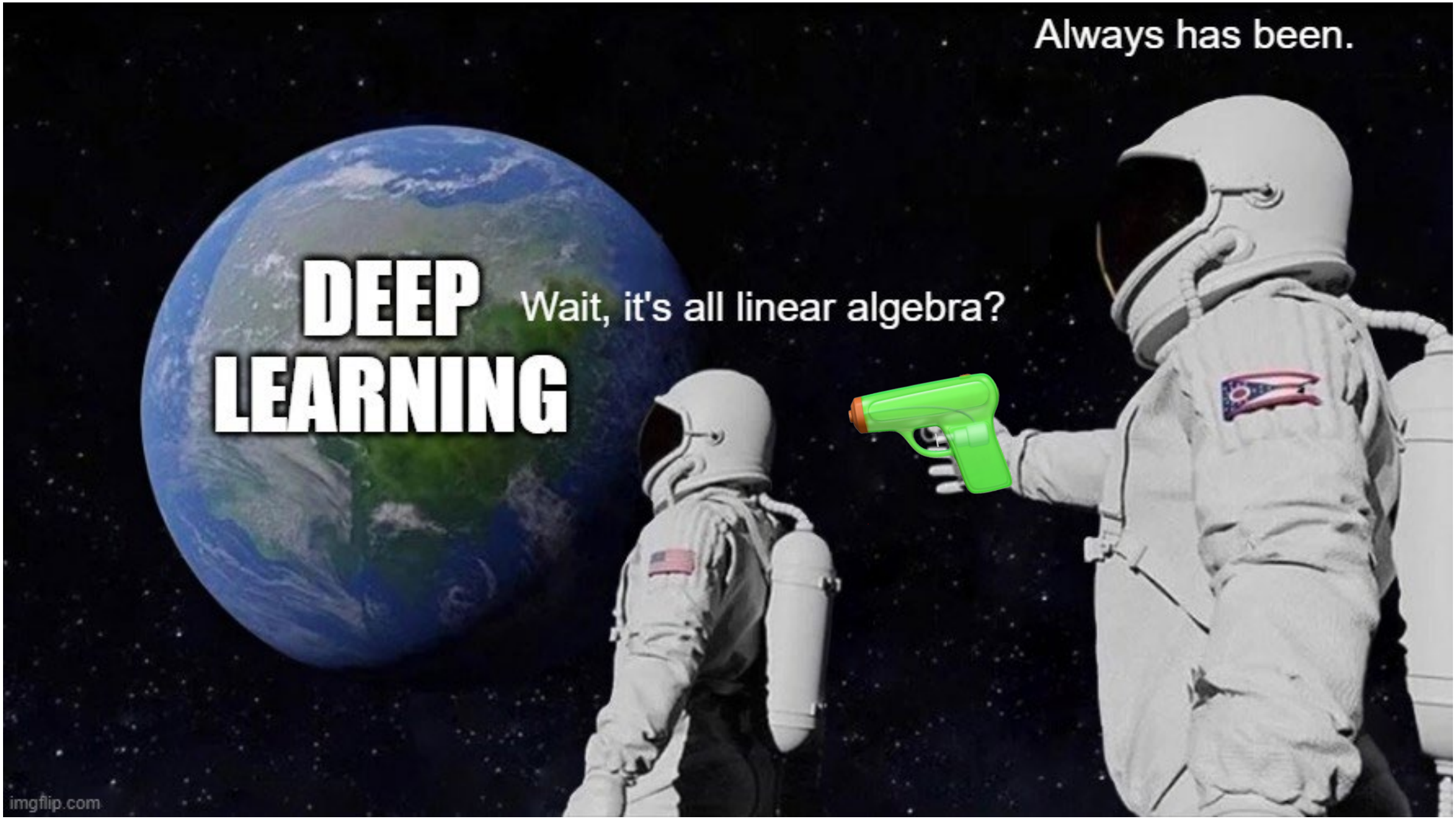
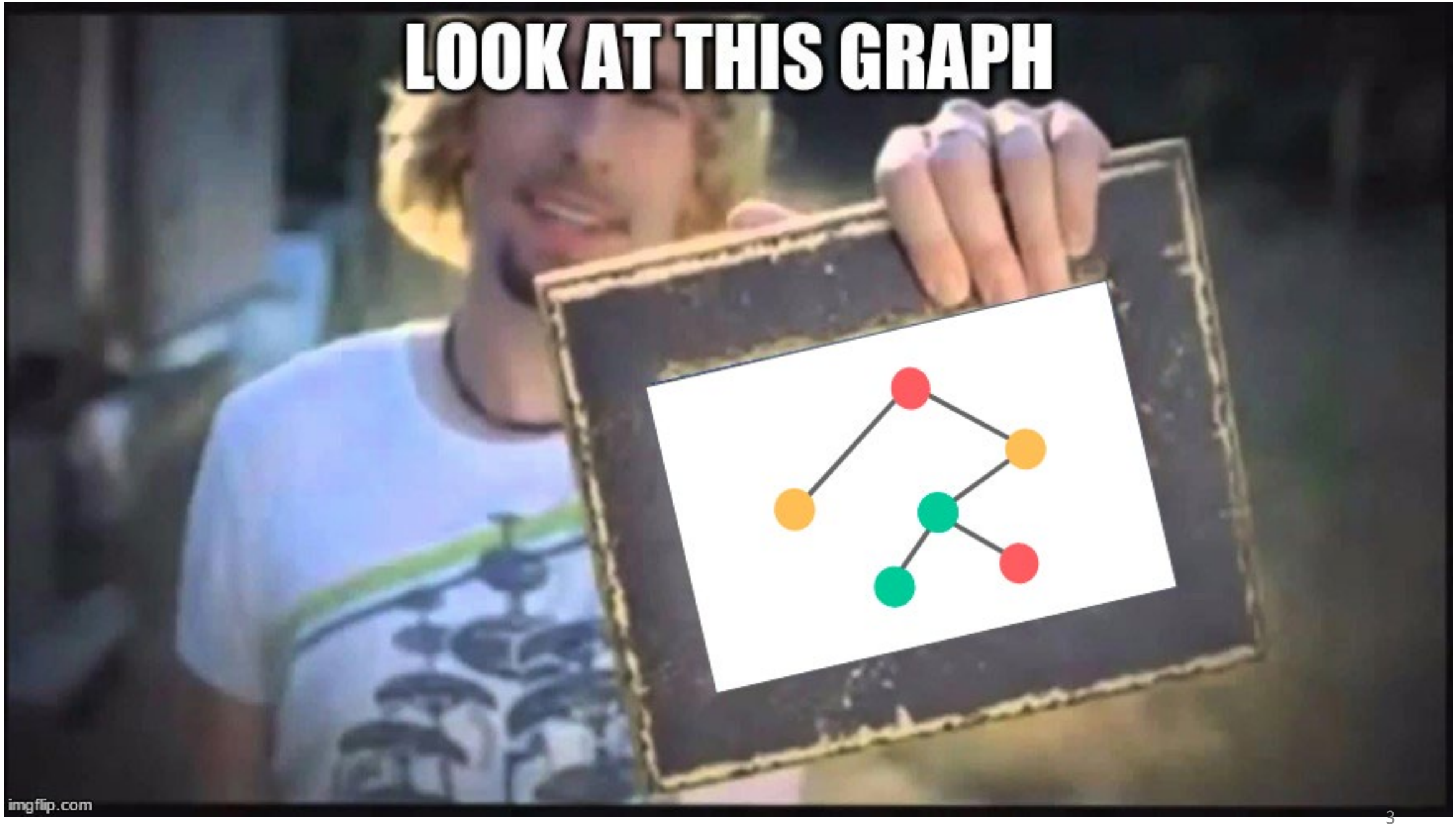
Graph Neural Networks for Molecular Property Prediction

Pablo Curiel

Department of Applied Mathematics, UC Merced


Always has been.
DEEP LEARNING
Wait, it's all linear algebra?
imgflip.com


LOOK AT THIS GRAPH
imgflip.com

About Me

- Obtained B.S. in Applied Math from CSU Chico (May 2023)
  - Minor in Computer Science
- First-year Applied Math Ph.D. student
- Research interests: mathematical biology, machine learning, graph theory
- Enjoy running, playing sports, spending time with friends/family

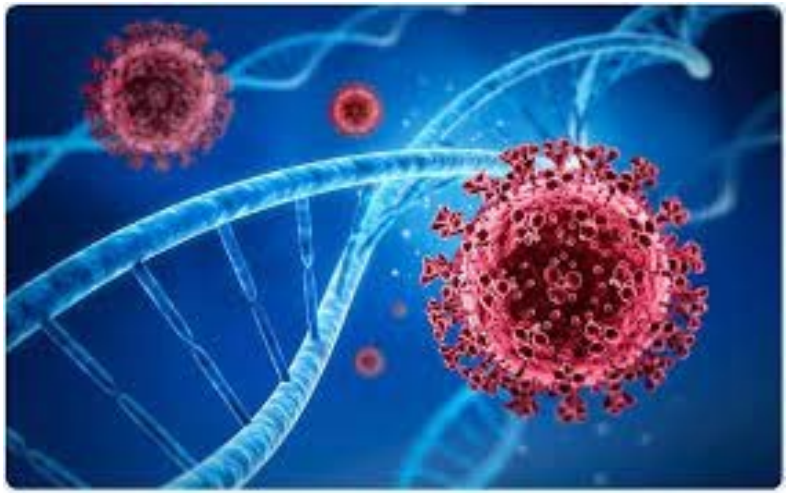
Outline

- Introduction
- Methods
- ChemProp
- Experiments
- Results and Discussion
- Conclusions and Future Work

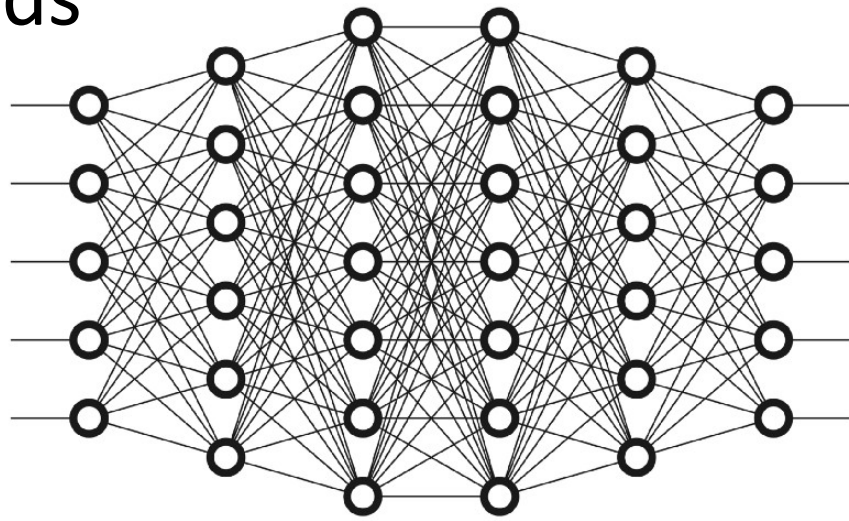
Outline

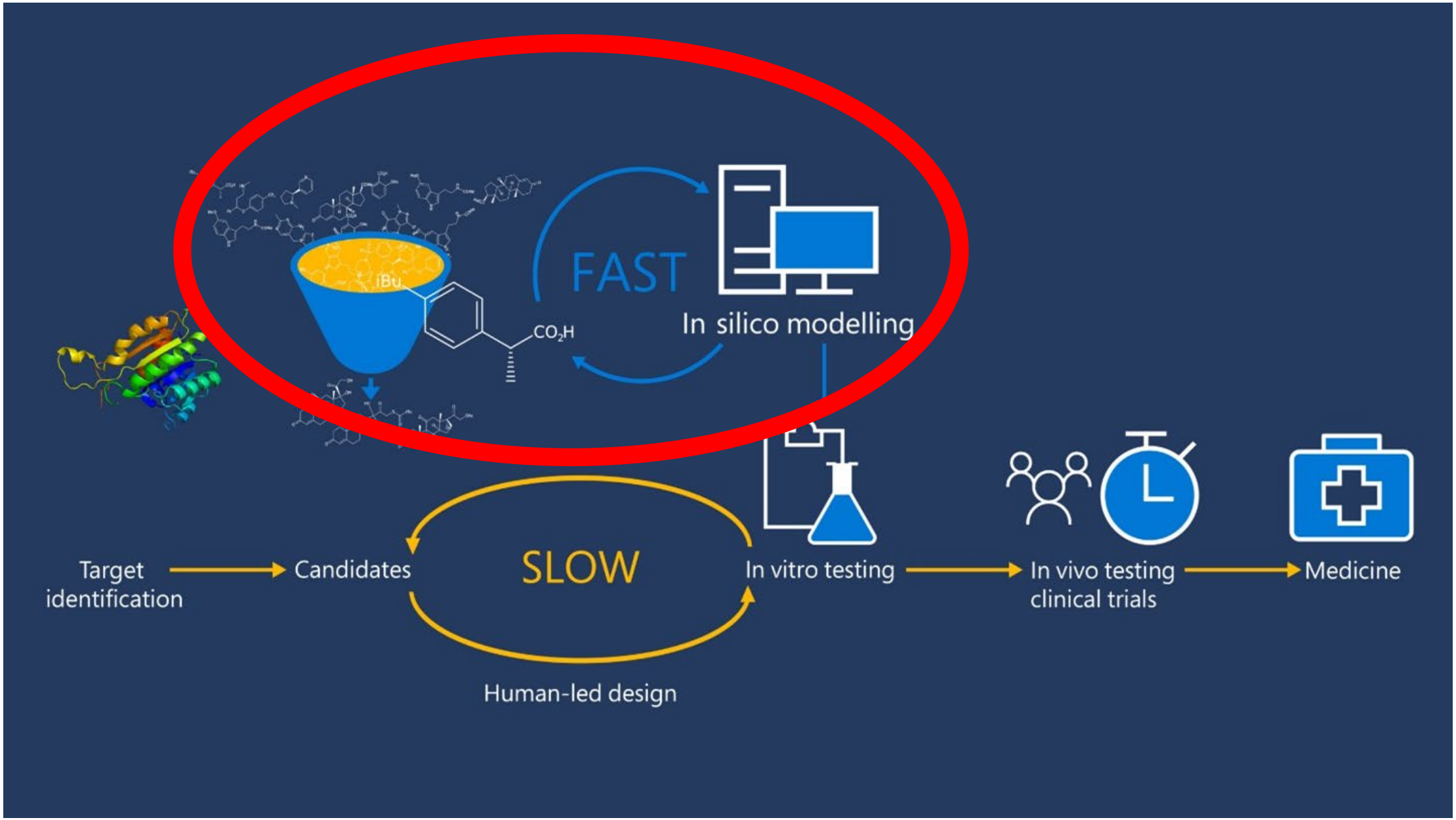
- **Introduction**
- Methods
- ChemProp
- Experiments
- Results and Discussion
- Conclusions and Future Work

Introduction



- Molecular property prediction (MPP)
  - Longstanding problem in biochemical/biomedical fields
- Drug design/discovery: process of identifying or generating drug-like molecular compounds
  - e.g., properties, structure, novelty, etc.
- Recent computational advancements have boosted prediction performance
  - i.e., GPUs, deep learning, etc.

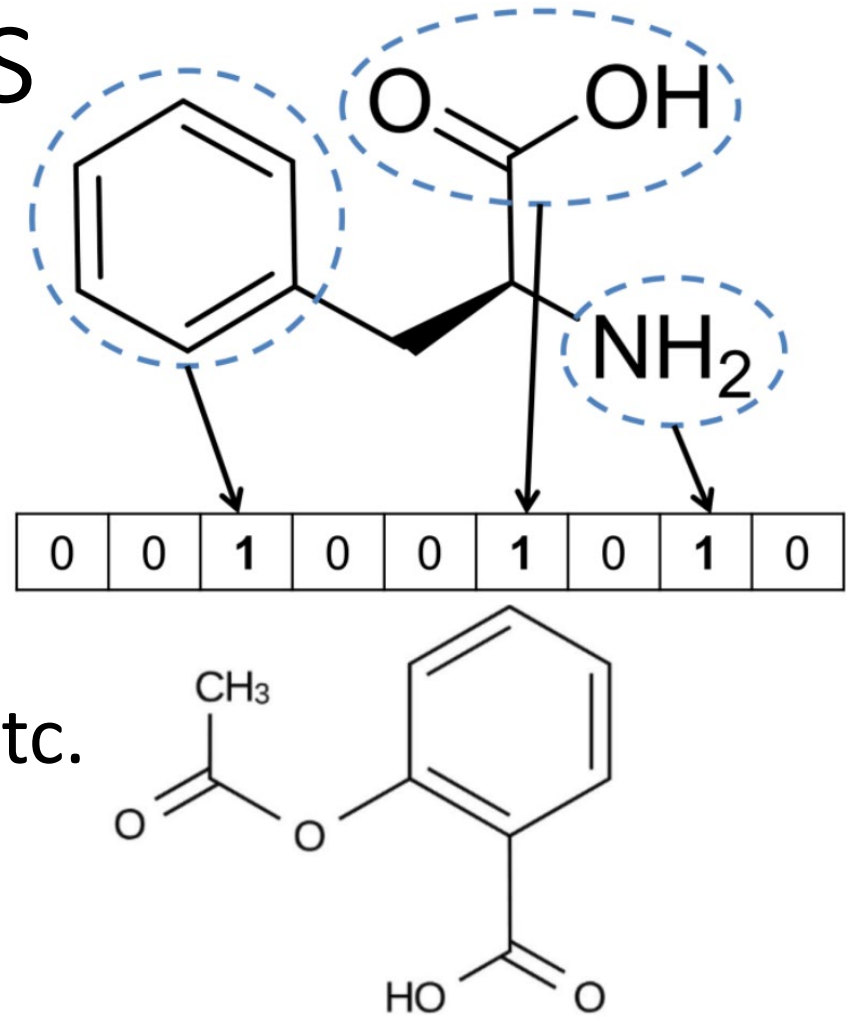




Kirkpatrick et al. "Chemical Space.“ (2004)
Maziarz et al. “MoLeR: Creating a path to more efficient drug design” (2022).

Representing Molecules

- Fingerprints
  - Based on molecular substructures
- Descriptors
  - Expert-engineered features
  - e.g., molecular weight, # rings/cycles, etc.
- Graphs
  - Converting SMILES strings to graphs



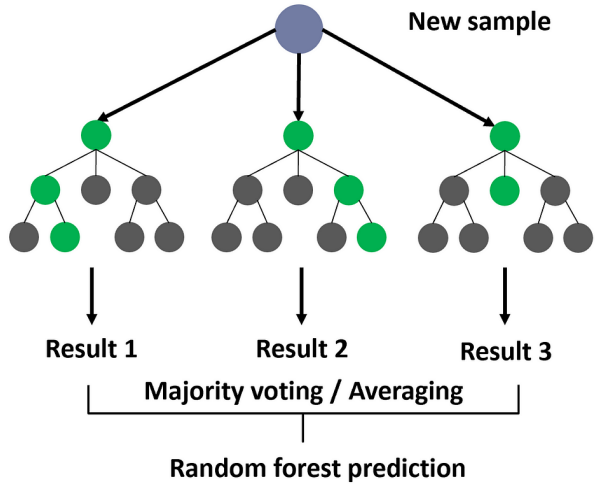
Acetylsalicylic Acid (Aspirin)

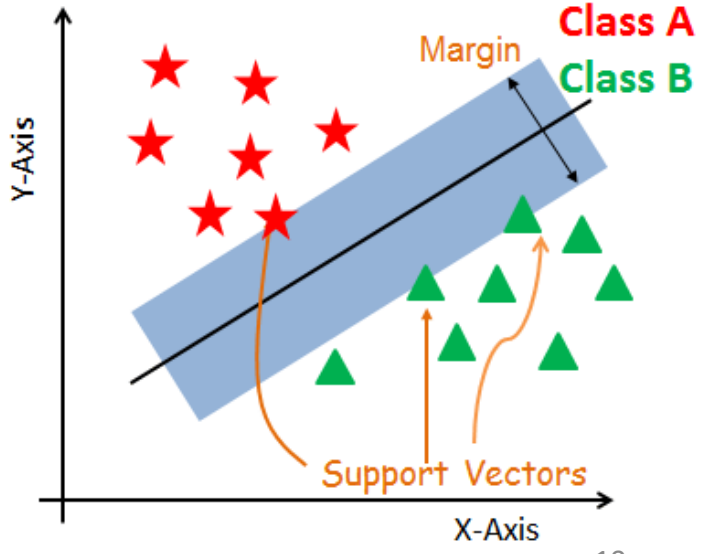
CC(=O)Oc1ccccc1C(=O)O

Bajusz et al. "Chemical Data Formats, Fingerprints, and Other Molecular Descriptions for Database Analysis and Searching." (2017)
Yasonik, Jacob. "Multiobjective de novo drug design with recurrent neural networks and nondominated sorting" (2020)

Traditional Approaches to MPP

- Based on **fixed**, hand-engineered molecular-level features
  - Feature vectors often become high-dimensional
  - Fixed feature vectors constrain model's ability to generalize
- Use classical ML methods
  - i.e., random forests, SVMs, etc.

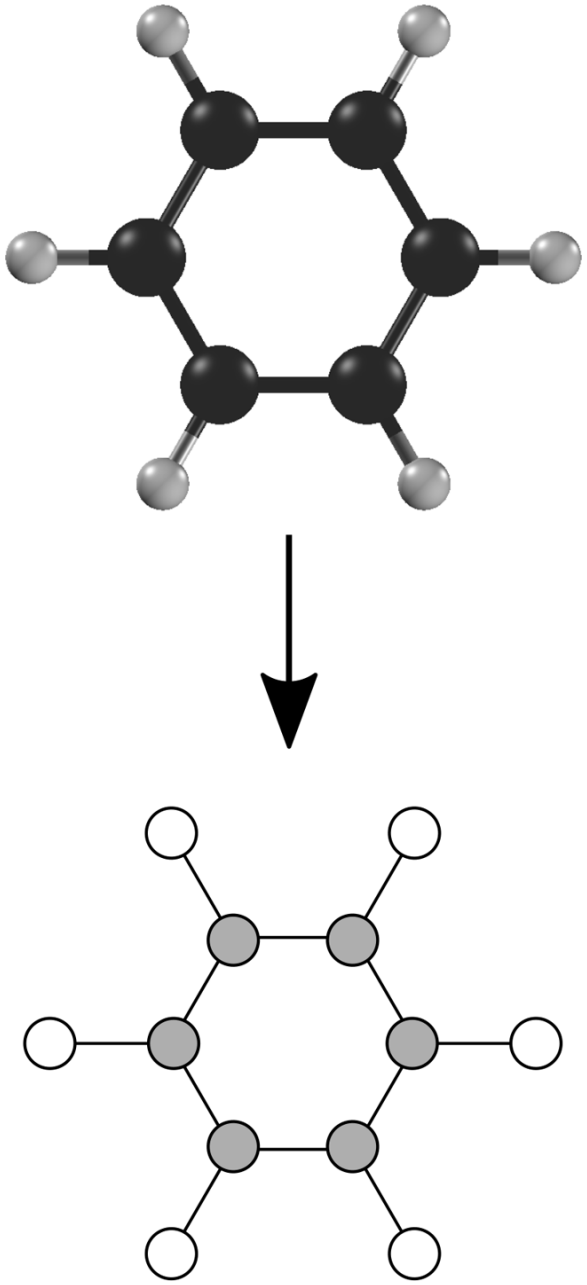




Wu et al. "MoleculeNet: a benchmark for molecular machine learning." (2017)

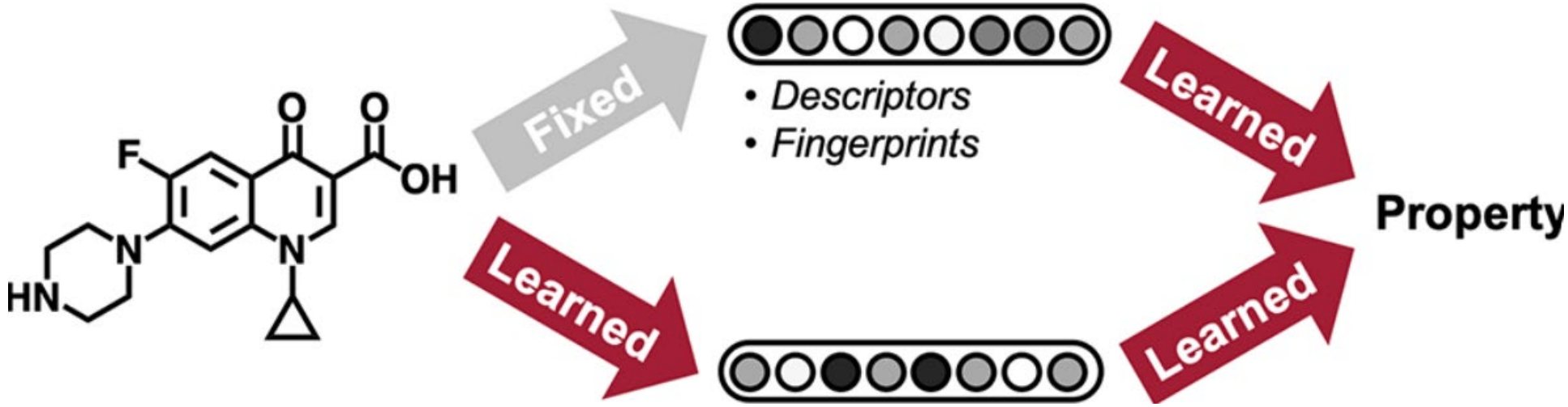
Molecules as Graphs

- Molecules have a graph-like structure
  - Nodes = atoms, edges = chemical bonds
- RDKit package used to obtain graph representation from SMILES string
- Graph representation can be used for GNN



Why GNNs?

- Classical models use **fixed**, expert-engineered features
  - High-dimensionality => increased computational cost
  - May not generalize well
- GNNs allow for molecular features to be **learned**

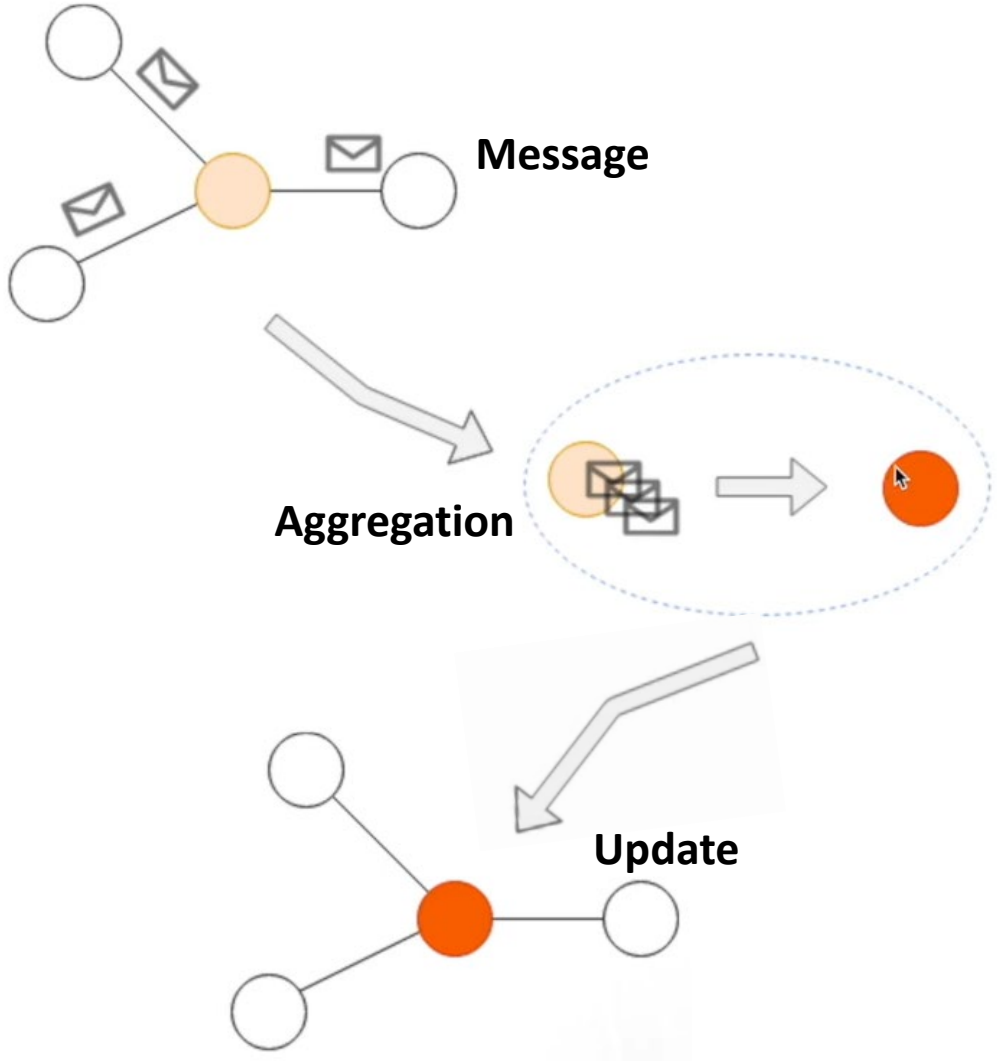


Outline

- Introduction
- **Methods**
- ChemProp
- Experiments
- Results and Discussion
- Conclusions and Future Work

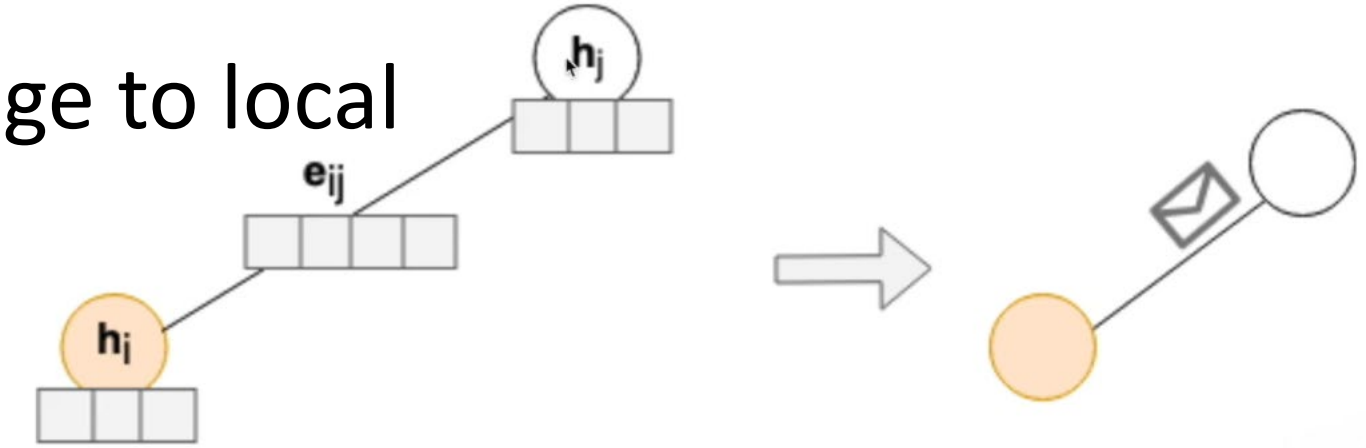
Message Passing Neural Network (MPNN)

- GNN model that transmits information across edges to update node representations
  - Type of graph convolutional NN (GCN)
- Three main functions:
  - Message
  - Aggregation
  - Update



Message Function $\mathbf{m}_{ij}^{(k)} = \mathcal{M}(\mathbf{h}_i^{(k)}, \mathbf{h}_j^{(k)}, \mathbf{e}_{ij})$

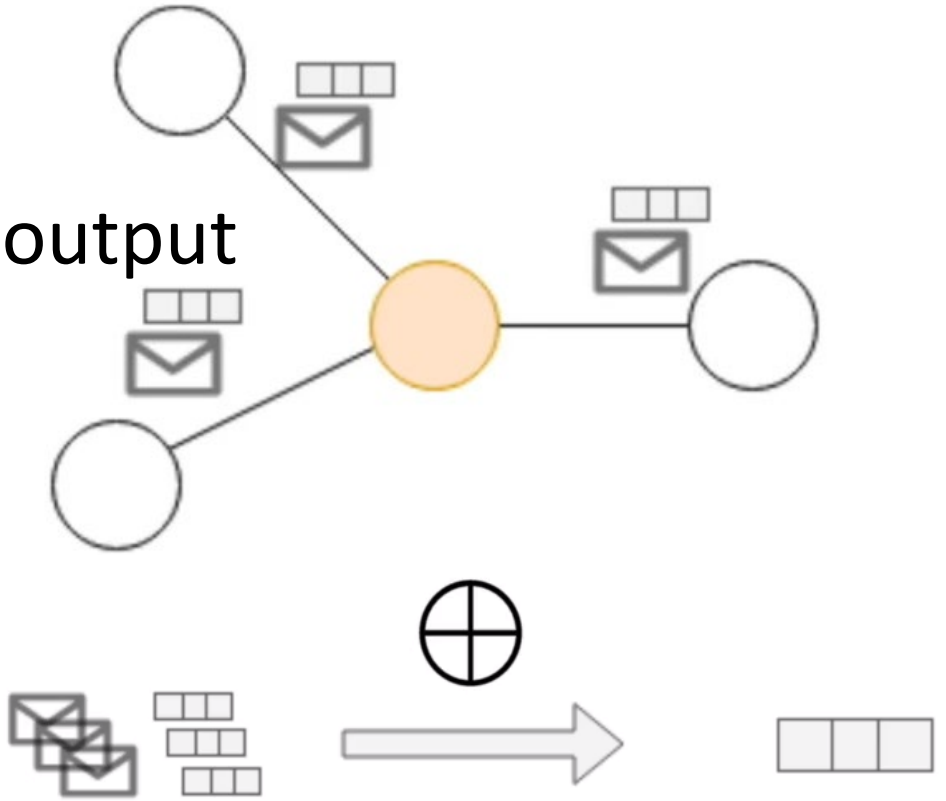
- Computes message based on node features
- Sends computed message to local one-hop neighborhood
- Message examples:
  - **Exact copy**
  - Normalized based on node degree



Aggregation Function

$$\hat{\mathbf{m}}_i^{(k)} = \bigoplus_{j \in \mathcal{N}_i} \mathbf{m}_{ij}^{(k)}$$

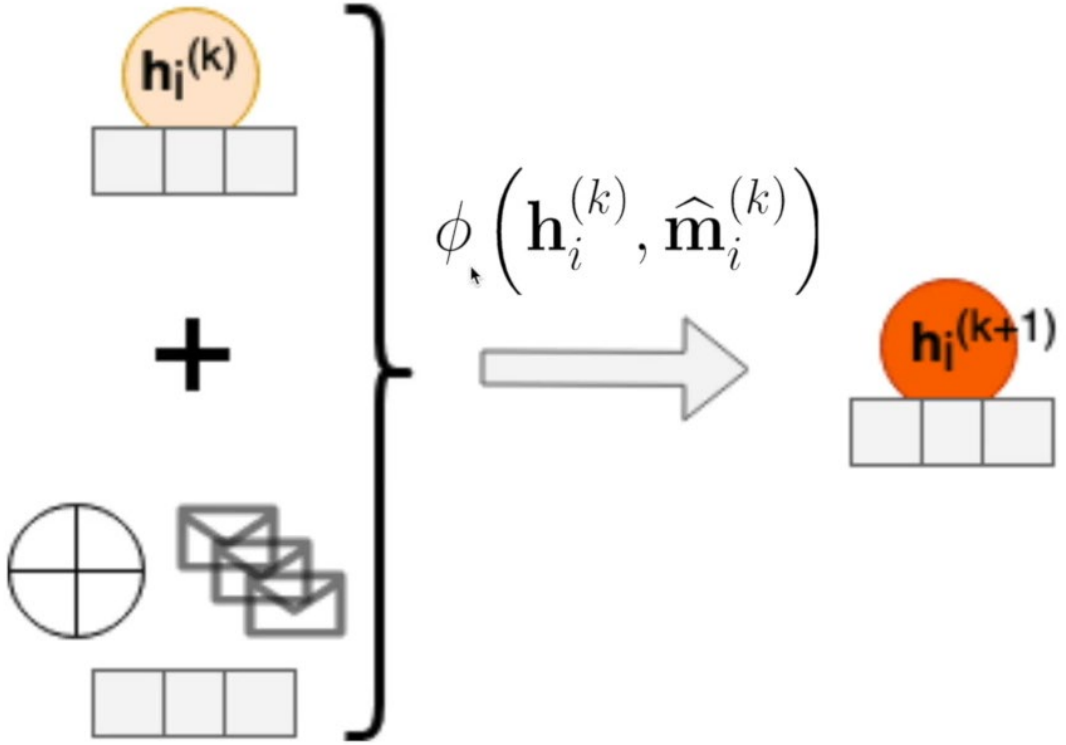
- Combines neighborhood messages into a **fixed-length** vector
- Must be **permutation invariant**
  - i.e. reordering input produces same output
- Aggregation examples:
  - **Sum**
  - Average
  - Max



Update Function

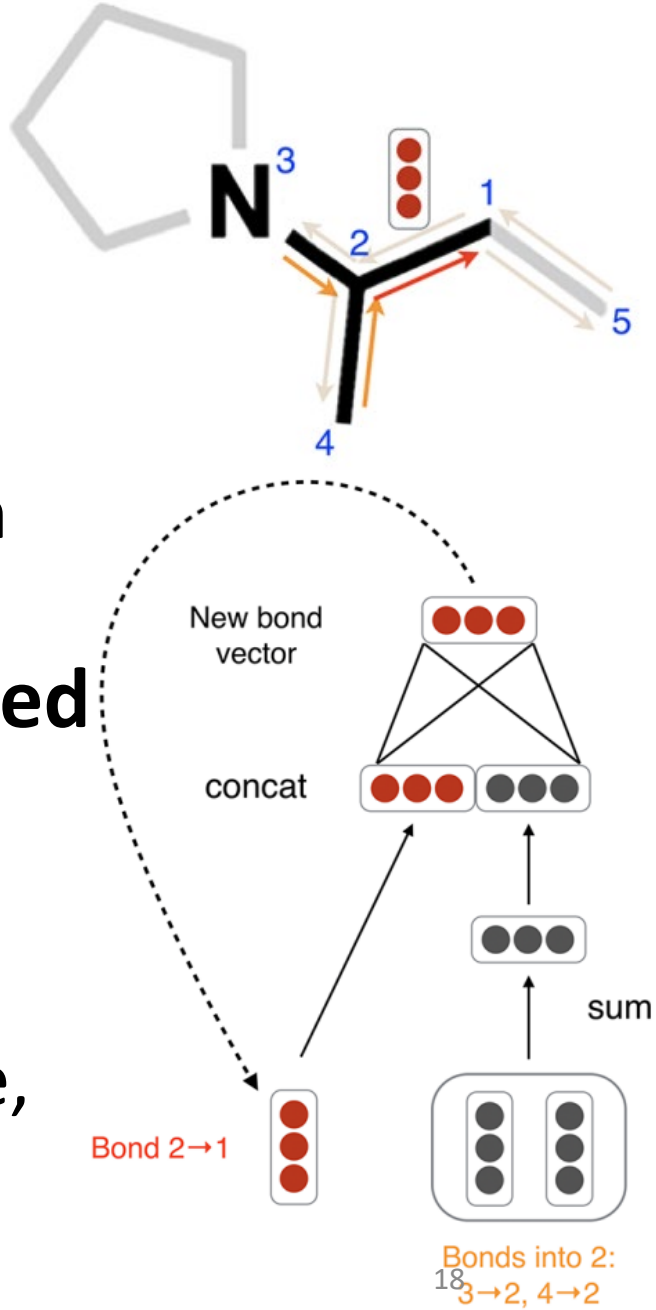
$$\mathbf{h}_i^{(k+1)} = \phi\left(\mathbf{h}_i^{(k)}, \widehat{\mathbf{m}}_i^{(k)}\right)$$

- Computes new node embedding using old embedding and aggregated messages
- **Update process**:
  - Apply linear projection (weights) to concatenated old embedding and agg. messages
  - Apply nonlinearity to each entry in result
    - e.g., **ReLU**, tanh, etc.



Directed-MPNN (D-MPNN)

- Key difference: **directed** edge-based messages instead of node-based
  - Avoids added noise in final graph representation
    - Aka “totters”
- Final states are summed to obtain a final **learned** molecular representation $\mathbf{z}^{(T)}$
  - $\mathbf{z}^{(T)}$ is fed to a standard NN for a desired task
- Initial features calculated using RDKit
  - e.g., atom/bond type, bond in ring, atom degree, etc.



D-MPNN: Model Enhancements

- Incorporating known features to model
  - i.e., fingerprints, descriptors, etc.
- Hyperparameter optimization: “Bayesian Optimization” applied via Hyperopt package in Python
- Ensembling: training N models independently and picking best model

Outline

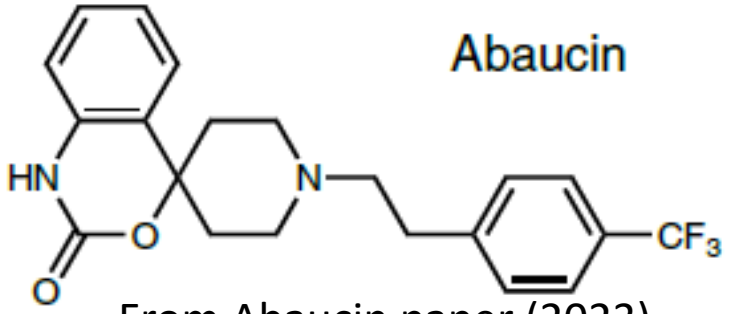
- Introduction
- Methods
- **ChemProp**
- Experiments
- Results and Discussion
- Conclusions and Future Work

ChemProp

- Python package with D-MPNN for MPP
- Contains several built-in model enhancements
  - i.e., multiple molecules as input, ensembling, hyperparameter optimization, etc.
- Currently being researched with and updated

ChemProp: Prior Works

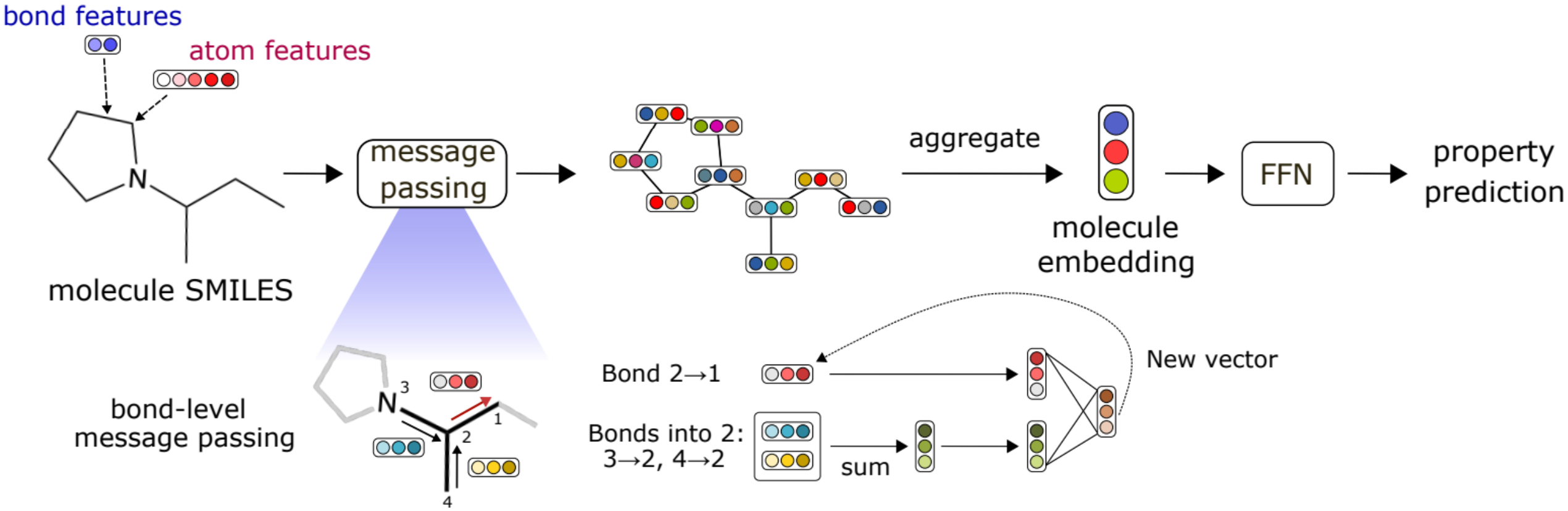
- 2019: first appearance of (unnamed) D-MPNN with several performance boosts
- 2020: ChemProp used to discover **Halicin**
  - Structurally novel compound with many antibacterial properties
  - Promising wet lab results
- 2023: ChemProp used to identify **Abaucin**
  - Novel growth inhibitor against Acinetobactor Baumannii



From Abaucin paper (2023)

Stokes et al. "A Deep Learning Approach to Antibiotic Discovery." (2020)
Liu et al. "Deep learning-guided discovery of an antibiotic targeting Acinetobacter baumannii." (2023)

ChemProp: Pipeline



Heid et al. "Chemprop: A Machine Learning Package for Chemical Property Prediction." (2017)

ChemProp: Demo

- Tox21 dataset: from “Toxicology in the 21st century” challenge (2014)
  - Hosted by NIH, EPA, and FDA
- Problem: classify molecules as being toxic or not towards various nuclear/stress receptors using ChemProp

https://tripod.nih.gov/tox21/challenge/

Outline

- Introduction
- Methods
- ChemProp
- **Experiments**
- Results and Discussion
- Conclusions and Future Work

Experiments

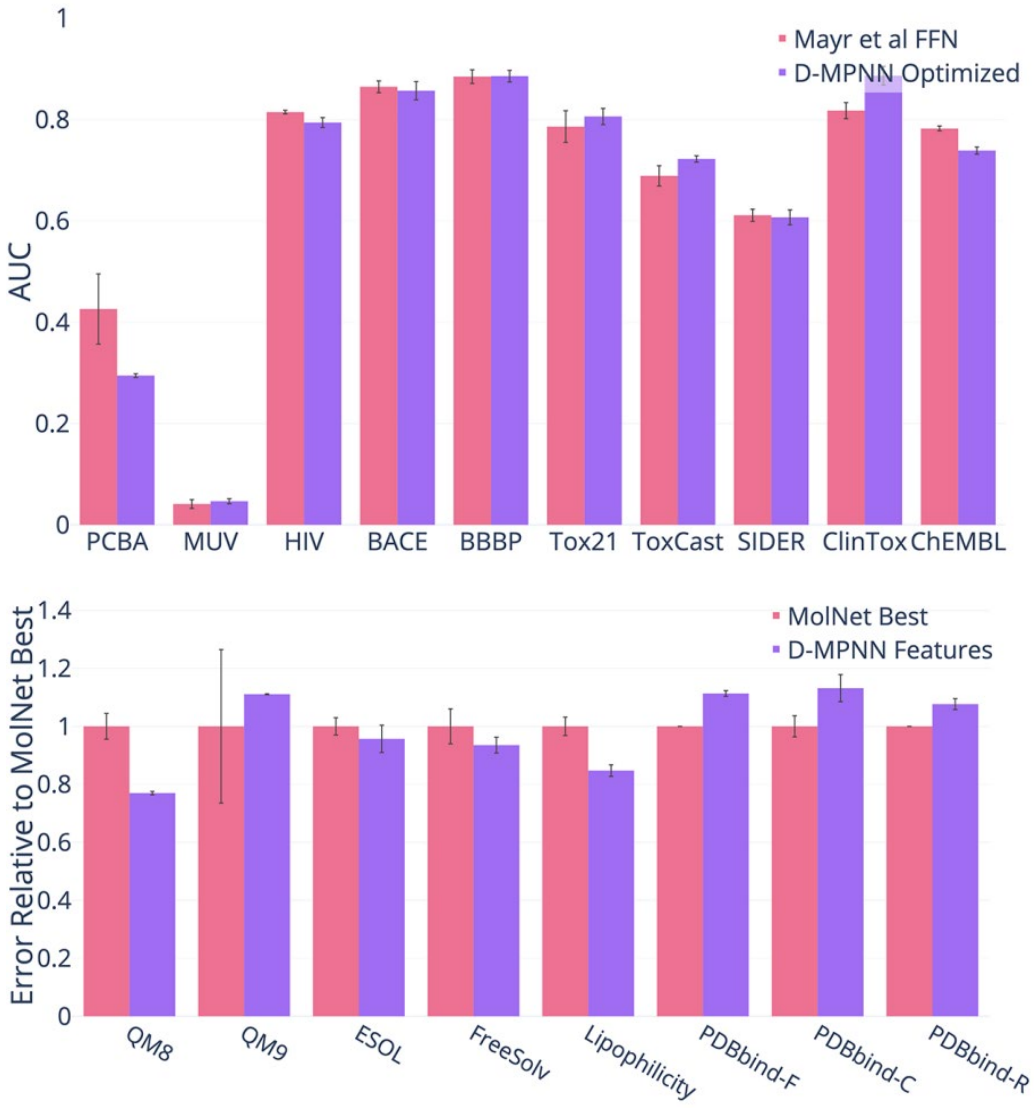
- Data
  - 19 public datasets (including Tox21)
    - 200 < number of molecules N < 450K+]
  - 16 private industrial datasets
  - 80-10-10 train-test-validate splits
  - 20 iterations of Bayesian Optimization on 10 random seeds
- Compared results to several baselines
  - i.e., feed-forward NN, random forest with fingerprint representations, etc.

Outline

- Introduction
- Methods
- ChemProp
- Experiments
- **Results and Discussion**
- Conclusions and Future Work

Results

- Comparison to baselines
  - D-MPNN is comparable to current methods
  - Seems to be very dataset dependent
- Private companies often split data based on time
  - Past = train, future = test/val
  - Results in better performance



Analyzing Modeling Choices

- Message type: **directed bond** vs. atom
  - Showed minimal improvement over atom-based
- RDKit features: incorporating molecule-level features
  - Highly dataset and task dependent
- Hyperparameter optimization
  - Slight improvements on most datasets
- Ensembling: train of 5 models, select best one
- Model struggles with low-quantity target labels

Outline

- Introduction
- Methods
- ChemProp
- Experiments
- Results and Discussion
- **Conclusions and Future Work**

Conclusions and Future Work

- Comparative experiments on 35 public/private datasets
  - D-MPNN often matches or outperforms baselines
  - Strong results on private datasets
- Future work
  - Incorporate 3D molecular information into model
  - Explore pre-trained models
    - Possibly generalize from large to limited/small datasets
  - Adapt to datasets with severe class imbalances
  - Study uncertainty quantification for model generalizability

Ideas f uture R arch

- Read relate
- Practice Chem omplicated datasets
- Explore oth
- Explore ative
  - Instead ecules to drug operties, why not properties o novel, drug-like m ecules???

- Study for prelims (not research)

Acknowledgments

- Financial support provided by IBioSTeP: NIH GRISE at UC Merced (T32GM141862)

Abstract: Model Details

- Initial edge features: $\mathbf{e}_{vw}^{\mathrm{d}} = \mathrm{cat}(\mathbf{x}_v, \mathbf{e}_{vw}))$
- Update function:

$$\mathbf{h}_{vw}^{0} = \tau(\mathbf{W}_i \mathbf{e}_{vw}^{\mathrm{d}})$$

$$\mathbf{h}_{vw}^{t+1} = \tau(\mathbf{h}_{vw}^{0} + \mathbf{W}_h \sum_{k \in \{N(v) \backslash w\}} \mathbf{h}_{kv}^{t})$$

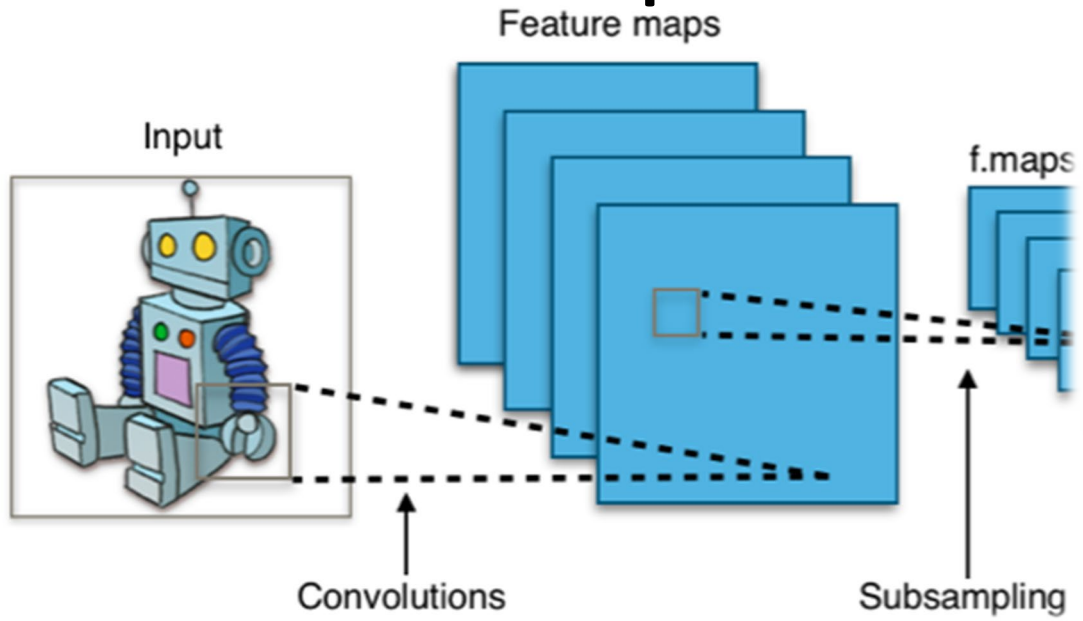
- Aggregate final edge states into atom embeddings: $\mathbf{h}_v = \tau(\mathbf{W}_o \mathbf{q})$ $\mathbf{q} = \mathrm{cat}\left(\mathbf{x}_v, \sum_{w \in N(v)} \mathbf{h}_{wv}^{T}\right)$

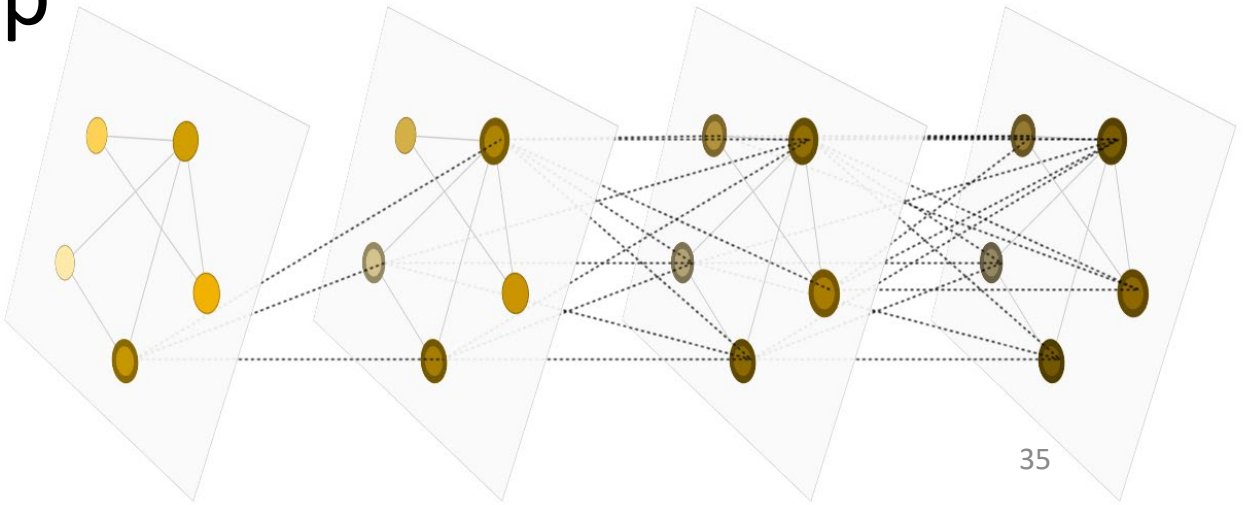
- Final molecular representation: $\mathbf{h}_m = \mathrm{cat}\left(\mathbf{h}'_m, \mathbf{x}_m\right)$

$$\mathbf{h}'_m = \sum_{v \in V} \mathbf{h}_v$$

Abstract: Convolutions on Graphs

- Convolutions provide local context in both CNNs and GNNs
- CNNs: “sliding window” to extract local features
- GNNs: Learn about (k-1)-hop neighborhood after k layers





Abstract: Learning Model Parameters

- Trainable parameters:
  - Weight matrix for aggregated messages W_h
- Feed W_h into any loss function and backpropagate

$$\mathbf{h}_{vw}^{0} = \tau(\mathbf{W}_i \mathbf{e}_{vw}^{\mathrm{d}})$$

Update **Message**

$$\mathbf{h}_{vw}^{t+1} = \tau(\mathbf{h}_{vw}^{0} + \boxed{\mathbf{W}_h} \sum_{k \in \{N(v) \setminus w\}} \mathbf{h}_{kv}^{t})$$

Aggregation