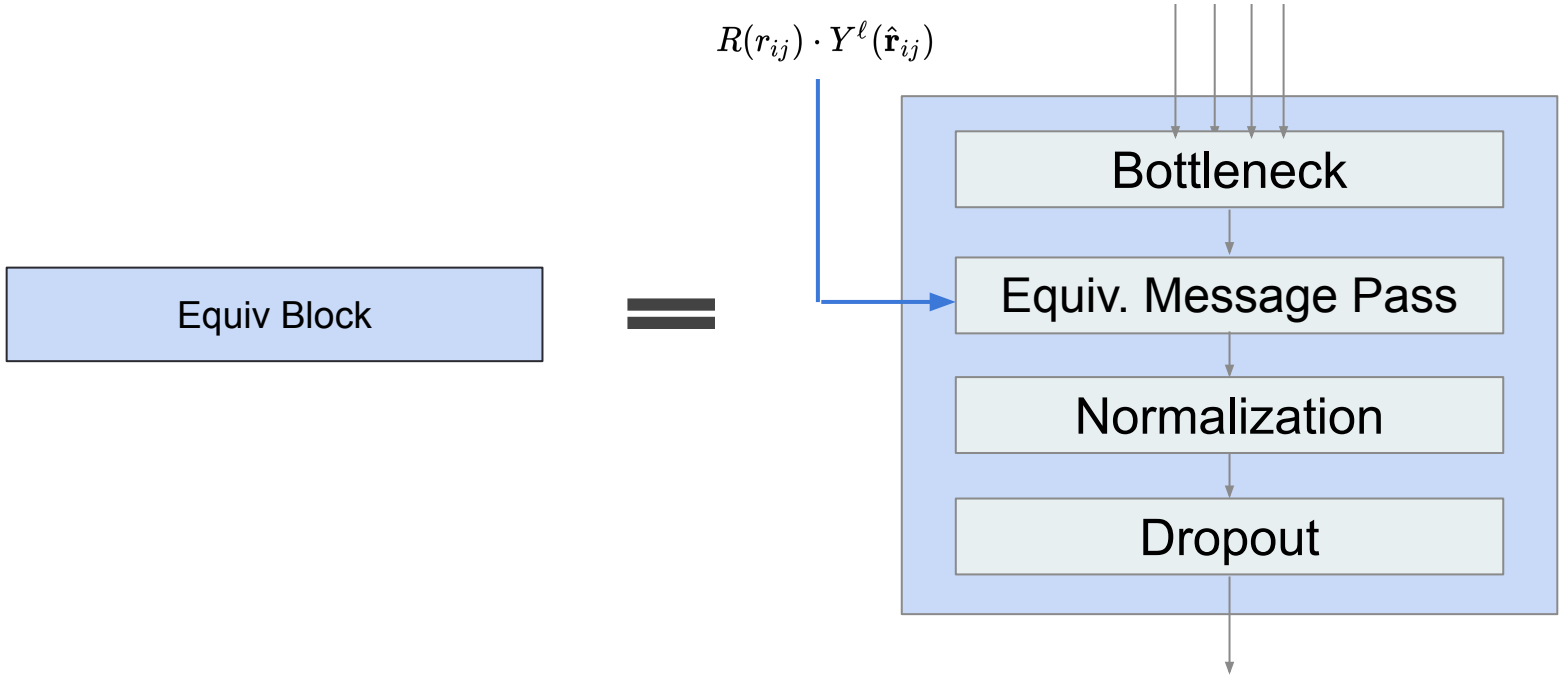
TFNs for pose classification

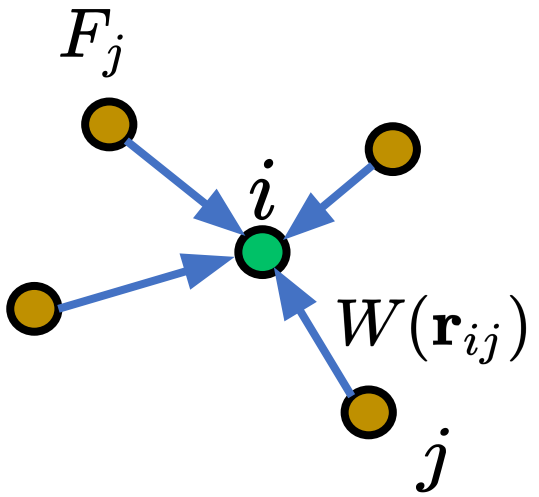
- Start with TFN ingredients
- Remove all receptor atoms > 7A from ligand
- Input embedding labels atom type and ligand/receptor
- All-to-all passing within a cutoff radius



Implementation based upon: https://github.com/FabianFuchsML/se3-transformer-public

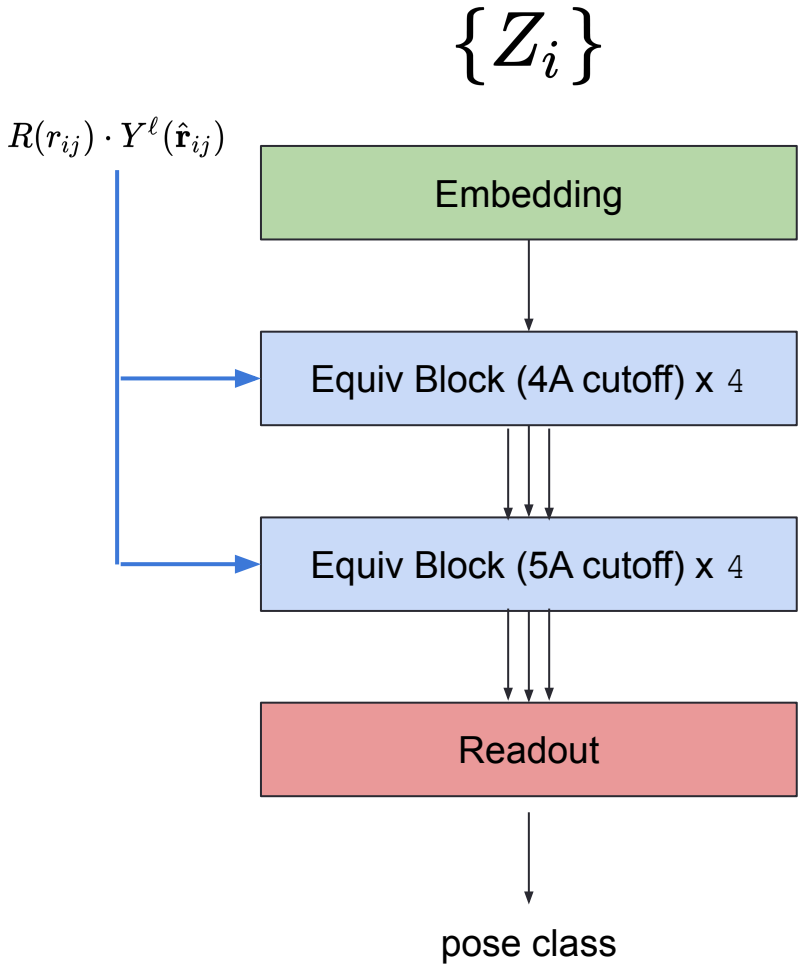
TFNs for pose classification - Equiv. block



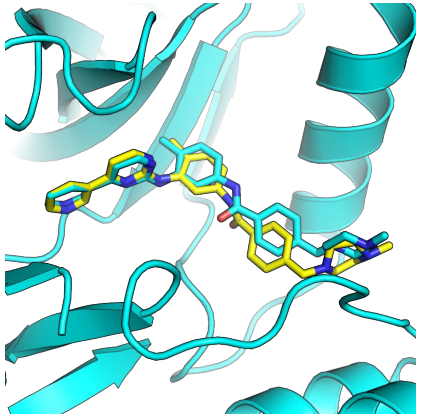


Our model

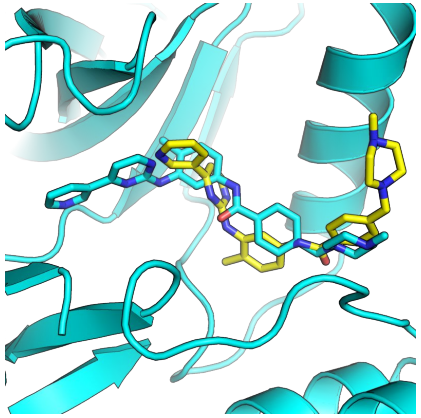
- 48 filters per layer
- *l*=(0, 1) (scalar and vector only)
- Readout:
  - All scalars (l=0) from all layers concatenated
  - Readout with a single hidden layer of size 256
- Dropout:
  - p=0.2 in pointwise filters
  - p=0.2 in normalization layer
  - p=0.2 in readout



Ligand pose classification



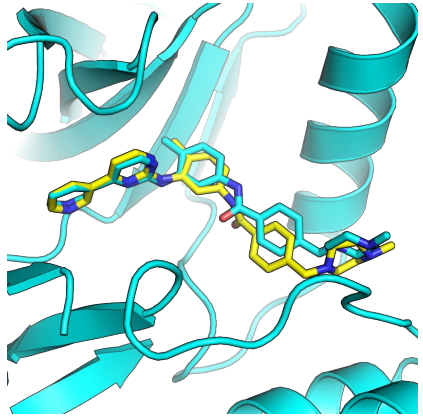
Good pose/positive example:
Docked molecule (yellow) 1.27Å RMSD to crystal structure (cyan)



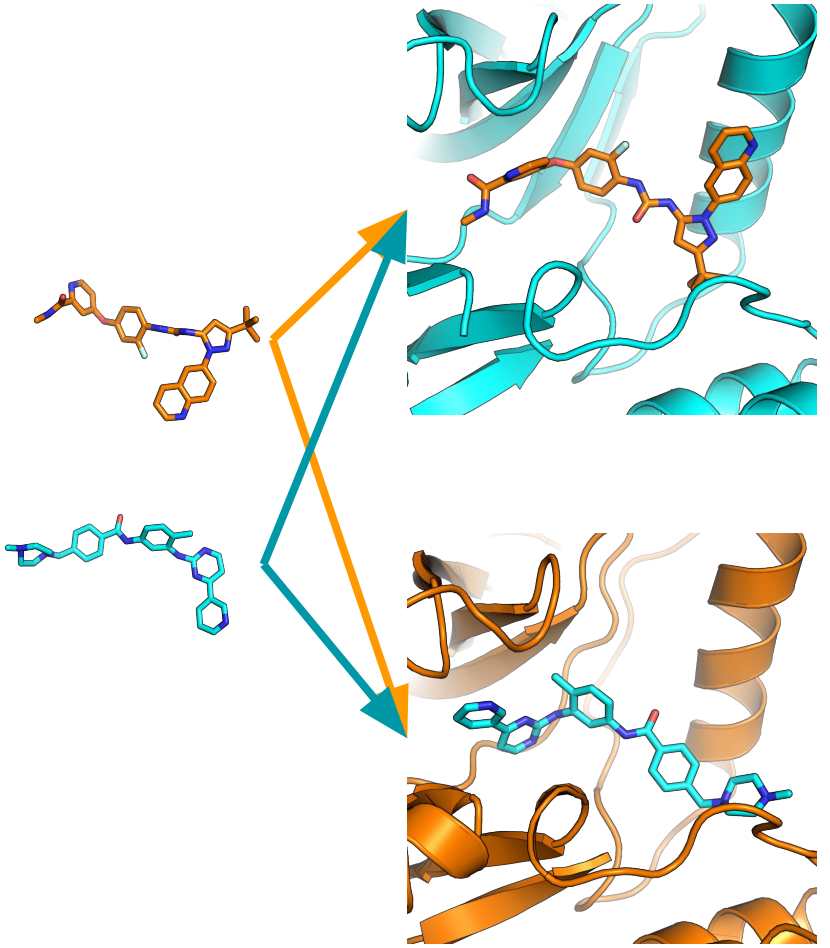
Bad pose/negative example:
Docked molecule (yellow) 6.57Å RMSD to crystal structure (cyan)

- Positive examples (good poses): ligand position <=2.5Å RMSD to the crystal structure (after receptor superposition)
- Negative examples (bad poses): ligand position >4Å RMSD

Docking modes



Self-docking:
Each ligand that has been co-crystallized with a given protein is docked to itself.



Cross-docking:
Each ligand that has been co-crystallized with a given protein is docked to every other crystal structure of the protein.

Dataset and pose sampling

- For each cluster, all ligands were docked using CUina, sampling up to 64 poses separated by at least 1Å RMSD
  - CUina: CUDA-powered docking framework, based upon Vina
  - Written by Adrian Morrison, soon to be released open source
- Dataset was prepared from the PDBBind v2019 protein-ligand set (pdbbind-cn.org) clustered by Uniprot and binding site location
- Final dataset was split into train and test sets according to the sequence similarity of the receptors, such that no receptor in the training set as more than 50% sequence identity to any receptor in the test set

Results - Self-docked

Model	AUC	logAUC**	logAUC (per-target)	top-1***	top-5***
Smina baseline			0.226	61%	86%
MPNN Baseline*	0.890	0.327	0.254	56%	85%
Equivariant Network	0.907	0.360	0.283	66%	90%

Smina Baseline: Use only Smina score for ranking
MPNN Baseline: Our best graph-neural-network on this dataset
logAUC: An early enrichment metric, variant of ROC AUC with log-linear scale.
logAUC=0 is purely random, logAUC=85.54% is perfect.
top-n: Probability the top-n poses contains a good pose (< 2.5A RMSD)

Results - Cross-Docked

Model	AUC	logAUC**	logAUC (per target)	top-1***	top-5***
Smina baseline			0.194	48%	76%
MPNN Baseline*	0.847	0.269	0.254	50%	80%
Equivariant Network	0.881	0.320	0.262	55%	83%

Smina Baseline: Use only Smina score for ranking
MPNN Baseline: Our best graph-neural-network on this dataset
logAUC: An early enrichment metric, variant of ROC AUC with log-linear scale.
logAUC=0 is purely random, logAUC=85.54% is perfect.
top-n: Probability the top-n poses contains a good pose (< 2.5A RMSD)

Summary

Equivariant networks were used to search for good poses

- Equivariance allows for collecting local geometric information

Both cross-docking and self-docking problems saw enrichment of top poses over existing baselines

Acknowledgments

Kate Stafford, Adrian Morrison, and the excellent team of scientists at Atomwise

Our Other Talks & Posters

https://info.atomwise.com/acs_spring2021

Join Us!

https://www.atomwise.com/careers/

Results - Self-docked

Model	AUC	logAUC**	top-1***	top-5***
Baseline*	0.890	0.327	56%	85%
Equivariant Network	0.907	0.360	66%	90%

Baseline: Our best graph-neural-network on this dataset
logAUC: An early enrichment metric, variant of ROC AUC with log-linear scale.
logAUC=0 is purely random, logAUC=85.54% is perfect.
top-n: Probability the top-n poses contains a good pose (< 2.5A RMSD)

Results - Cross-Docked

Model	AUC	logAUC**	top-1***	top-5***
Baseline*	0.847	0.269	50%	80%
Equivariant Network	0.881	0.320	55%	83%

Baseline: Our best graph-neural-network on this dataset
logAUC: An early enrichment metric, variant of ROC AUC with log-linear scale.
logAUC=0 is purely random, logAUC=85.54% is perfect.
top-n: Probability the top-n poses contains a good pose (< 2.5A RMSD)

Directional Message Passing

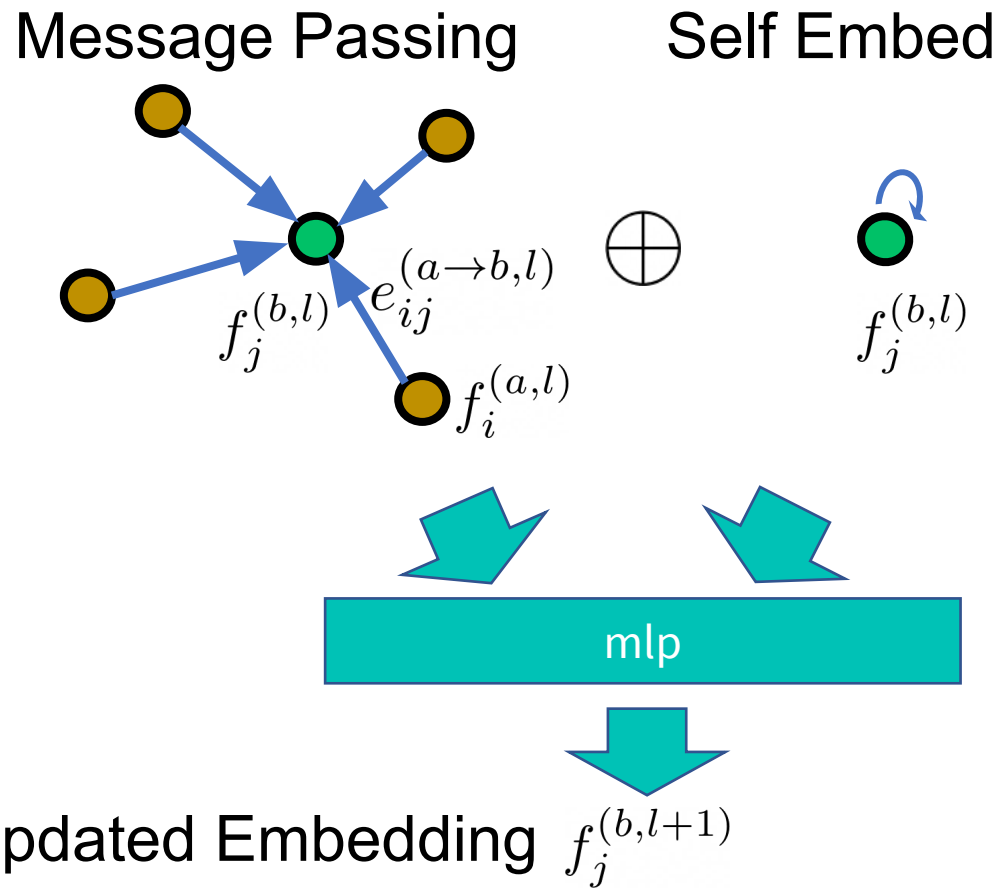
Point-based MPNN ideas

- Update messages depend on
  - Source atom embedding
  - Distance between atoms
- Messages are linear combinations of Radial Basis Functions:

$$e_{ij}^{(a\to b,l)} = \sum_c c_n^{(l)} R_n(r_{ij})$$

e.g., radial Bessel functions

$$R_n(r_{ij}) = j_0\left(\frac{z_{0n} r_{ij}}{R}\right)$$



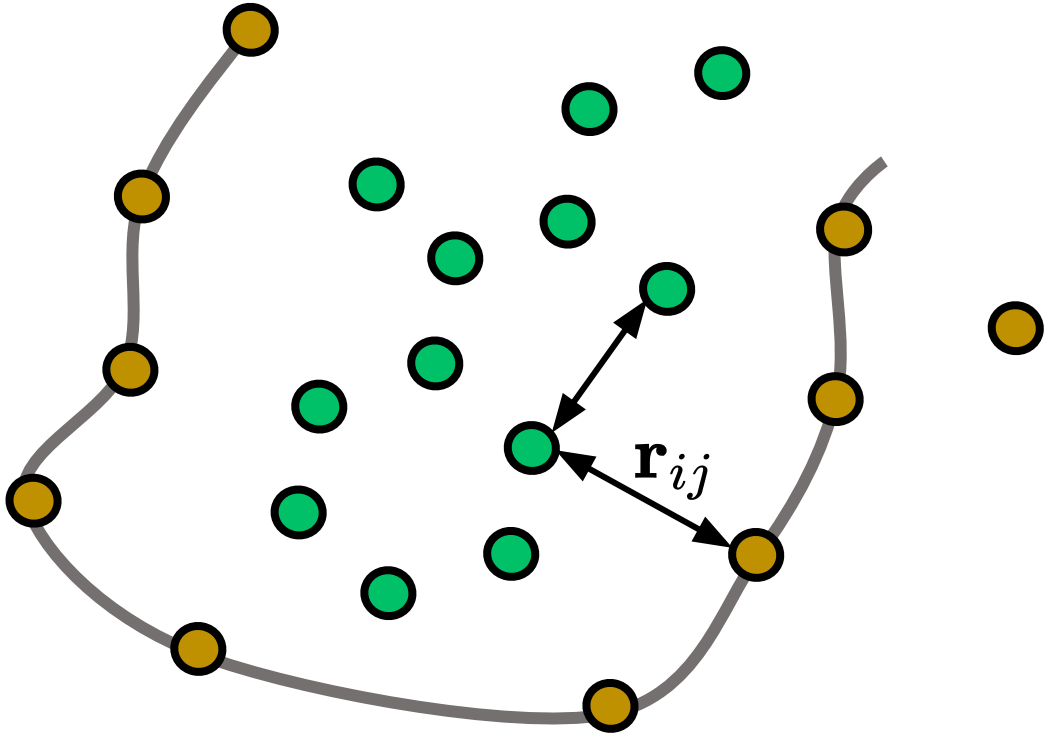
Tensor Field Networks (TFNs)

Pairwise convolution

Pairwise Convolution:

$$F_i = \sum_j \mathrm{CG}\left(W(\mathbf{r}_{ij}) \otimes F_j\right)$$

$$W(\mathbf{r}) = R^{(\ell)}(r) \cdot Y^{\ell}(\hat{\mathbf{r}})$$



Thomas, et. al., https://arxiv.org/abs/1802.08219

Model	AUC	logAUC**	top-1***	top-5***
Smina baseline	0.687	0.129		
Baseline*	0.847	0.269	50%	80%
Equivariant Network	0.881	0.320	55%	83%

Model	AUC-per-target	logAUC-per-target**	top-1***	top-5***
Smina baseline	0.687	0.129		
Baseline*	0.834	0.254	50%	80%
Equivariant Network	0.881	0.262	55%	83%