DSSI
DATA SCIENCE
INSTITUTE

Machine Learning Methods to Screen Compounds Targeting COVID-19

Team 4: Marina Dunn, Lance Fletcher,
Moises Santiago Cardenas, Robert Stephany

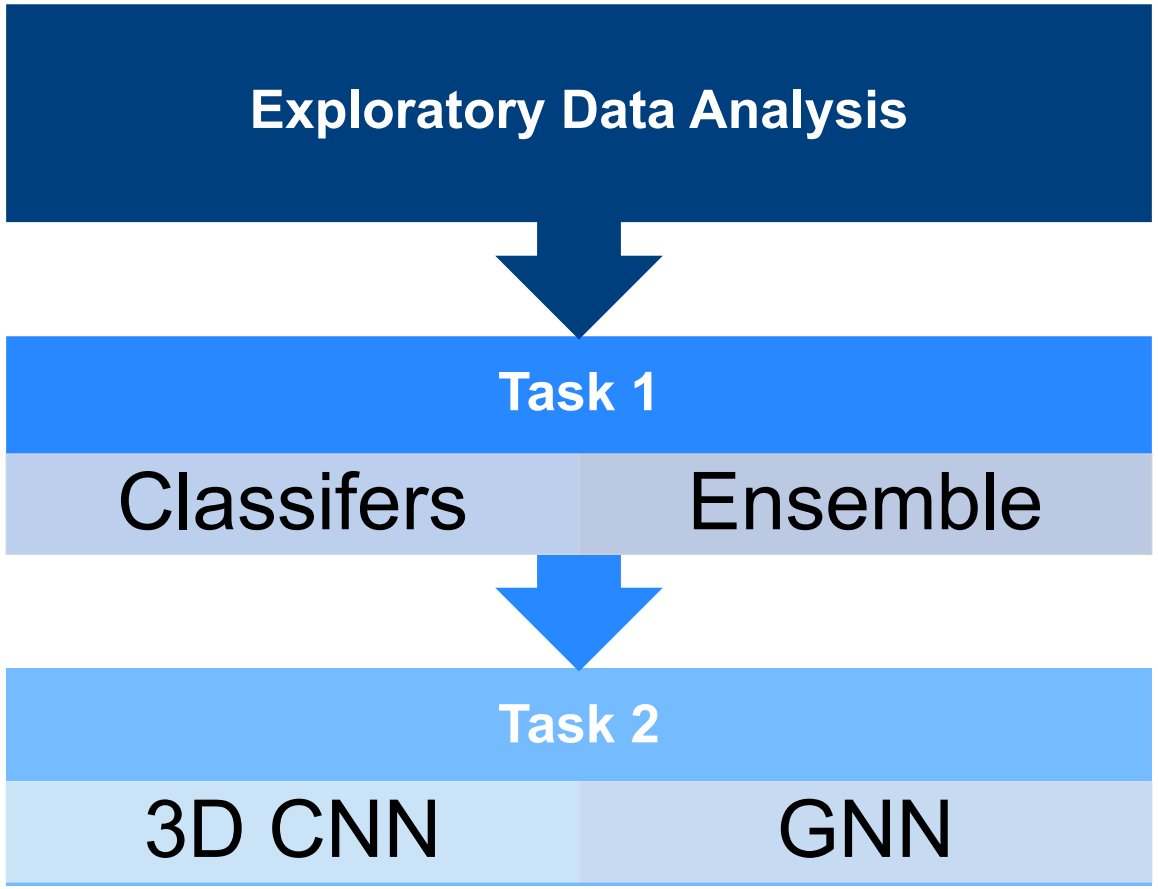
August 9, 2022

DSSI Challenge Problem

LLNL-PRES-XXXXXX
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC
Updated March 2022

Lawrence Livermore National Laboratory

Approach


Exploratory Data Analysis
Task 1
Classifers
Ensemble
Task 2
3D CNN
GNN

DATA SCIENCE INSTITUTE

Task 1

Molecular Descriptors

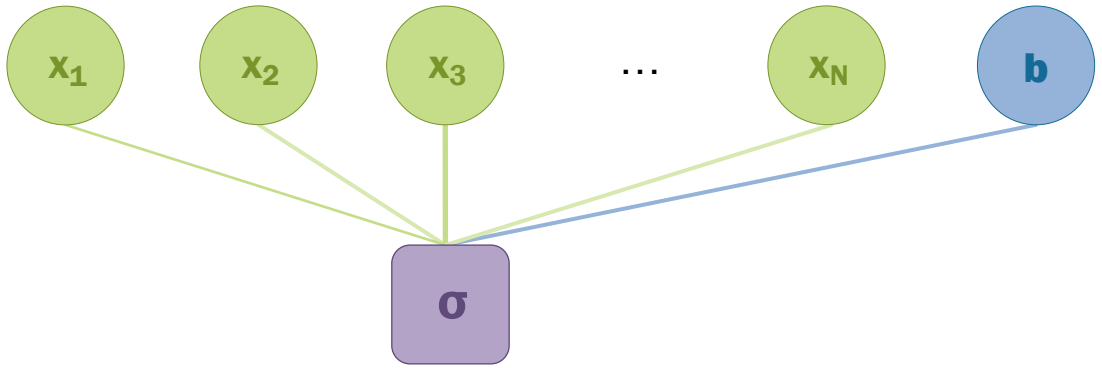
Lawrence Livermore National Laboratory

Exploratory Data Analysis

- **Data Cleaning:**
  - Numerical & Non-numerical statistics
  - Drop columns & rows
  - Sorting, Indexing, Grouping, etc.
  - Duplicates, Missing data, Outliers
    - Fill missing features with mean
  - Scale data

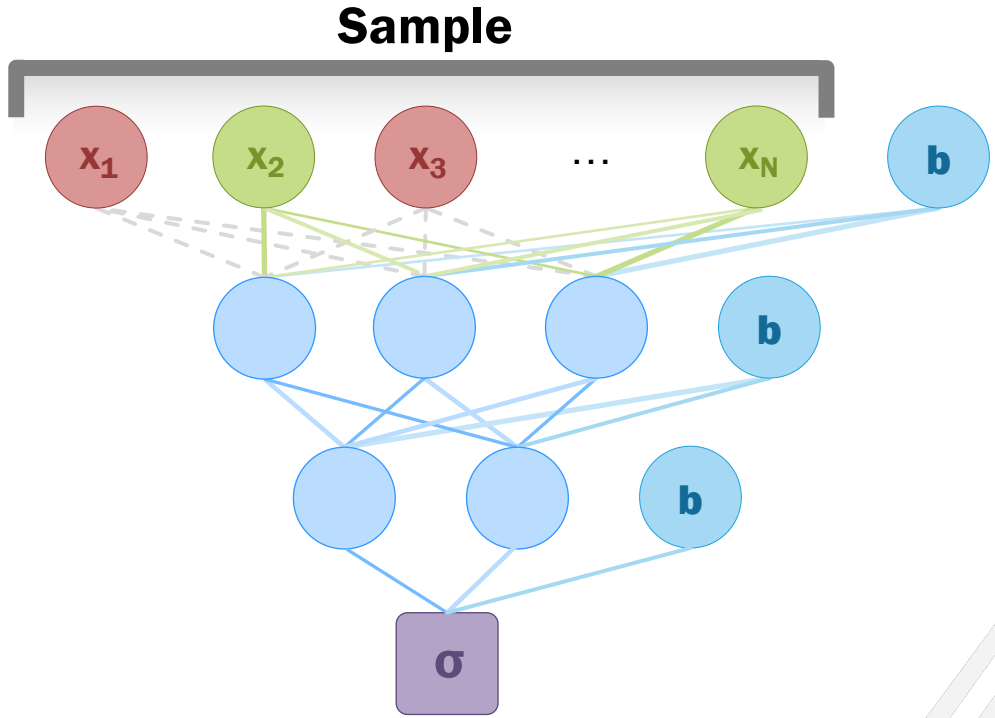
First Attempt: Logistic Regression Classifier

- **Architecture**: Logistic regression classifier
  - $f(x) = \sigma(Wx + b)$
  - Learn $W \in \mathbb{R}^{N\,x\,1}, b \in \mathbb{R}$ using SGD (via Adam optimizer).
- **Results**: Mediocre (≈90%) training set accuracy, poor generalization (≈60% test accuracy)
- This paradoxically suggests that the model is both **too complex** (overfitting) and **too simple** (there is no affine decision boundary)



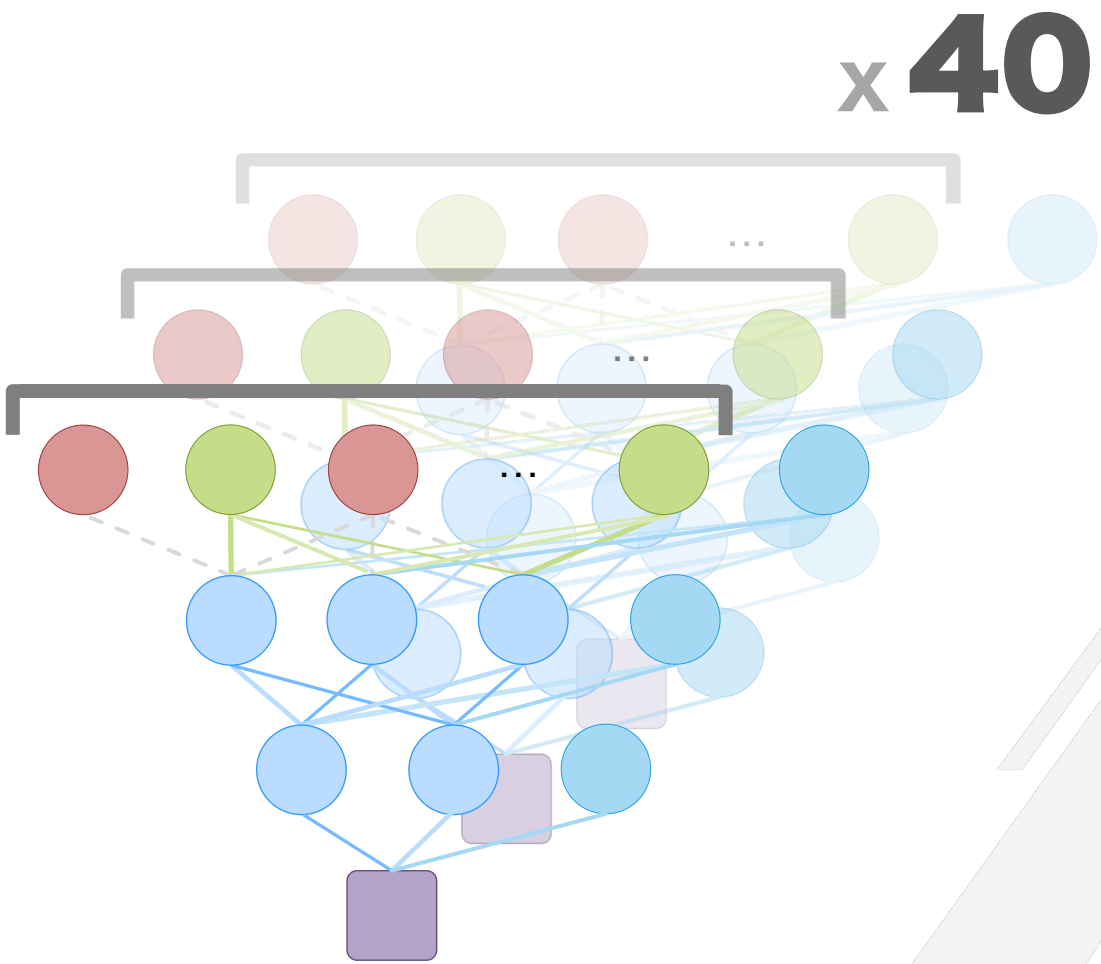
Second Attempt: Feature Subset

- **Architecture**: MLP that accepts a subset of the input features
  - Uses ≈ 100 of 208 features
  - 3 hidden layers (40-20-10 units)
- Can learn a complex decision boundary
- **Results**: Some models generalize well (> 75%), others do not (< 60%)
- This suggest that some combinations features are more useful than others



Third Attempt: Ensemble Approach

- **Architecture**: Ensemble of $N \approx 40$ models from the last slide
  - To evaluate, first select sub-model subset
  - Evaluate sub-models on the input and report the average of their predictions
- After training, rank the models according to their validation set performance
- Evaluate the ensemble when we use the best n sub-models, for $n \in \{1, 2, \ldots, N\}$
  - Select the best n
  - Evaluate ensemble on the test set



Set	Accuracy	Precision	Recall	F1
Train	99.1%	.988	.992	.990
Validation	88.4%	.836	.836	.876
Test	80.5%	.740	.740	.812

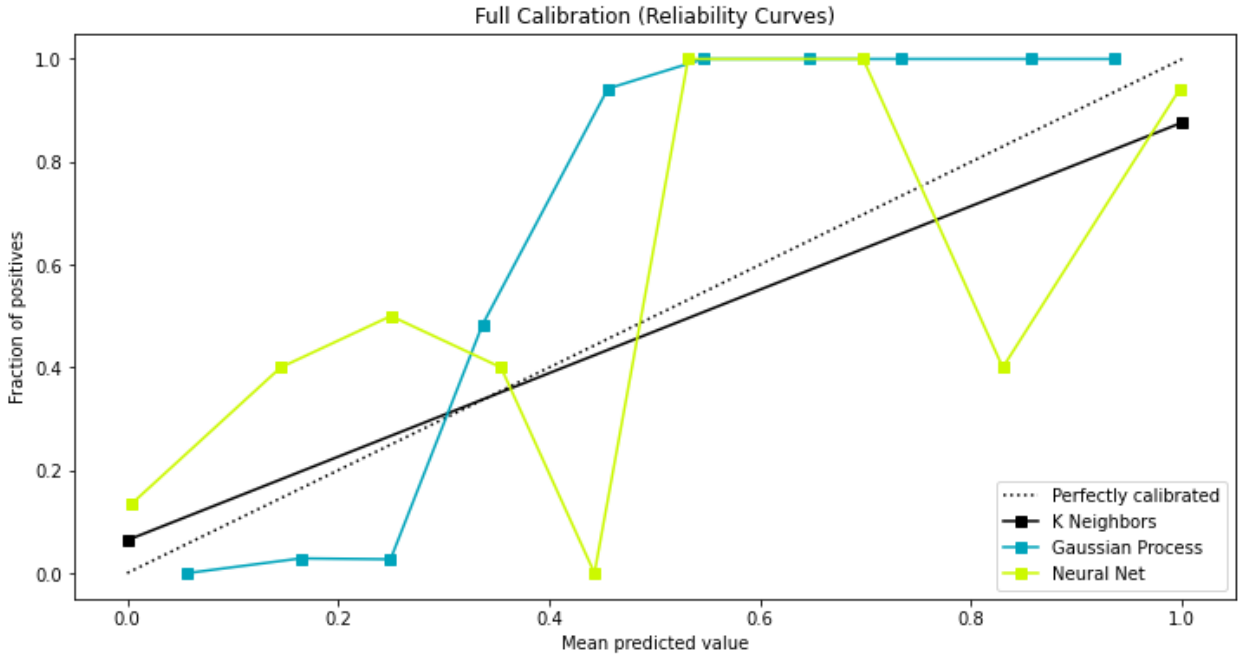
First Attempt Models

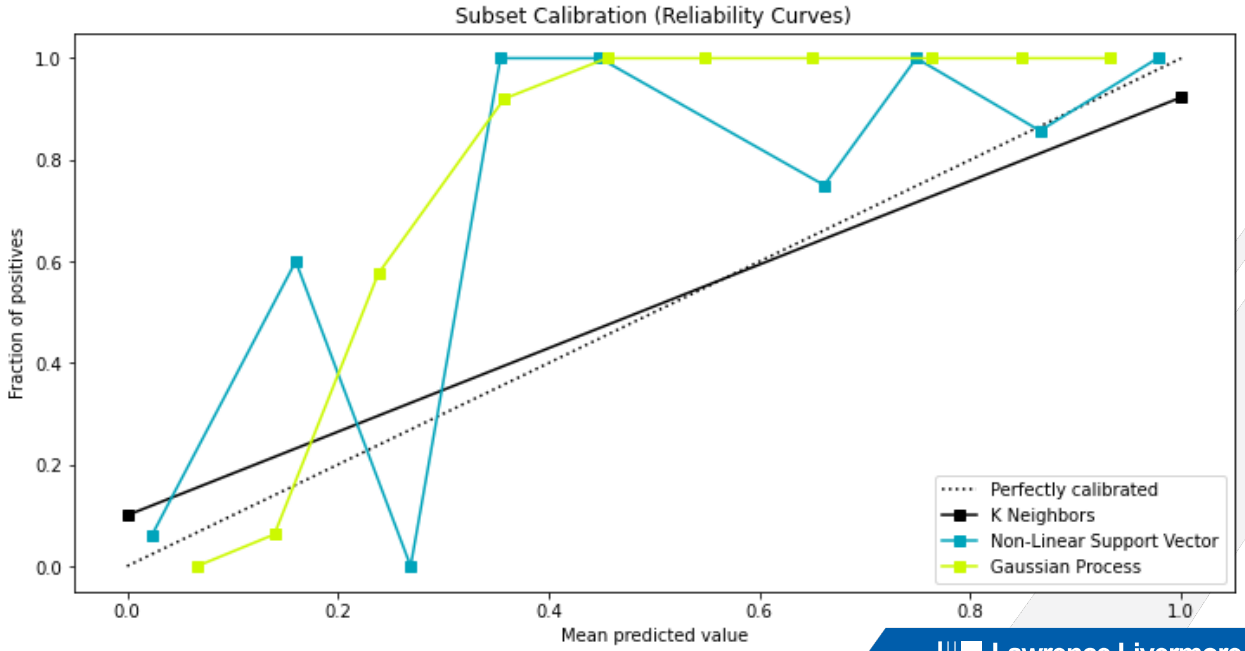
- Changed train-test ratio, used class weights, tried for both cleaned full & smaller datasets
- Used **scikit-learn grid search** for **11** initial models:
  - Linear & Non-linear Support Vector Classifiers
  - K-Nearest Neighbors Classifier
  - Gaussian Process Classifier
  - Decision Tree & Random Forest Classifiers
  - Ada Boost Classifier
  - Bagging Classifier
  - Gaussian Naïve-Bayes
  - Quadratic Discriminant Analysis
  - Neural Network

Classifier Results

Model	Accuracy	Precision (weighted)	Recall (weighted)	F1 (weighted)
K-Nearest Neighbors	89% (full) 94% (small**)	90% (full) 94% (small**)	89% (full) 94% (small**)	89% (full) 94% (small**)
Gaussian Process	88% (full) 90% (small**)	89% (full) 90% (small**)	88% (full) 90% (small**)	88% (full) 90% (small**)
Neural Network	89% (full)	89% (full)	89% (full)	89% (full)

** Refers to smaller data subset with rows 4+ standard deviations away from mean removed; scores less meaningful for this data.





Task 1 Challenges

- Data Cleaning: fewer samples, data imbalance
- Dataset nuances
- Scientific background
- Time Constraints

If We Had More Time….

- Look at additional metrics (ROC AUC, etc.)
- Try fusion of models (i.e. average final layer activations)
- Additional feature analysis

Task 2

3D Ligand Structures

Graph Neural Network Convolutions

- Let $G = (V, E)$ be a graph
  - Assume for each vertex $v \in V$, there is a feature vector h_v^0
- **Goal**: map feature vectors, $\{ h_v^0 : v \in V \}$ to a graph label.
- Modern architectures generally use **graph convolution** approach.
  - Update each node's feature vector based on its neighbors'.
    - Do this $L \in \mathbb{N}$ times.
  - Combine or "pool" final feature vectors.
  - Pooled vector goes through fully-connected network to make a final prediction.
- Graph Convolutional Networks (GCN) – *Kipf 2016*

$$h_v^{l+1} = \sum_{u \in N(v)} c_{u,v} W h_u^l$$

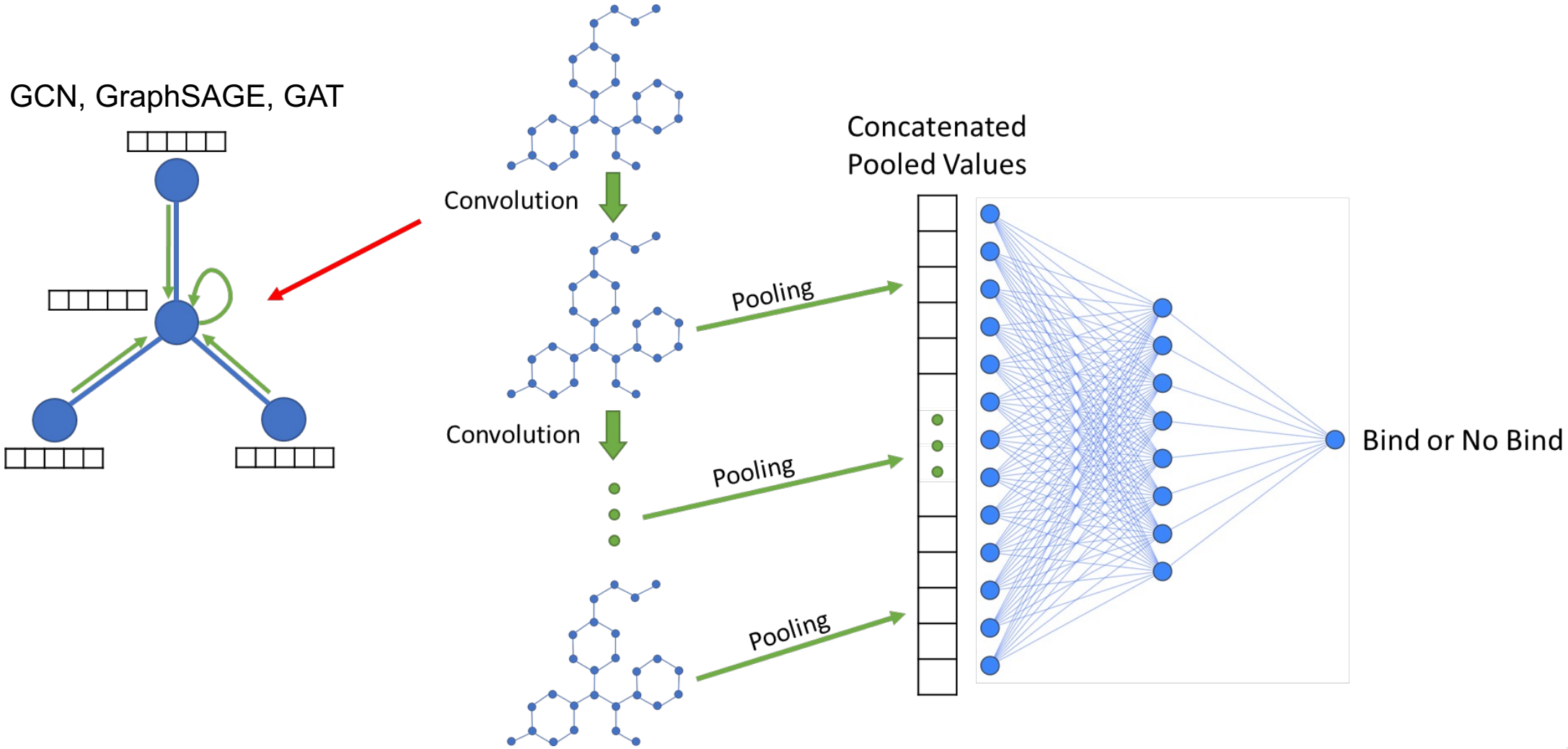
- Graph Sampling and Aggregation (GraphSAGE) – *Hamilton 2017*

$$h_v^{l+1} = W_1 h_v^l + \sum_{u \in N(v)} W_2 h_u^l$$

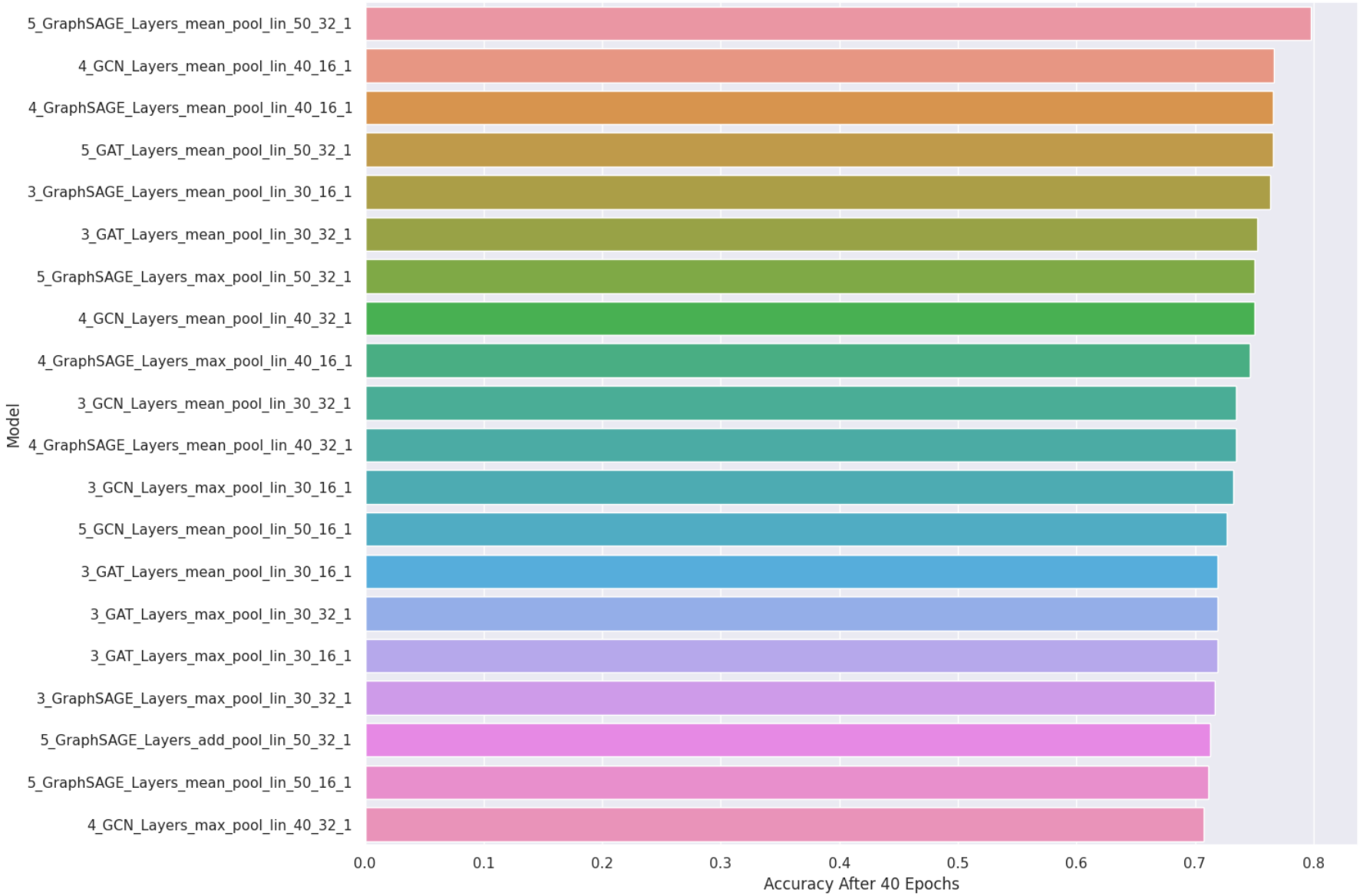
- Graph Attention Networks (GAT) – *Velickovic 2017*

$$h_v^{l+1} = \sum_{u \in N(V)} w_{u,v} W h_u^l$$

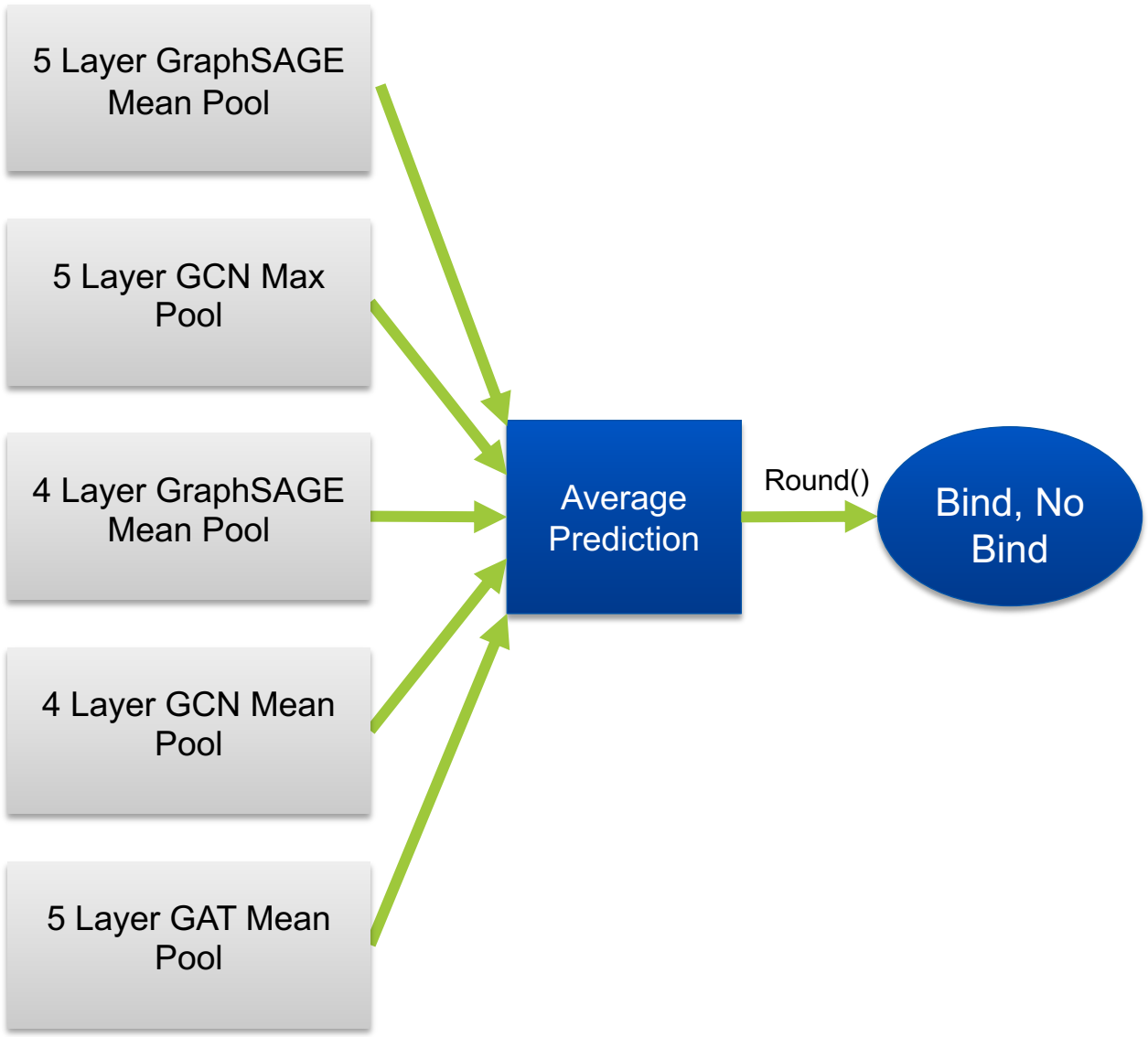
Graph Neural Network Architecture

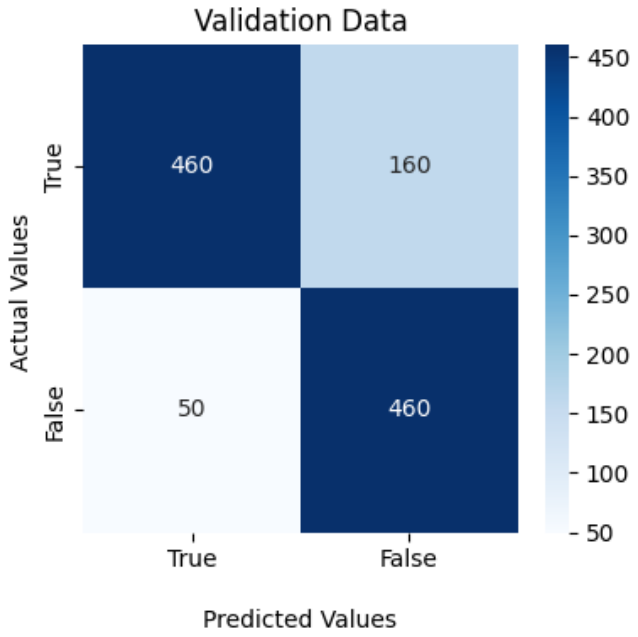


GNN – Hyperparameter Search

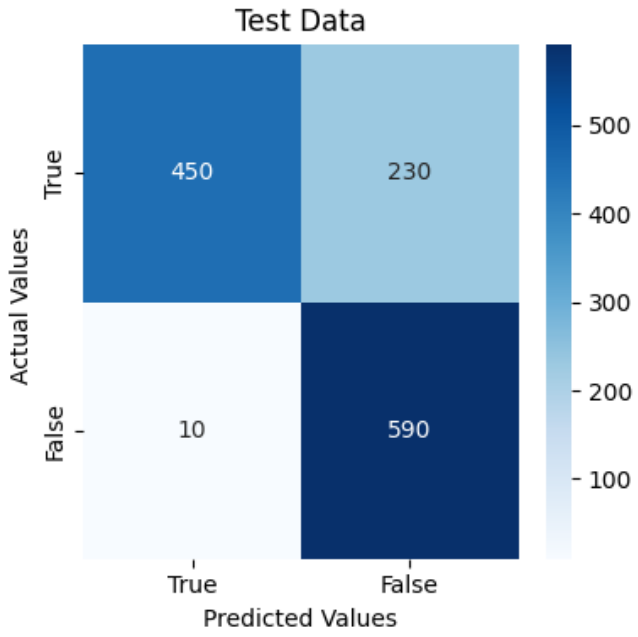


Ensemble GNN Results





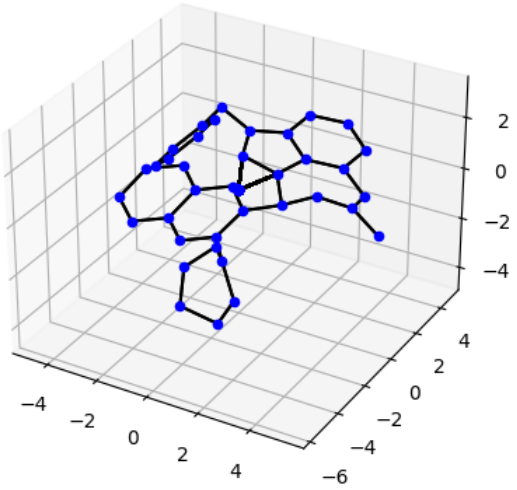
Accuracy: 81.42%



Accuracy: 81.25%

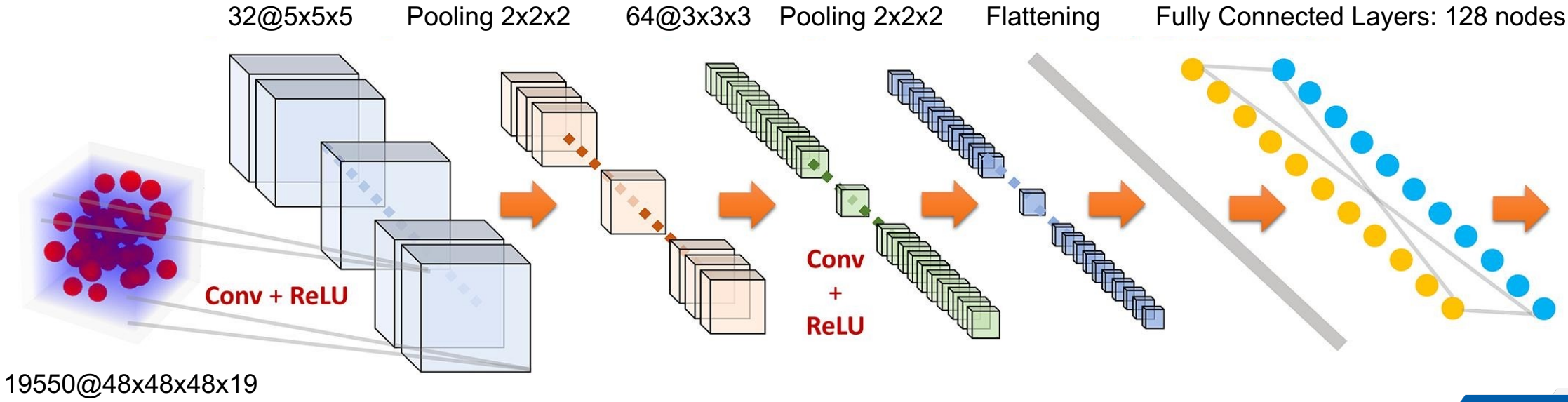
If We Had More Time….

- Explore & test with other kinds of pooling
    - Our hyperparameter search suggests that pooling type is the most important variable
- Weight “binding” examples more than “non-binding”
- Try other ensemble methods (Bagging, Stacking, Boosting)
- Investigate mixing different convolution layers in a single architecture
- Explore different methods of translating ligand data to a graph



3D Convolutional Neural Networks

- Used **TensorFlow** to build models similar to Stevenson et al. (2021):
  - Adam Optimizer
  - Binary Cross-entropy Loss Function
  - Use Early-Stopping
  - Measure Accuracy, Precision, Recall, F1
  - Use class weights



3D CNN Challenges

- Data Extraction & Restructuring
- SLURM Allocation & LC System Issues
- Time Constraints

If We Had More Time….

- Test more architectures and implement fusion model
- Try Binary Focal Cross-entropy Loss
- Explore using uncertainty quantification
- Do comprehensive performance and behavior analysis

Thank You!

- Jen Bellig
- Nisha Mulakken
- Goran Konjevod
- Hyojin Kim
- Garret Stevenson
- Derek Jones
- DSSI Course Lecturers

Public Github repo:

SCAN ME