CSE7850/CX4803 Machine Learning in Computational Biology

Lecture 20: GNN for Molecular Structure

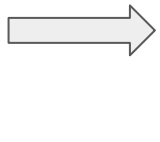
Yunan Luo

Protein Folding & Protein Design

- **Protein folding** (protein structure prediction)
  - Sequence -> Structure

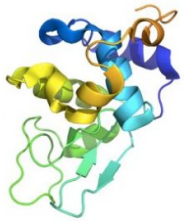
Amino acid sequence

MEKVNFLKNGVLRLPPGFRFRPTDEELVVQYLKRKVFSFPLPASIIPEVEVYKSDPWDLPGDMEQEKYFFSTK
EVKYPNGNRSNRATNSGYWKATGIDKQIILRGRQQQQQLIGLKKTLVFYRGKSPHGCRTNWIMHEYRLAN
LESNYHPIQGNWVICRIFLKKRGNTKNKEENMTTHDEVRNREIDKNSPVVSVKMSSRDSEALASANSELKK



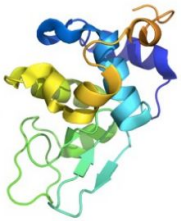
Algorithm / Model

Protein structure



- **Protein design**
  - Structure -> Sequence

Protein structure



Algorithm / Model

Amino acid sequence

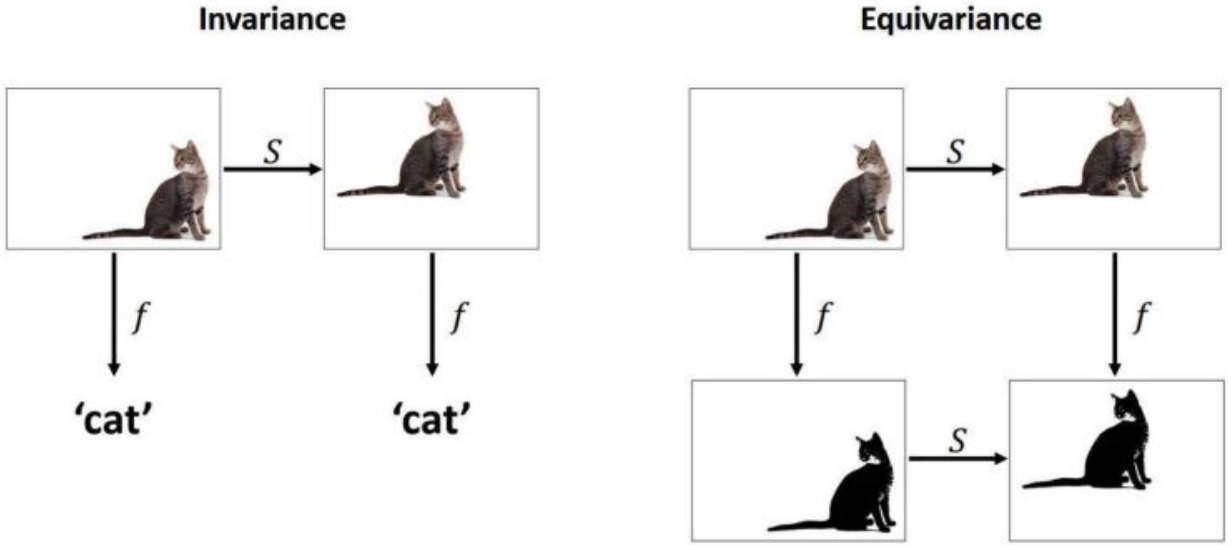
MEKVNFLKNGVLRLPPGFRFRPTDEELVVQYLKRKVFSFPLPASIIPEVEVYKSDPWDLPGDMEQEKYFFSTK
EVKYPNGNRSNRATNSGYWKATGIDKQIILRGRQQQQQLIGLKKTLVFYRGKSPHGCRTNWIMHEYRLAN
LESNYHPIQGNWVICRIFLKKRGNTKNKEENMTTHDEVRNREIDKNSPVVSVKMSSRDSEALASANSELKK

Two GNN papers today

- Paper #1: Invariant GNN
- Paper #2: Equivariant GNN

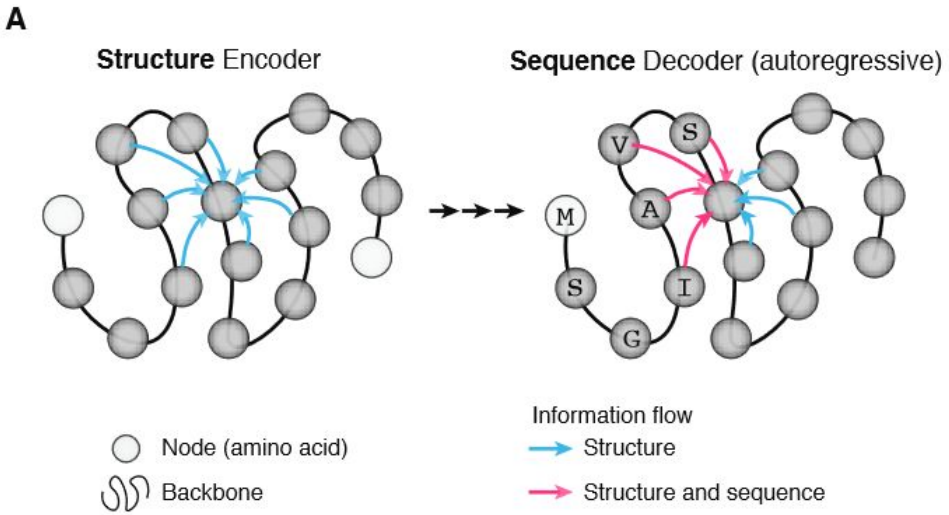
Invariance & Equivariance

- The analogy in image domain
  - Classification: invariant label
  - Segmentation: equivariant pixel coordinates



https://www.doc.ic.ac.uk/~bkainz/teaching/DL/notes/equivariance.pdf

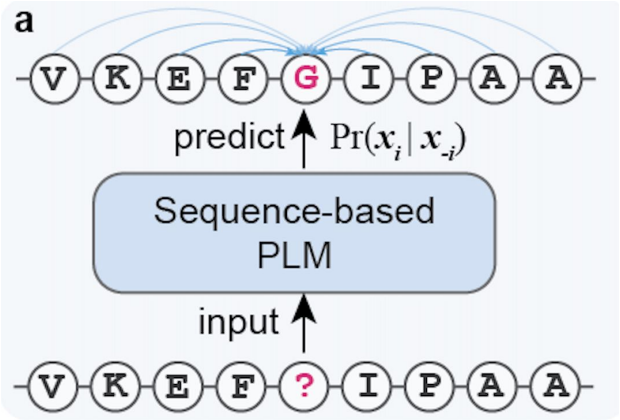
Paper #1: GNN for structure-based protein design



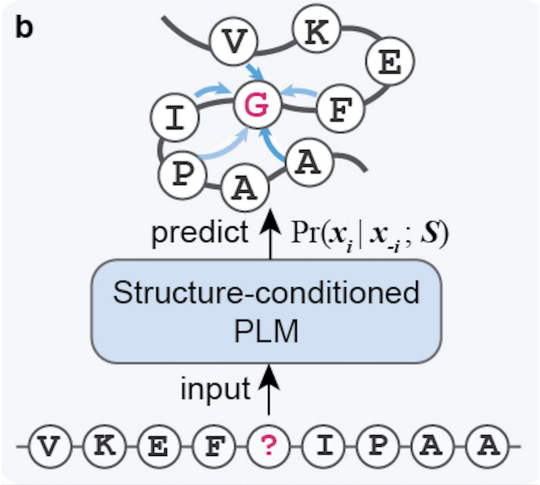
Ingraham et al. “Generative models for graph-based protein design”, NeurIPS, 2019

Structure-based PLM

Traditional PLM (sequence-based)



Structure-based PLM

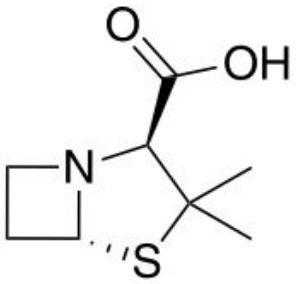


Invariant GNN

3D considerations For a rigid-body design problem, the structure for conditioning is a fixed set of backbone coordinates $\mathcal{X} = \{\boldsymbol{x}_i \in \mathbb{R}^3 : 1 \leq i \leq N\}$, where N is the number of positions[1]. We desire a graph representation of the coordinates $\mathcal{G}(\mathcal{X})$ that has two properties:

- *Invariance*. The features are invariant to rotations and translations.
- *Locally informative.* The edge features incident to $\boldsymbol{v}_i$ due to its neighbors $\mathrm{N}(i)$, i.e. $\{\boldsymbol{e}_{ij}\}_{j \in \mathrm{N}(i)}$, contain sufficient information to reconstruct all adjacent coordinates $\{\boldsymbol{x}_j\}_{j \in \mathrm{N}(i)}$ up to rigid-body motion.

Molecule structure

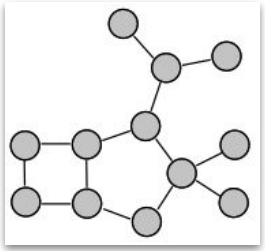


Molecule

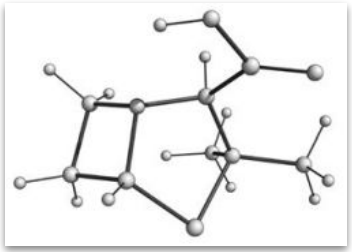
- 1D representation (SMILES string)

```
CC1(C)[C@H](C(O)=O)N2[C@@H](CC2)S1
```

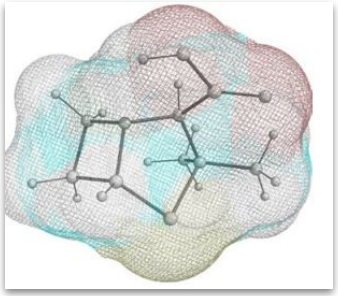
- 2D representation (graph)



- 3D representation (coordinates)



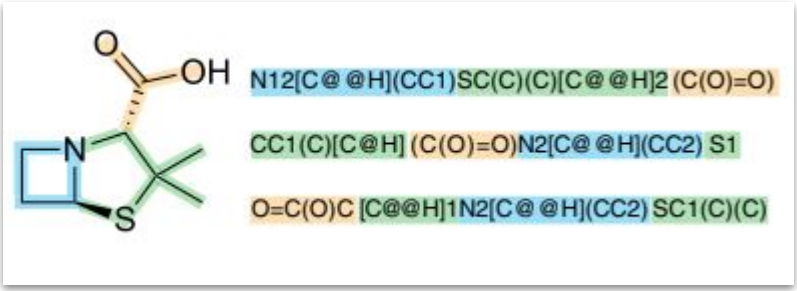
- Surface representation (mesh)



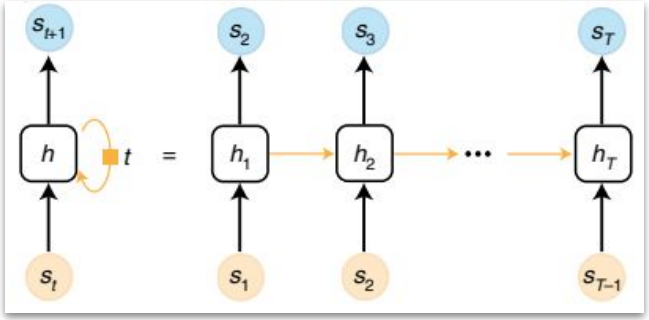
Atz et al., *Nature Machine Intelligence*, 2021

Deep learning for molecule structure (1D)

SMILES string: linear representation of a molecule



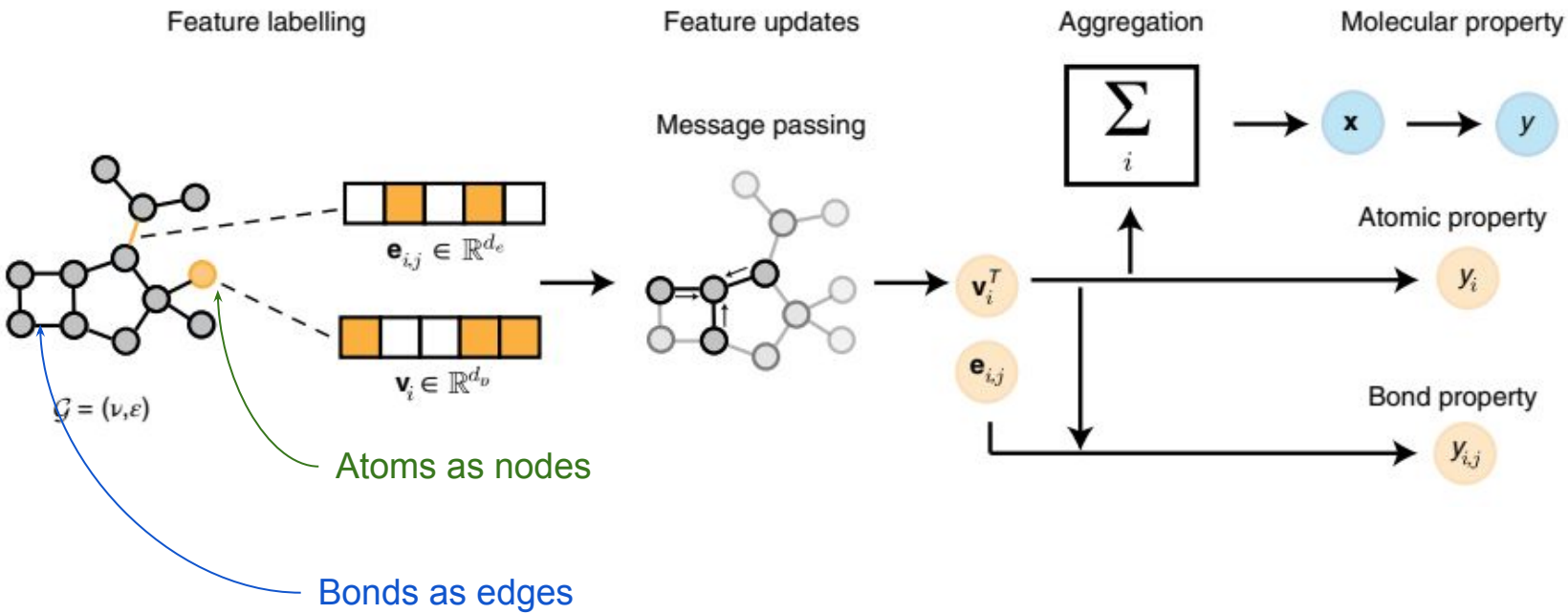
Example: RNN-based models



Atz et al., *Nature Machine Intelligence*, 2021

Deep learning for molecule structure (2D)

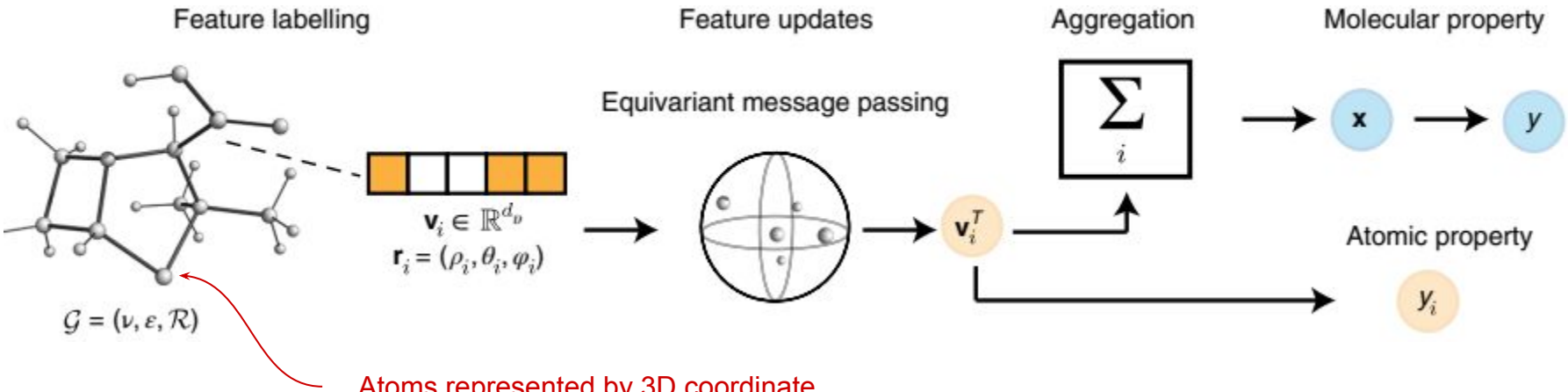
Example: Graph neural network (GNN)



Atz et al., *Nature Machine Intelligence*, 2021

Deep learning for molecule structure (3D)

Example: *Equivariant* Graph neural network



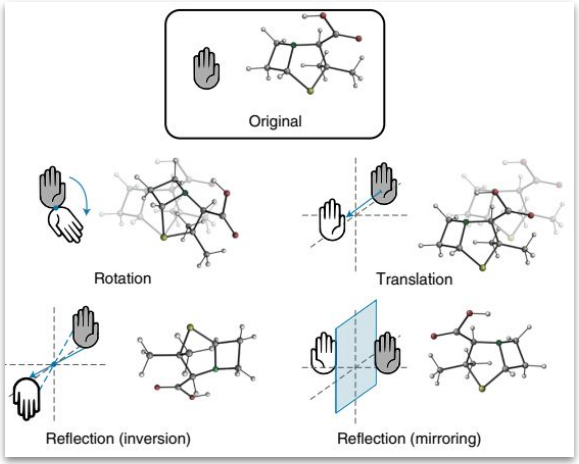
Atz et al., *Nature Machine Intelligence*, 2021

Why invariance and equivariance?

- ***X***: input molecule
- ***F***: neural network
- ***T***: transformation (e.g., rotation, translation, reflection)

- Invariance
  - ***F***(***T***(***X***)) = ***F***(***X***)
    - Output remains the same no matter how the input is rotated, shifted, etc
  - Motivation: many molecular descriptors are invariant to the rotation and translation of the molecular representation

- Equivariance
  - ***F***(***T***(***X***)) = ***TF***(***X***)
    - Output changes in the same way as the input
  - Motivation: some property changes following a symmetry transformation (e.g., chiral properties that change under reflection of the molecule)



Atz et al., *Nature Machine Intelligence*, 2021