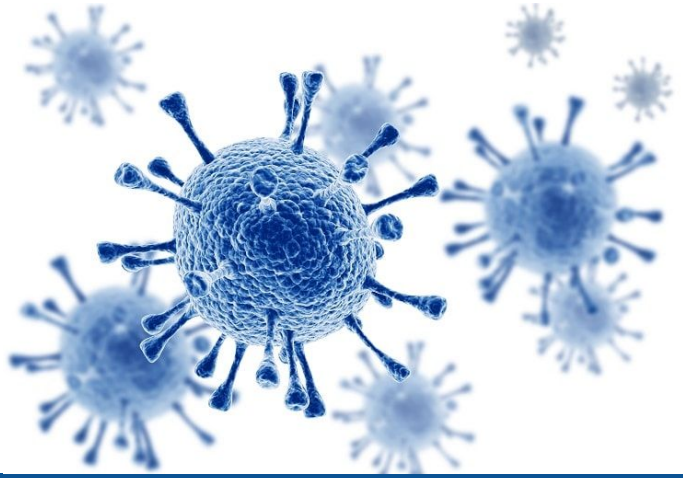
Berkeley
UNIVERSITY OF CALIFORNIA



Prediction of Molecule's Ability on HIV Replication Inhibition

BioE C242 – Machine Learning for Molecular Problems

30 Apr 2024 Adrien Bourgain MEng in Bioengineering

Summary

Context

Definition

HIV (human immunodeficiency virus) = virus that **attacks the body's immune system**, making a person more **vulnerable to other infections and diseases**.

Background

If not treated: lead to **AIDS** = **degenerescence of the immune system**, with falling number of CD4 cells

Antiretroviral therapy (ART): treatment for HIV infection that uses a **combination of antiretroviral drugs** to suppress the virus

The **DTP (Drug Therapeutics Program) AIDS Antiviral Screen** (National Cancer Institute) **=** created the HIV dataset to **discover new treatments**

Objective of the study

→ developing **ML/DL algorithms** to predict the **anti-HIV benefits of drugs**

Reasons:

- **Screening** for new candidates
- Suggestion of **structural modifications** to improve drug efficacy and safety
- **Drug Repurposing**

[1–7]

The HIV Dataset

Composition

41,127 compounds

- **SMILES representation of the molecule**
- Activity: confirmed inactive (CI), confirmed active (CA), confirmed moderately active (CM)
- **HIV_active: binary classification**
  (1: active, 0: inactive)
  → **~3.5%** of active molecules

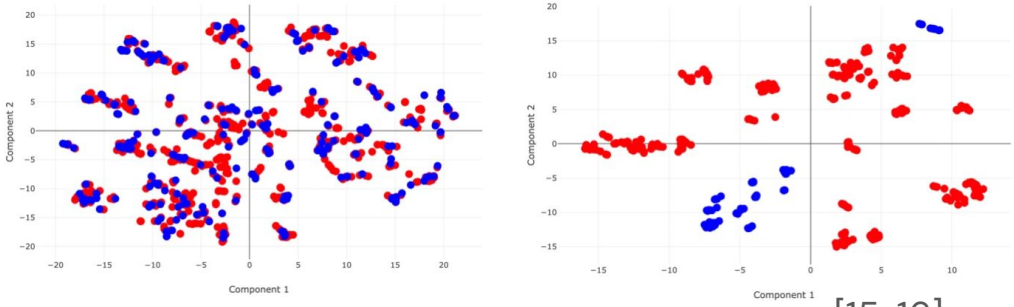
Scaffold Split

→ split into **training**, **validation**, and **test** subsets following an **80/10/10 ratio**, using a **scaffold split**

Scaffold split: based on the **scaffold of the molecules**

→ train/val/test sets are **more structurally different**, making it **more challenging** for the model than a random split.

Benchmarks for HIV are done with scaffold split.

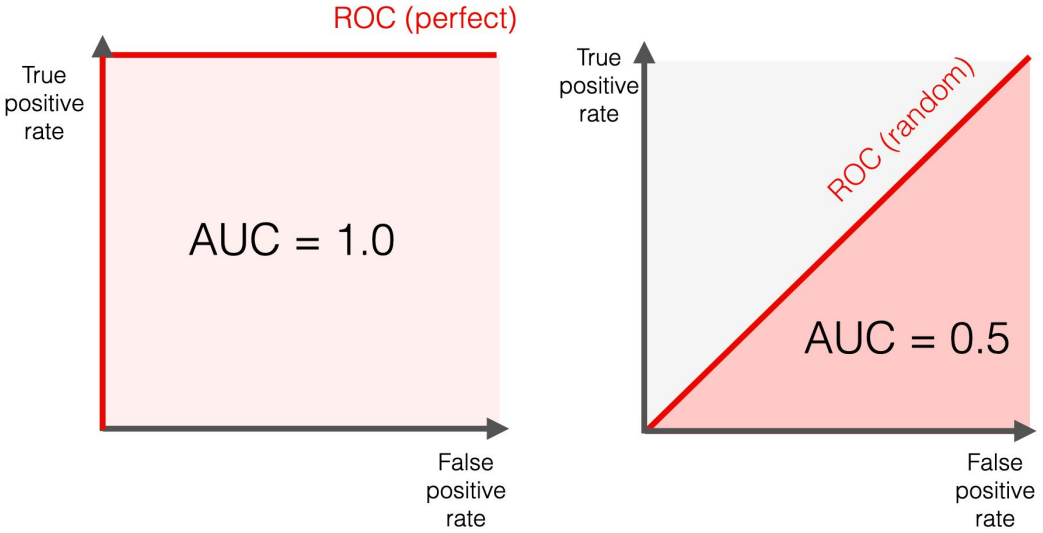


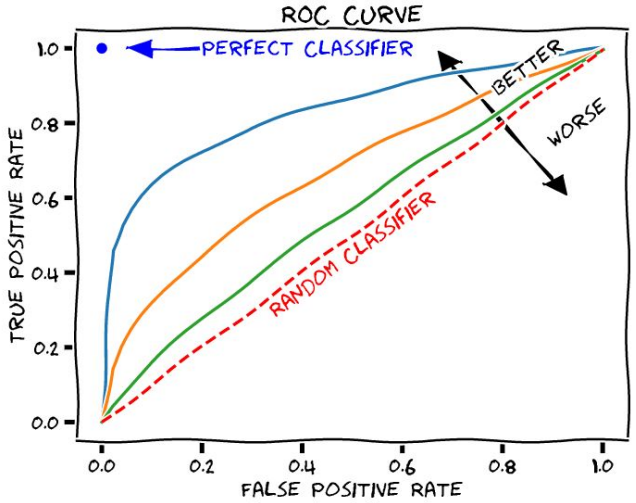
[15, 16]

Metric for the benchmark

ROC AUC

Area under the ROC curve (**ROC curve**: performance of a binary classifier model **at varying threshold**) **values**





[20, 21]

How can we represent our molecules?
How does it impact the prediction capabilities?

The General Descriptors

210 numerical features generated with **RDKit**

Type

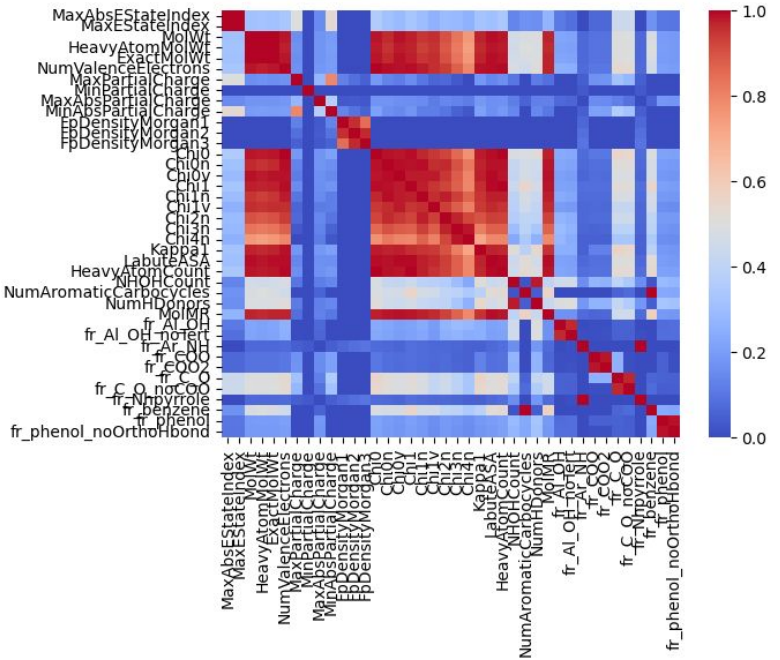
- Constitutional (MolWt, HeavyAtomMolWt, etc.),
- Topological,
- Geometrical,
- Electronic,
- Hydrophobicity and solubility,
- Pharmacophore,
- Molecular fingerprints,
- Functional group counts (e.g., fr_Al_COO, fr_ether, etc).

Missing values?

Difficult to fill by taking means or k-neighbors (at least I don't have the knowledge in Chemistry)
→ chose **algorithms that handle missing values** (Decision Tree, XGBoost)

Dimensionality reduction strategy

→ Removing highly correlated features (>0.95)

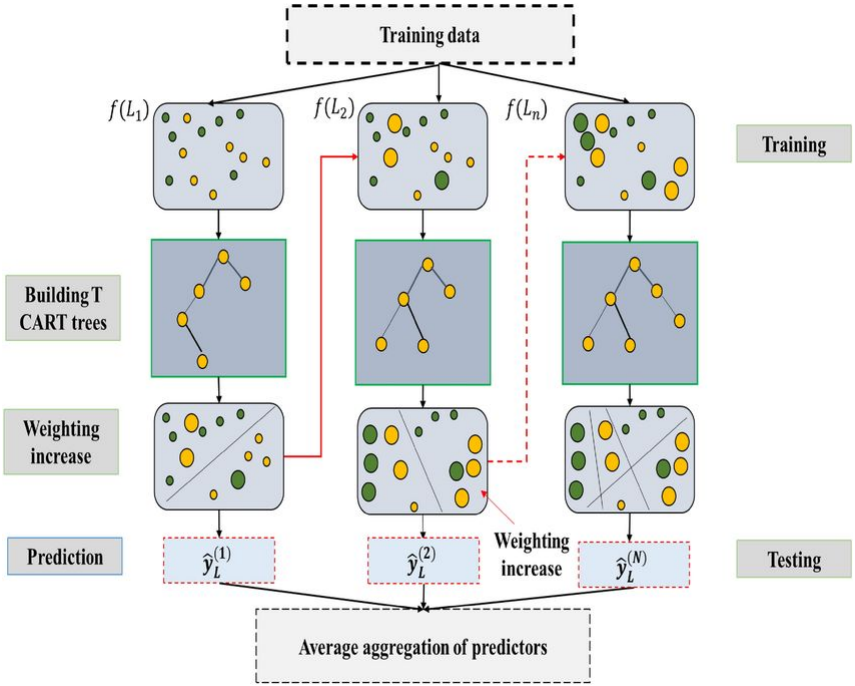


XGBoost classifier

XGBoost

e**X**treme **G**radient **Boost**ing: an **efficient** and **scalable** implementation of **gradient boosting**

- **eXtreme:** enhancements for efficiency, speed (**parallelization**)
- **Gradient:** Refers to the **gradient descent algorithm** used to **minimize the error in predictions** by iteratively improving the model.
- **Boosting:** ensemble technique that builds **multiple models sequentially**, with **each new model correcting errors** made by the previous ones to improve accuracy.



[22]

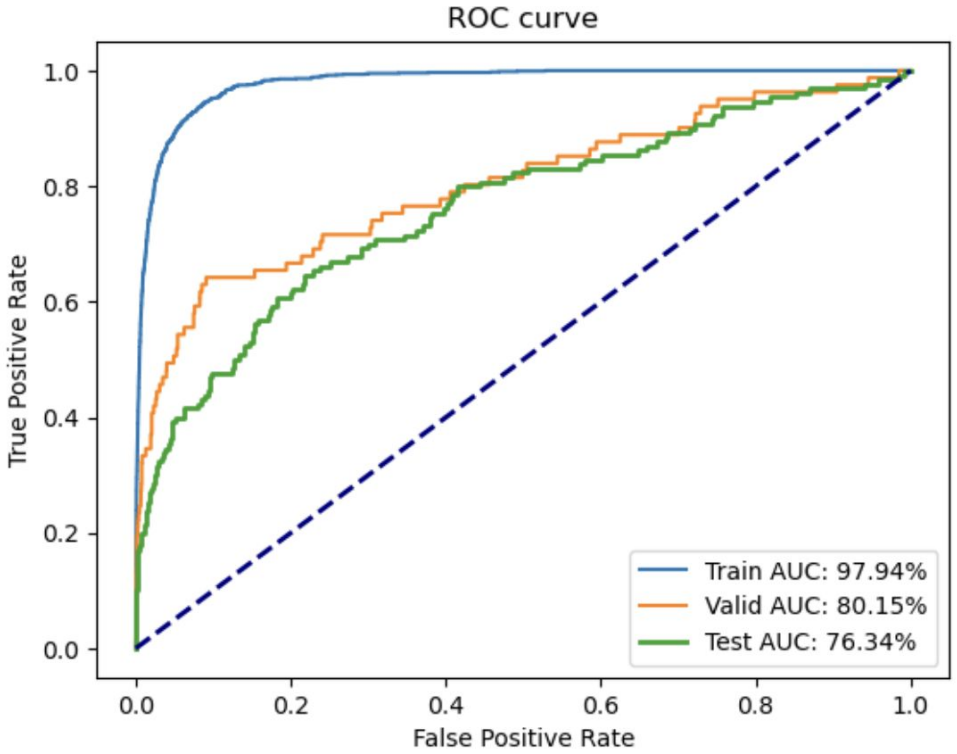
XGBoost classifier

Hyperparameters exploration

Hyperparameter	Description	Space Explored
`learning_rate`	The step size shrinkage used to prevent overfitting.	{0.1, 1e-3}
`n_estimators`	The number of gradient boosted trees.	{25, 40, 50, 60, 100, 500, 1000}
`max_depth`	The maximum depth of the trees.	{3, 5, 7, 9}
`min_child_weight`	Minimum sum of instance weight (hessian) needed in a child.	{1, 3, 5}
`gamma`	Minimum loss reduction required to make a further partition on a leaf node of the tree.	{0.0, 0.1, 0.2}
`subsample`	Subsample ratio of the training instances.	{0.6, 0.7, 0.8, 0.9}
`colsample_bytree`	Subsample ratio of columns when constructing each tree.	{0.6, 0.7, 0.8, 0.9}
`objective`	The learning task and the corresponding learning objective.	'binary:logistic'
`nthread`	Number of parallel threads used to run XGBoost.	4
`scale_pos_weight`	Balancing of positive and negative weights.	{1, 5.07}

XGBoost classifier

Results



Test AUC:

76.34%

The Molecular Fingerprints

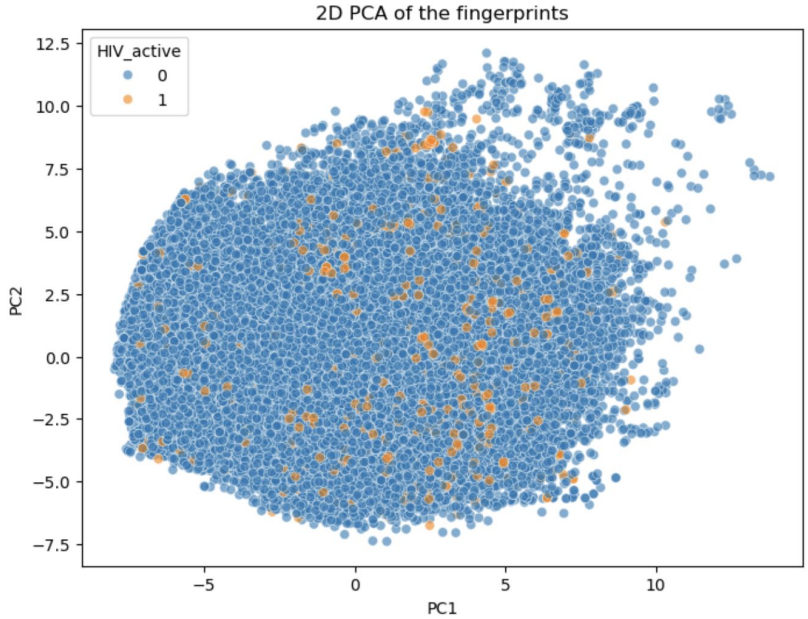
5,287 one-hot encoded features

Extended-connectivity fingerprints

→ Introduced by **Rogers et al.** in **2010**
→ **Topological fingerprints** designed to **capture molecular activity features**

Composed of:

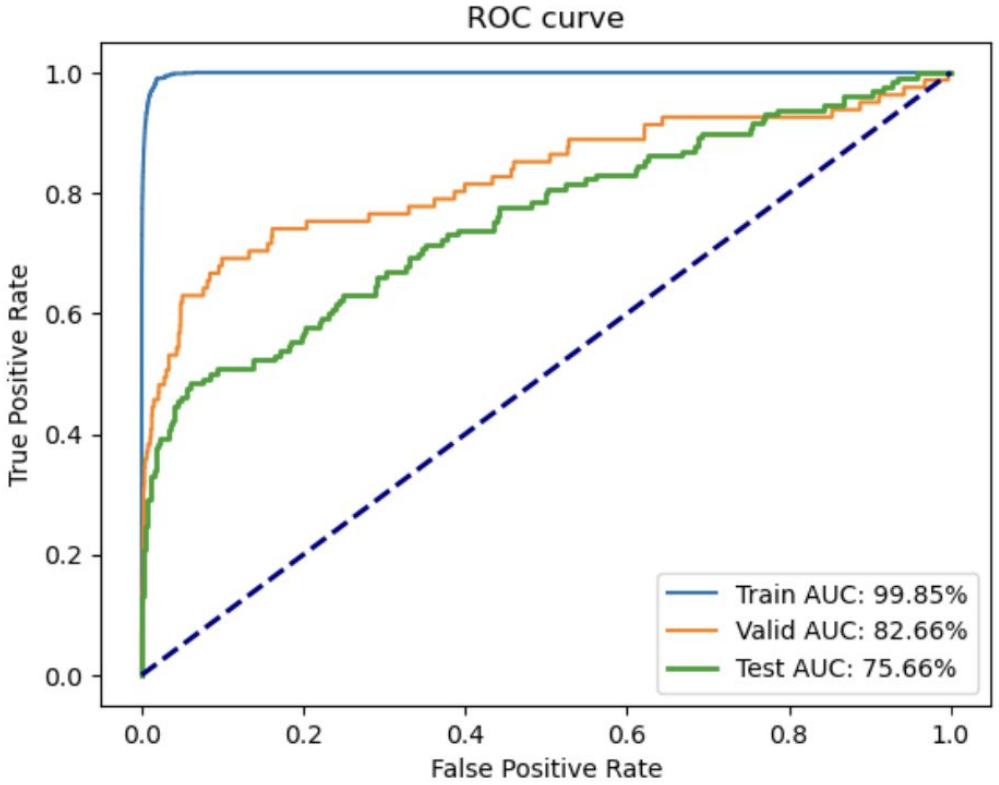
- **Morgan fingerprints:** capture **local molecular structures** with **variable granularity**
- **RDKit fingerprints:** binary vector from paths of atoms up to a certain length
- **MACCS keys:** use a predefined set of 166 keys that represent common molecular substructures. [23]



	HIV_active	1	2	3	...	5285	5286	5287
0	0	0	0	0	...	0	0	0
1	0	0	0	0	...	0	0	0
2	0	0	0	0	...	0	0	1
3	0	0	0	0	...	0	0	0
4	0	0	0	0	...	0	0	0

XGBoost classifier

Results



Test AUC:

75.66%

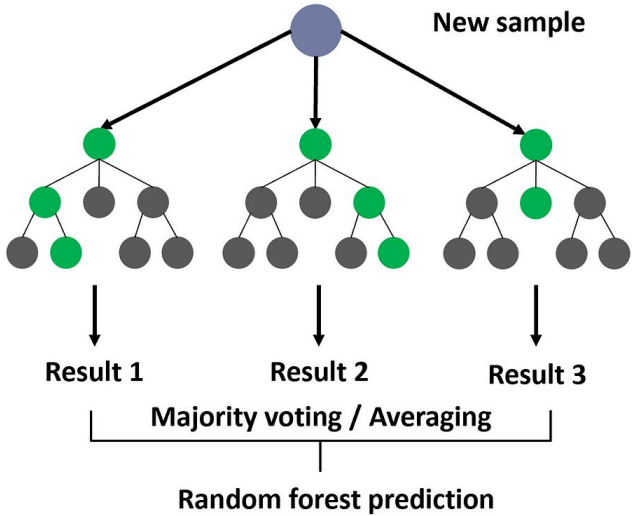
Random Forest classifier

Random Forest

→ **ensemble** machine learning technique
→ multiple decision trees during training and outputs the **majority vote** (for classification) or **mean prediction** (for regression)
→ **averages** multiple deep decision trees, each trained on both **subsets** of **training data** and **features**

Advantage over XGBoost

→ can be **more robust against overfitting**
→ **easier to tune** (fewer hyperparameters)



[25]

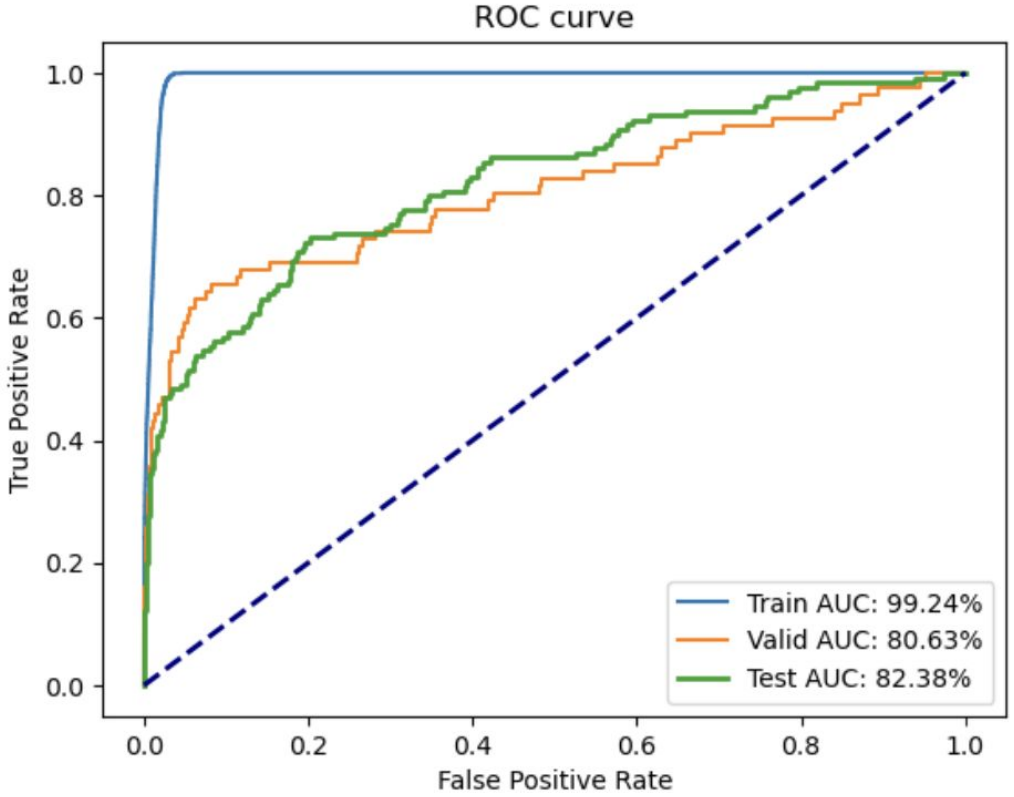
Random Forest classifier

Hyperparameters exploration

Hyperparameter	Description	Space Explored
`n_estimators`	The number of trees in the forest.	500, 1000
`min_samples_leaf`	The minimum number of samples required to be at a leaf node.	2, 5
`min_samples_split`	The minimum number of samples required to split an internal node.	2, 5, 10
`min_impurity_decrease`	A node will be split if this split induces a decrease of the impurity greater than or equal to this value.	0
`min_weight_fraction_leaf`	The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node.	0
`class_weight`	Weights associated with classes in the form `{class_label: weight}` .	`{0: 1, 1: 26.71}`
`warm_start`	When set to `True` , reuse the solution of the previous call to fit and add more estimators to the ensemble.	`True` , `False`

Random Forest classifier

Results



Test AUC:

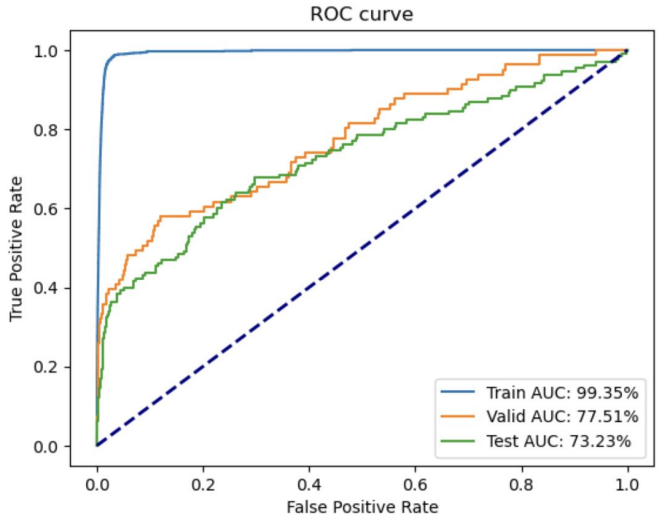
82.38%

MLP

63,157 parameters

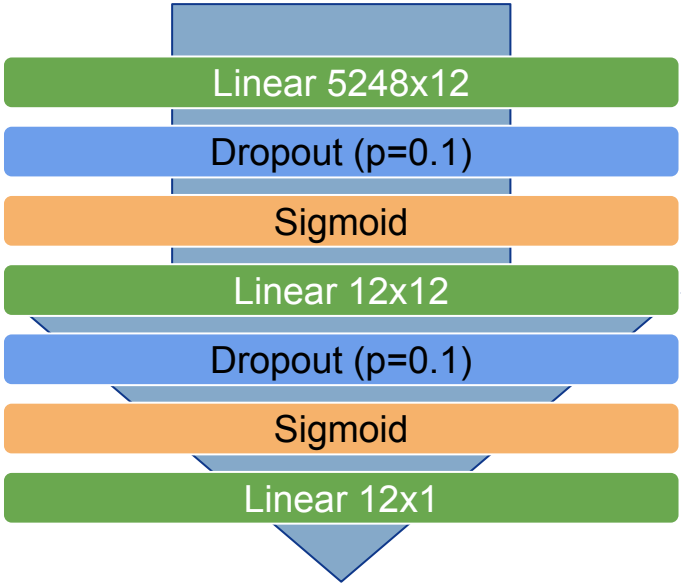
- **Dropout** (p=0/0.1)
- **nn.BCEWithLogitsLoss()** or **Focal Loss**

$$FL(p) = -(y(1-p)^{\gamma}\log p + (1-y)p^{\gamma}\log(1-p))$$



Test AUC:

73.23%



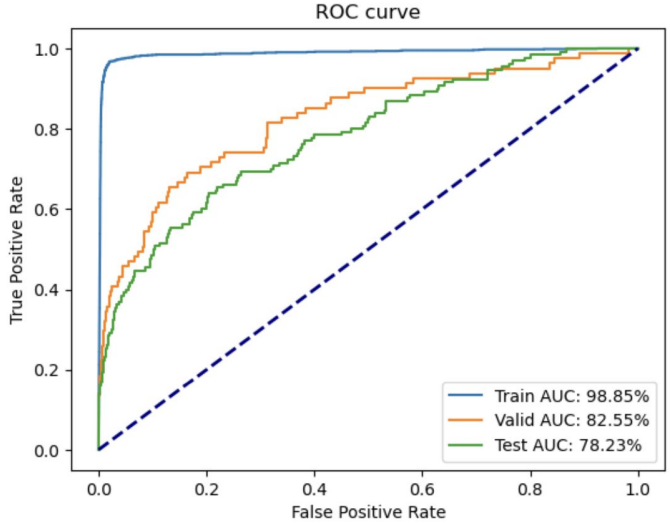
[24]

MLP – PCA

99% explained variance

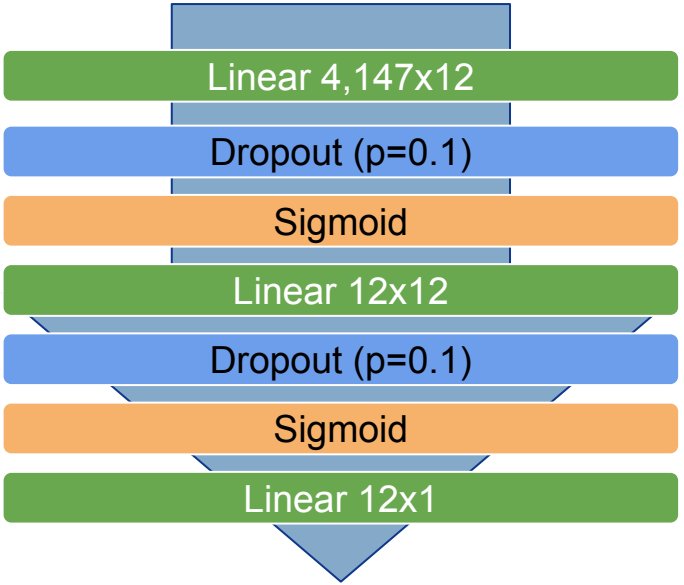
→ **4,147** components

Hyperparameter	Description	Space Explored
`dropout`	The dropout rate for regularization during training.	{0, 0.1, 0.5}
`l2` (weight decay)	The L2 regularization factor.	{0, 1e-3, 1e-5}
`gamma`	Regularization term on the aggregation step (not standard in GCN, might be a custom addition).	{0, 0.5, 1, 2}
`imbalance_factor`	Factor to adjust for class imbalance in the dataset.	{0, 0.5, 1, 10}



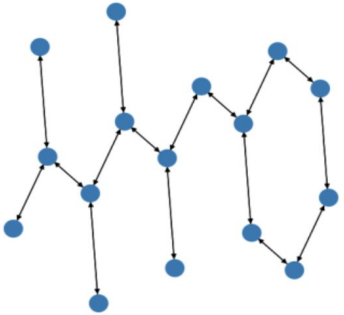
Test AUC:

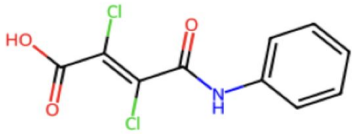
78.23%



Graph Neural Networks

Molecule O=C(O)C(Cl)=C(Cl)C(=O)Nc1ccccc1 and its graph





9 atoms (nodes) features

- Atomic number,
- Chirality,
- Degree,
- Formal electric charge,
- Number of hydrogen atoms connected,
- Number of radical electrons,
- Hybridization state,
- Part of ring,
- Aromaticity.

3 bonds (edges) features

- Bond multiplicity,
- Stereoisomers info,
- Conjugation.

Imbalance dataset

~3.5% active molecules

→ **nn.BCEWithLogitsLoss()** or **Focal Loss**

$$FL(p) = -(y(1-p)^{\gamma}\log p + (1-y)p^{\gamma}\log(1-p))$$

Configuration

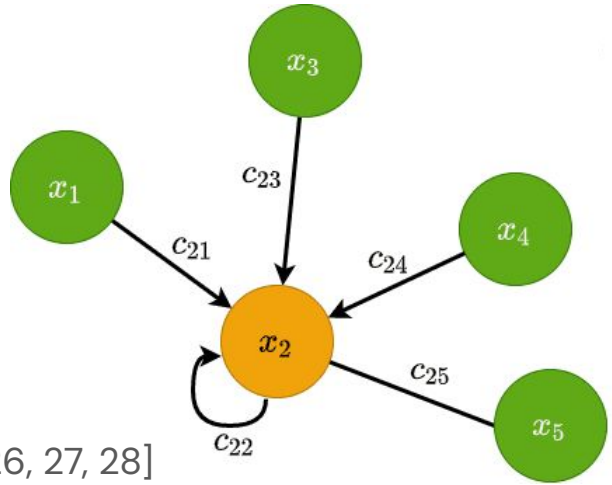
- Adam Optimizer
- ReLU activation
- Residual Connections: “True” or “False”
- Normalization: “batch” or “layer”
- Dropout: p=0/0.1/0.15

GCN 1: GNN with Convolutional Message Passing

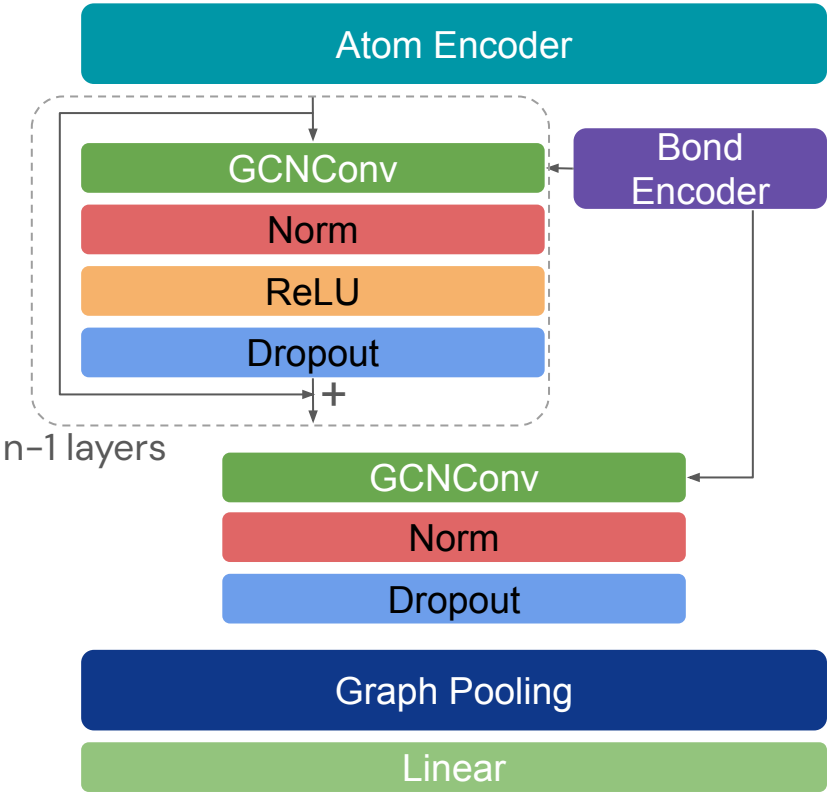
GCNConv

$$\mathbf{X}' = \hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}\mathbf{X}\mathbf{\Theta}$$: nodes matrix

$$\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$$: adjacency matrix
(// with weights here)



[26, 27, 28]



GCN 1: GNN with Convolutional Message Passing

Hyperparameters exploration

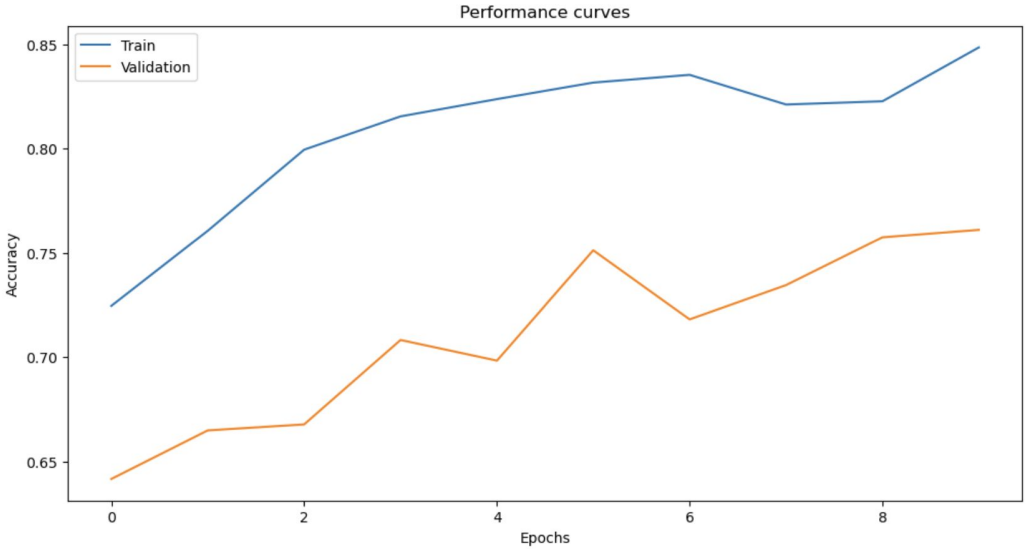
Hyperparameter	Description	Space Explored
`pooling`	The graph pooling strategy to generate a graph-level representation from node representations.	"sum", "mean", "max"
`norm`	The type of normalization layer used within the GCN.	"batch", "layer"
`hidden_dim`	The dimensionality of the hidden layers in the GCN.	{10, 25, 100}
`num_layers`	The number of GCNConv layers in the model.	{2, 3, 4, 5, 10}
`dropout`	The dropout rate for regularization during training.	{0, 0.15, 0.5}
`l2` (weight decay)	The L2 regularization factor.	{0, 1e-3, 1e-5}
`res` (residual)	Whether to use residual connections between layers.	`True`, `False`
`gamma`	Regularization term on the aggregation step (not standard in GCN, might be a custom addition).	{0, 0.5, 1, 2}
`imbalance_factor`	Factor to adjust for class imbalance in the dataset.	{0, 0.5, 1, 10}

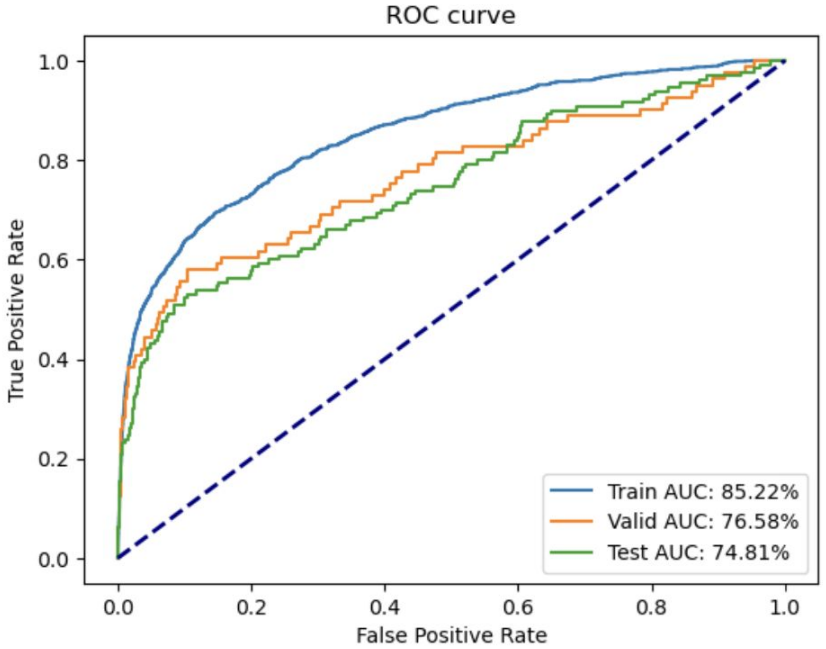
GCN 1: GNN with Convolutional Message Passing

Results

Test AUC:

74.81%



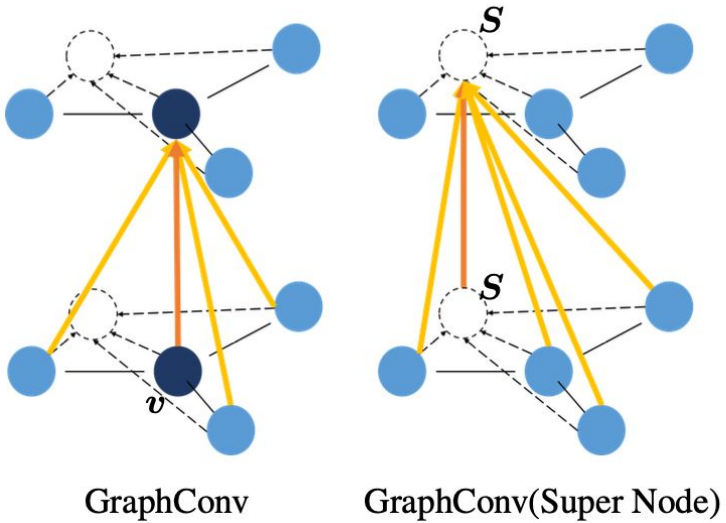


GCN 2: GCN + Super Node

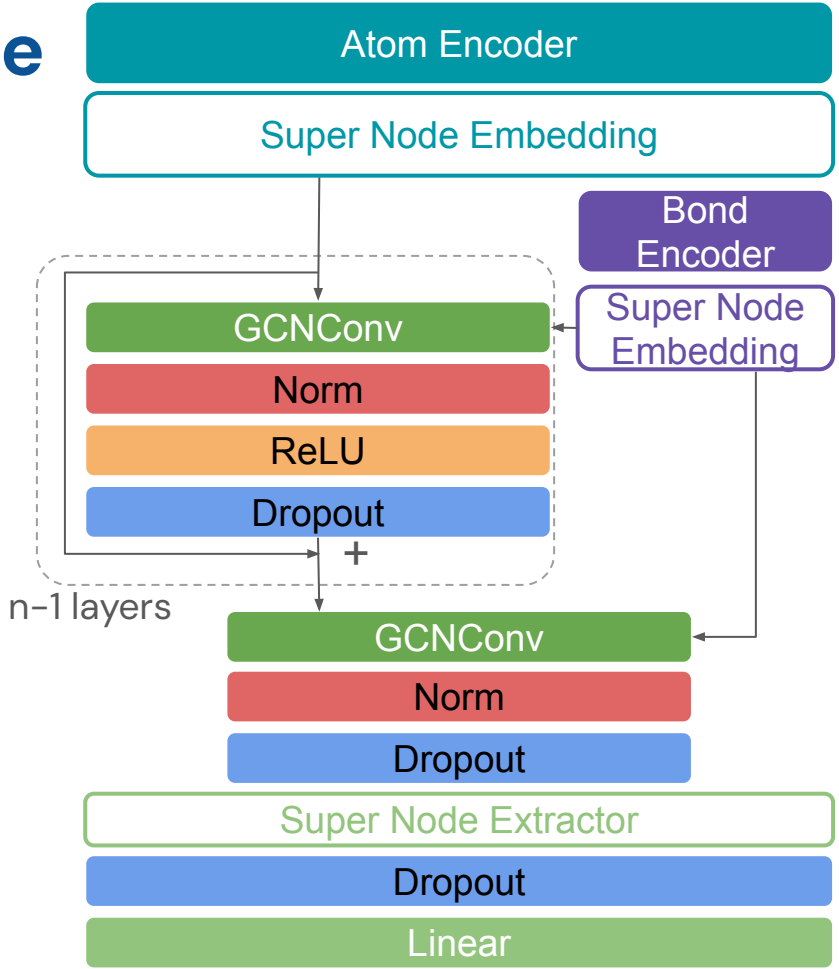
Super node

→ **connected with all nodes** in the graph by a **directed edge**

→ learn **graph-level features**



[29]



GCN 2: GCN + Super Node

Hyperparameters exploration

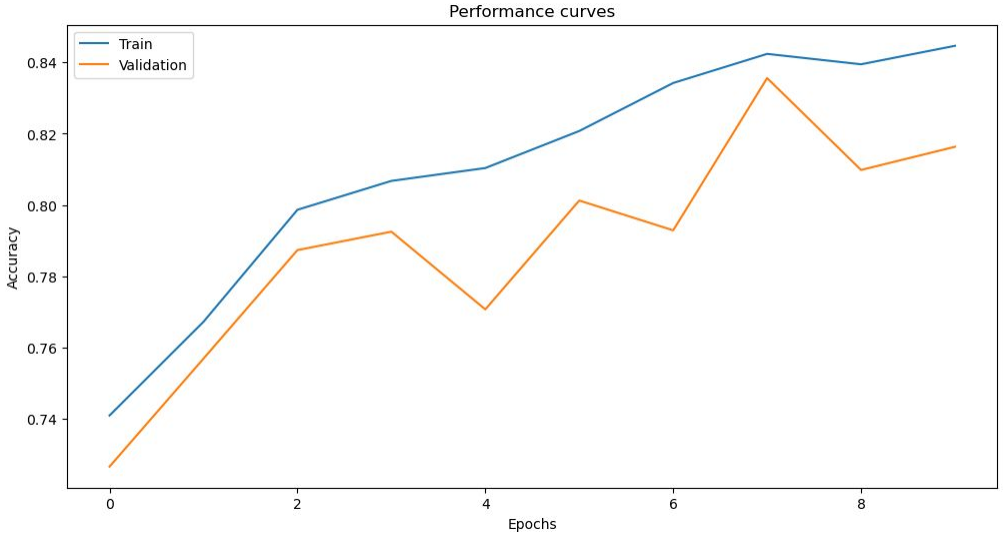
Hyperparameter	Description	Space Explored
`norm`	The type of normalization layer used within the GCN.	"batch", "layer"
`hidden_dim`	The dimensionality of the hidden layers in the GCN.	{10, 25, 100}
`num_layers`	The number of GCNConv layers in the model.	{2, 3, 4, 5, 10}
`dropout`	The dropout rate for regularization during training.	{0, 0.15, 0.5}
`l2` (weight decay)	The L2 regularization factor.	{0, 1e-3, 1e-5}
`res` (residual)	Whether to use residual connections between layers.	`True` , `False`
`gamma`	Regularization term on the aggregation step (not standard in GCN, might be a custom addition).	{0, 0.5, 1, 2}
`imbalance_factor`	Factor to adjust for class imbalance in the dataset.	{0, 0.5, 1, 10}

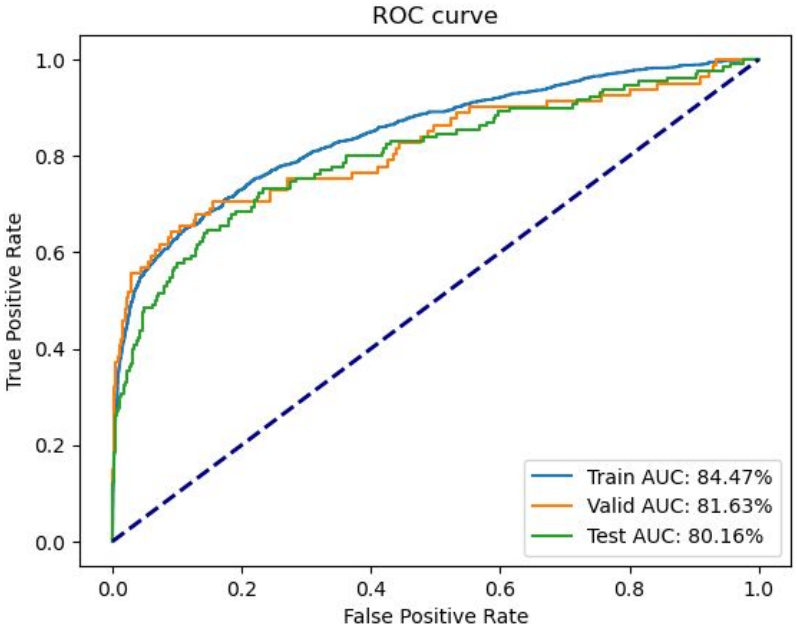
GCN 2: GCN + Super Node

Results

Test AUC:

80.16%



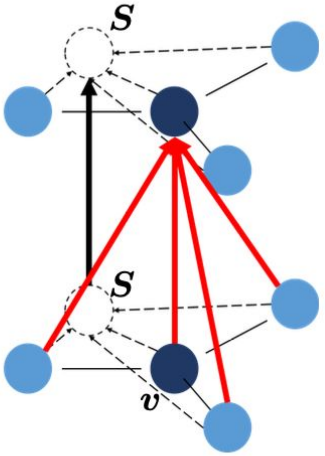


GCN 3: GCN + Super Node + Graph Pooling

Graph Pooling

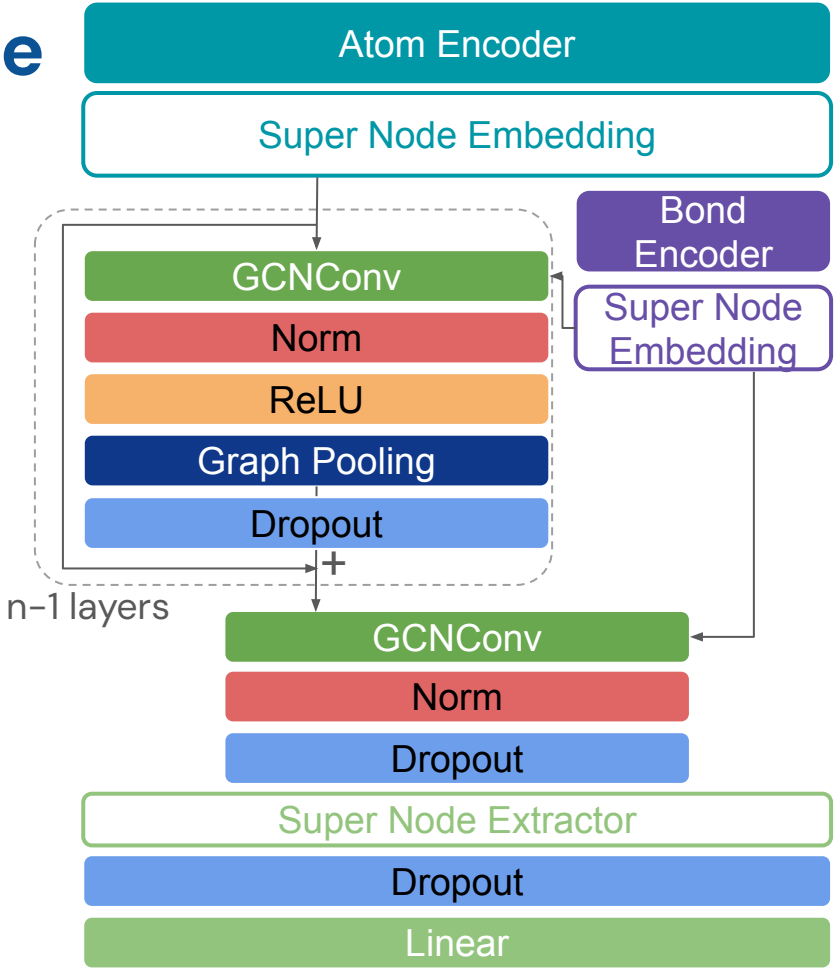
→ returns the **maximum activation** across the node and its neighbours

→ **enlarge the receptive field** without adding extra weights



GraphPool

[29]



GCN 3: GCN + Super Node + Graph Pooling

Hyperparameters exploration

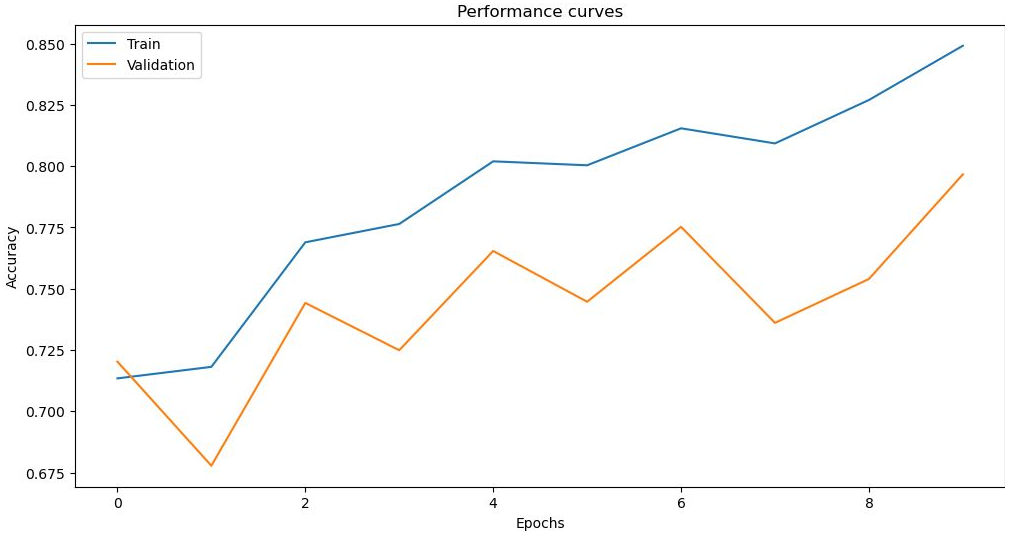
Hyperparameter	Description	Space Explored
`norm`	The type of normalization layer used within the GCN.	"batch", "layer"
`hidden_dim`	The dimensionality of the hidden layers in the GCN.	{10, 25, 100}
`num_layers`	The number of GCNConv layers in the model.	{2, 3, 4, 5, 10}
`dropout`	The dropout rate for regularization during training.	{0, 0.15, 0.5}
`l2` (weight decay)	The L2 regularization factor.	{0, 1e-3, 1e-5}
`res` (residual)	Whether to use residual connections between layers.	`True`, `False`
`gamma`	Regularization term on the aggregation step (not standard in GCN, might be a custom addition).	{0, 0.5, 1, 2}
`imbalance_factor`	Factor to adjust for class imbalance in the dataset.	{0, 0.5, 1, 10}

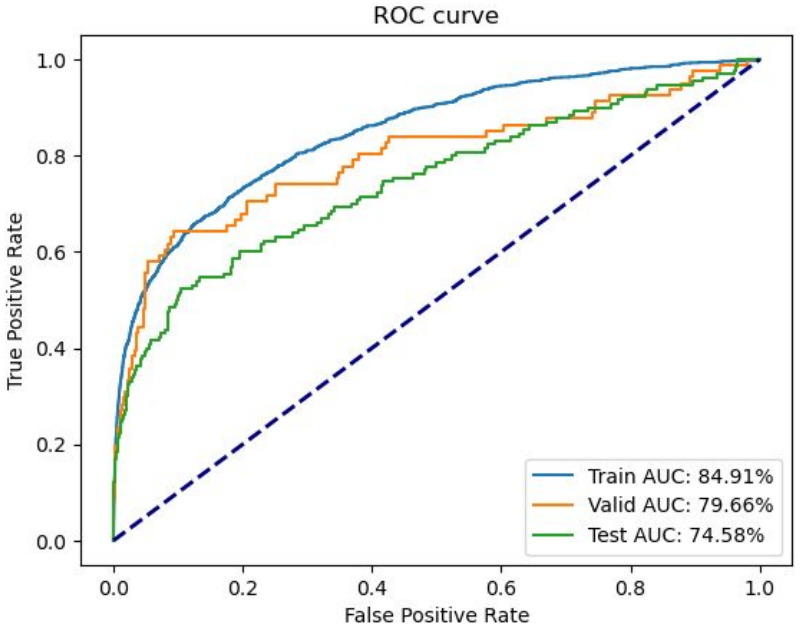
GCN 3: GCN + Super Node + Graph Pooling

Results

Test AUC:

74.58%





Conclusion

Features Type	Model	AUC Training	AUC Validation	AUC Test
General Descriptors	XGBoost	97.94%	80.15%	76.34%
Molecular Fingerprints	XGBoost	98.85%	82.66%	75.66%
Molecular Fingerprints	**Random Forest**	**99.24%**	**80.63%**	**82.38%**
Molecular Fingerprints	MLPPCA	98.85%	82.55%	78.23%
Graph	GCN	85.22%	76.58%	74.81%
Graph	**GCN + Super Node**	**84.47%**	**81.63%**	**80.16%**
Graph	GCN + Super Node + Pooling	84.91%	79.66%%	74.58%

Random Forest classifier on the fingerprints: **best results** on the test set but **overfitting** on the training set

GCN + Super Node: **best trade-off** between overfitting and underfitting

Future Work

→ GNNs implemented by X. Zhang et al., X. Wang et al. and Y. Wang et al. seem to **improve the Test AUC to >84%**

References

[1] “About HIV/AIDS,” 2024. https://www.cdc.gov/hiv/basics/whatishiv.html#:~:text=HIV%20(human%20immunodeficiency%20virus)%20is,care%2C%20HIV%20can%20be%20controlled (accessed Apr. 25, 2024).

[2] “What Are HIV and AIDS?,” *HIV.gov*, 2022. https://www.hiv.gov/hiv-basics/overview/about-hiv-and-aids/what-are-hiv-and-aids (accessed Apr. 25, 2024).

[3] “HIV Treatment: The Basics | NIH.” *2021. Nih.gov*. 2021. https://hivinfo.nih.gov/understanding-hiv/fact-sheets/hiv-treatment-basics.

[4] J. Martinez-Picado and S. G. Deeks, “Persistent HIV-1 replication during antiretroviral therapy,” *Current opinion in HIV and AIDS*, vol. 11, no. 4, pp. 417–423, Jul. 2016, doi: https://doi.org/10.1097/coh.0000000000000287.

[5] Y. Mu, S. Kodidela, Y. Wang, and S. Kumar. 2018. “The Dawn of Precision Medicine in HIV: State of the Art of Pharmacotherapy.” *Expert Opinion on Pharmacotherapy* 19 (14): 1581–95. https://doi.org/10.1080/14656566.2018.1515916.

[6] “The Future—and the End?—of AIDS.” *2019. Columbia University Irving Medical Center*. November 27, 2019. https://www.cuimc.columbia.edu/news/future-and-end-aids. (accessed Apr. 25, 2024).

[7] A. Talukdar and S. Pal, “Computational Approaches Toward Development of Topoisomerase I Inhibitor: A Clinically Validated Target,” *Elsevier eBooks*, pp. 441–462, Jan. 2021, doi: https://doi.org/10.1016/b978-0-12-822312-3.00018-7.

References

[8] “Papers with Code - HIV (Human Immunodeficiency Virus) Dataset,” *Paperswithcode.com*, 2022. https://paperswithcode.com/dataset/qm9-charge-densities-and-energies-calculated (accessed Apr. 25, 2024).
[9] K. Riesen and H. Bunke, “IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning,” *Lecture notes in computer science,* pp. 287–297, Jan. 2008, doi: https://doi.org/10.1007/978-3-540-89689-0_33.
[10] yeonseokcho, “BBBP Classification by Mol Descriptor,” *Kaggle.com*, Aug. 22, 2023. https://www.kaggle.com/code/yeonseokcho/bbbp-classification-by-mol-descriptor/notebook (accessed Apr. 25, 2024).
[11] “rdkit.Chem.Descriptors module — The RDKit 2024.03.1 documentation,” *Rdkit.org*, 2024. https://www.rdkit.org/docs/source/rdkit.Chem.Descriptors.html (accessed Apr. 25, 2024).
[12] “Datasets,” *Moleculenet.org*, 2024. https://moleculenet.org/datasets-1 (accessed Apr. 25, 2024).
[13] “AIDS Antiviral Screen Data - NCI DTP Data - NCI Wiki,” *Nih.gov*, 2021. https://wiki.nci.nih.gov/display/NCIDTPdata/AIDS+Antiviral+Screen+Data (accessed Apr. 25, 2024).
[14] “Getting Started with the RDKit in Python — The RDKit 2024.03.1 documentation,” *Rdkit.org*, 2024. https://www.rdkit.org/docs/GettingStartedInPython.html#list-of-available-descriptors (accessed Apr. 25, 2024).

References

[15] “Dataset Splits,” *TDC,* 2024. https://tdcommons.ai/functions/data_split/ (accessed Apr. 25, 2024).
[16] “Introduction to Scaffold Splitting - Oloren AI,” *Oloren.ai,* 2022. https://www.oloren.ai/blog/scaff-split (accessed Apr. 25, 2024).
[17] “Sustainable Futures / P2 Framework Manual 2012 EPA-748-B12-001 Appendix F. SMILES Notation Tutorial.” Available: https://www.epa.gov/sites/default/files/2015-05/documents/appendf.pdf
[18] “Dataset Cheatsheet — pytorch_geometric documentation,” *Readthedocs.io*, 2016. https://pytorch-geometric.readthedocs.io/en/latest/cheatsheet/data_cheatsheet.html (accessed Apr. 25, 2024).
[19] J. Li, D. Cai, and X. He, “Learning Graph-Level Representation for Drug Discovery,” *arXiv.org*, 2017. https://arxiv.org/abs/1709.03741v2 (accessed Apr. 25, 2024).
[20] Sefik Serengil, “A Gentle Introduction to ROC Curve and AUC in Machine Learning - Sefik Ilkin Serengil,” *Sefik Ilkin Serengil*, Dec. 10, 2020. https://sefiks.com/2020/12/10/a-gentle-introduction-to-roc-curve-and-auc/ (accessed Apr. 28, 2024).
[21] “How to explain the ROC AUC score and ROC curve?,” *Evidentlyai.com*, 2024. https://www.evidentlyai.com/classification-metrics/explain-roc-curve (accessed Apr. 28, 2024).
[22] “Graphical scheme of XGBoost model,” *ResearchGate*, 2023. https://www.researchgate.net/figure/Graphical-scheme-of-XGBoost-model_fig1_370000558 (accessed Apr. 30, 2024).

References

[23] D. Rogers and M. Hahn, “Extended-Connectivity Fingerprints,” *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, Apr. 2010, doi: https://doi.org/10.1021/ci100050t.

[24] J. Li, D. Cai, and X. He, “Learning Graph-Level Representation for Drug Discovery.” Available: https://arxiv.org/pdf/1709.03741v2

[25] W. Koehrsen, “Random Forest Simple Explanation - Will Koehrsen - Medium,” *Medium*, Dec. 27, 2017. https://williamkoehrsen.medium.com/random-forest-simple-explanation-377895a60d2d (accessed Apr. 30, 2024).

[26] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” *arXiv.org*, 2016. https://arxiv.org/abs/1609.02907 (accessed Apr. 30, 2024).

[27] Sergios Karagiannakos, “Best Graph Neural Network architectures: GCN, GAT, MPNN and more | AI Summer,” *AI Summer*, Sep. 23, 2021. https://theaisummer.com/gnn-architectures/ (accessed Apr. 30, 2024).

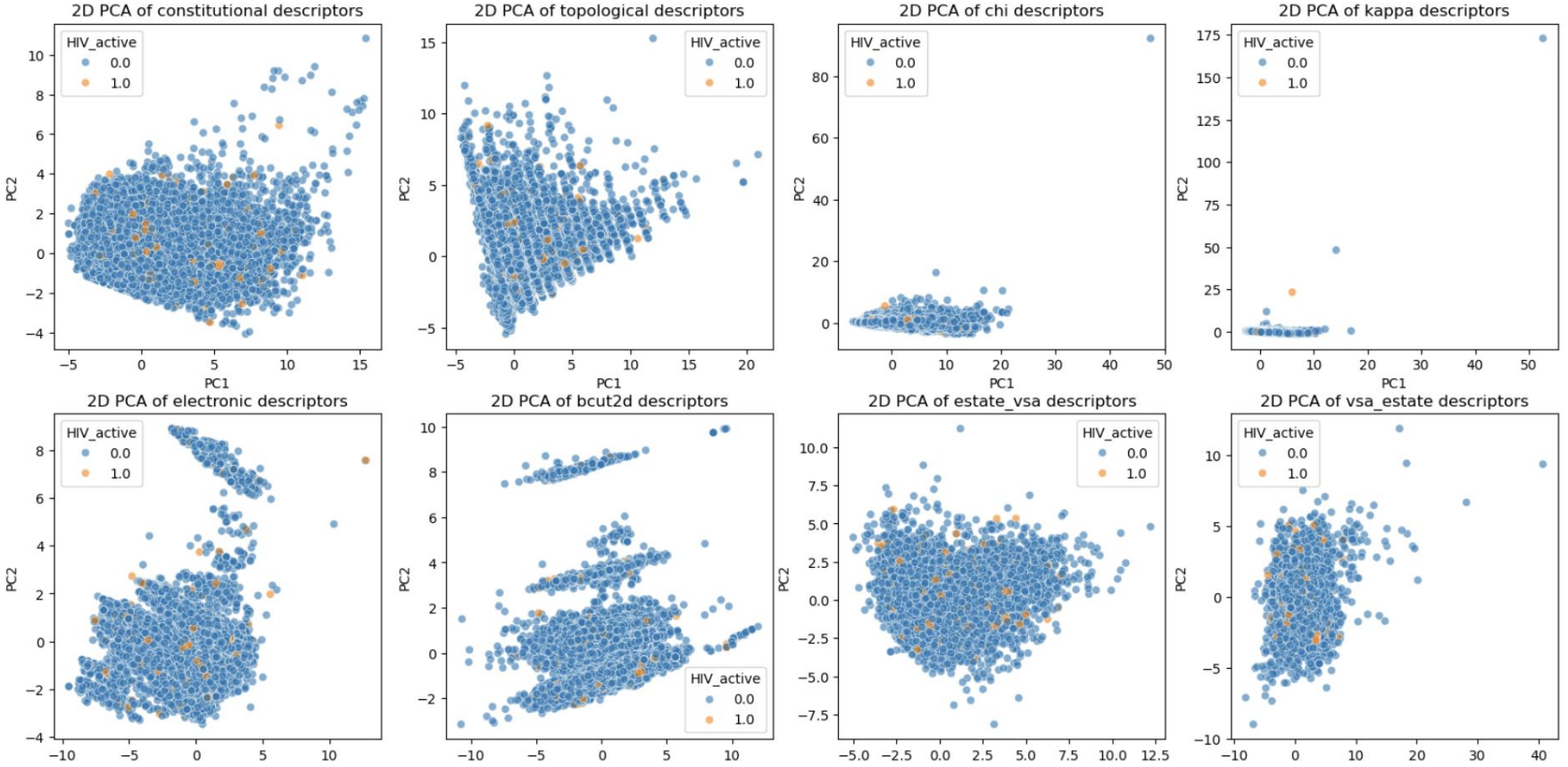
[28] “torch_geometric.nn.conv.GCNConv — pytorch_geometric documentation,” *Readthedocs.io*, 2024. https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GCNConv.html (accessed Apr. 30, 2024).

[29] J. Li, D. Cai, and X. He, “Learning Graph-Level Representation for Drug Discovery,” *arXiv.org*, 2017. https://arxiv.org/abs/1709.03741 (accessed Apr. 30, 2024).

Any questions?

Appendix

The General Descriptors



The Molecular Fingerprints