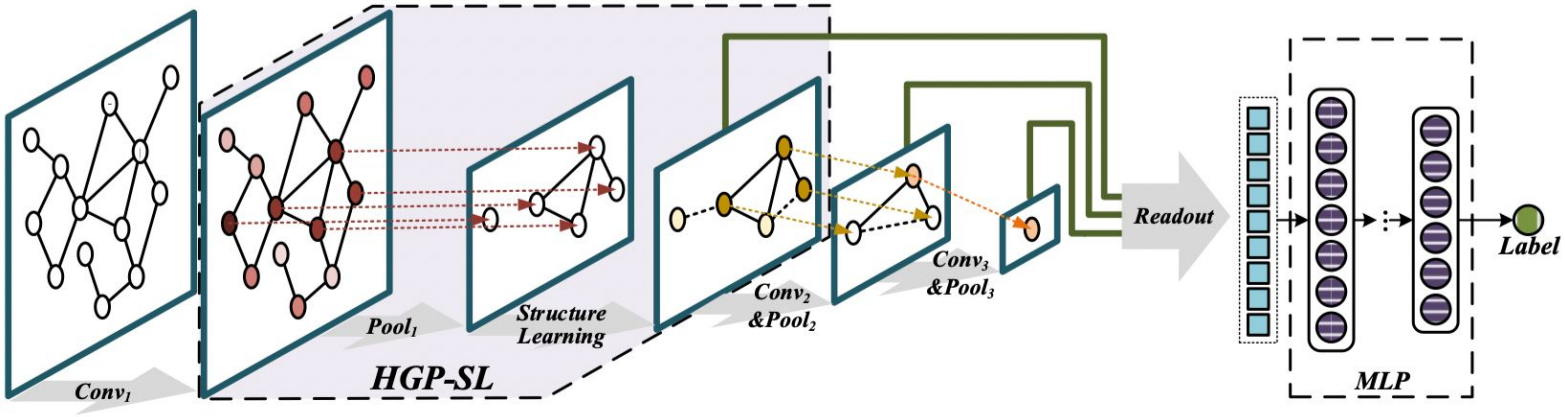
AML: Semester project presentation

Carlos Kirchdorfer, Oliver Aschwanden, Marco Heiniger

Table of Contents

Project Overview: A GNN project

Goal of this project: Explore possibilities of improving the approach of the paper **Hierarchical Graph Pooling with Structure Learning** by Zhang et al. (2019).



Problem formulation

Bioinformatics: Protein structures can be modeled as graph (nodes = amino acids).

- Two nodes are less than six Angstroms apart → there is an edge between them

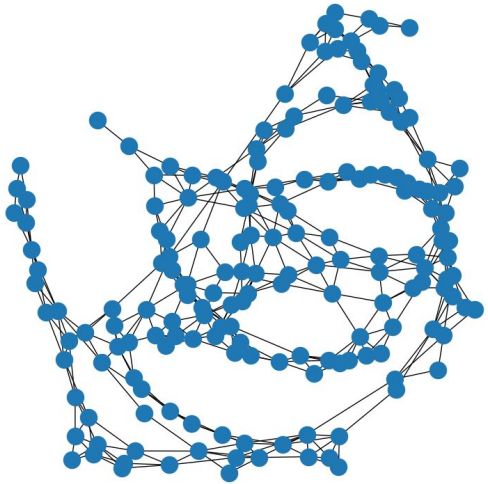
The **D&D dataset**:

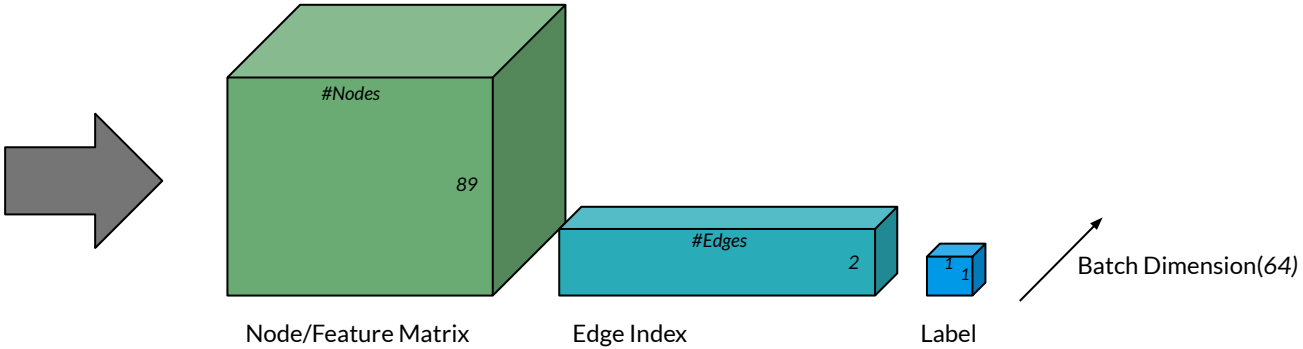
- n = 1178 protein graphs $G = \{G_1, G_2, \dots, G_n\}$
- A protein graph has the structure $G_i = (V_i, E_i, X_i)$ with n_i nodes and e_i edges:
  - $\boldsymbol{X}_i \in \mathbb{R}^{n_i * f}$ represents the node feature matrix, f is the dimension of the node attributes
  - **Binary classification**: Graph is an enzyme or not → target labels can be represented with label vector $\boldsymbol{y} \in \mathbb{R}^n$

Training: Training of model with a batch of protein graphs G_L whose label information is $\boldsymbol{y_L}$ (targets)

Task and Goal: Decide if the protein graph G_i should be classified as an enzyme $y_i = 1$ or not $y_i = 0$.

Model Input (D&D Dataset)



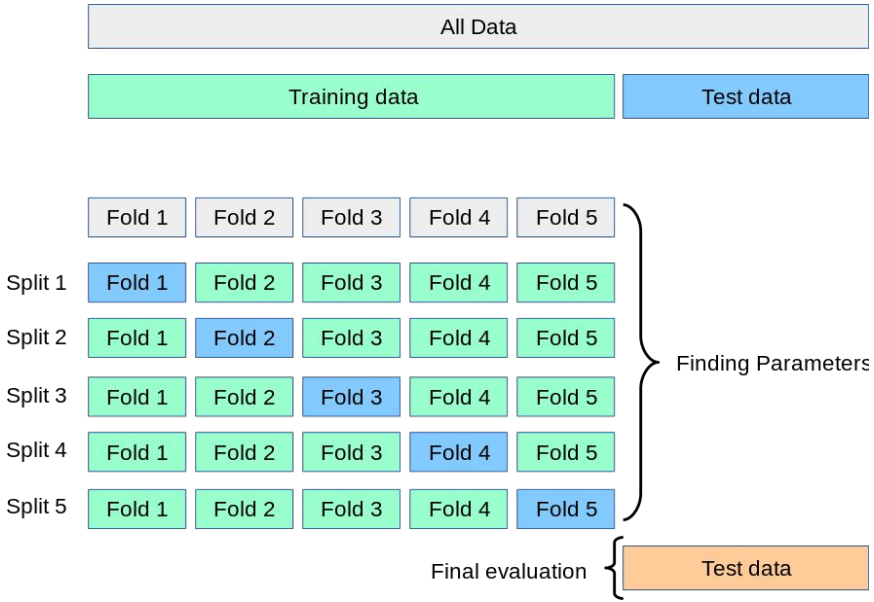


Planned self contributions

1. Rebuilding the proposed model from the original paper and bringing it to a newer Python version
2. Implementing the simple baseline model **Shortest-Path Kernel** by Borgwardt and Kriegel (2005)
3. The next step is to apply the following modifications and check them for possible improvements:
    1. Modify the read out function
    2. Introduce a deeper network
    3. Use a different node information score metric
    4. Different top-rank node selection (based on node information score)

Evaluation protocol and Baselines

Cross-Validation (CV):



Baselines:

- **HPG-SL**: paper's baseline, 80.96%* accuracy.
- **Shortest-Path Kernel:** statistical baseline

* Different evaluation protocol

Shortest-Path Kernel Baseline

- by Borgwardt and Kriegel (2005)
- All paths → NP-hard
- shortest path → solvable in polynomial time O(n^3)
- Algorithm:
  - transform input graphs into shortest path graphs
  - **Definition 3 (Shortest-path graph kernel)** *Let* G_1 *and* G_2 *be two graphs that are Floyd-transformed into* S_1 *and* S_2*. We can then define our shortest-path graph kernel on* $S_1 = (V_1, E_1)$ *and* $S_2 = (V_2, E_2)$ *as*

$$k_{shortest\ paths}(S_1, S_2) = \sum_{e_1 \in E_1} \sum_{e_2 \in E_2} k^{(1)}_{walk}(e_1, e_2),$$

where $k^{(1)}_{walk}$ *is a positive definite kernel on edge walks of length 1.*

Modifications: Deeper Network

Introduce a deeper network

- According to Zhang et al. (2019), model performs best with K = 2 layers
- We tried out K = 3, 4, and 5 layers

Motivation:

- *HGP-SL almost achieves the best performance across different datasets when setting K = 3, d = 128 and r = 0.8, respectively*. For D&D, they have set K = 2.
- Li et al. (2018): More convolutional layers do not necessarily lead to a better result
  - More layers → more difficult to train
  - Mix of features of nodes → makes them indistinguishable

Modification: Read out function

Modify the read out function

- Use all subgraphs as in the paper
- Use only the first subgraph
- Use only the last subgraph

$$\mathbf{r}_i^k = \mathcal{R}(\mathbf{H}_i^k) = \sigma\left(\frac{1}{n_i^k}\sum_{p=1}^{n_i^k}\mathbf{H}_i^k(p,:)\,||\,\max_{q=1}^{d}\mathbf{H}_i^k(:,q)\right)$$

$$\mathbf{z}_i = \mathbf{r}_i^1 + \mathbf{r}_i^2 + \cdots + \mathbf{r}_i^K$$

Motivation:

- **Use all:** Used by Zhang et al. (2019) and Zhang et al. (2023)
- **First subgraph:** Node representations may be well distinguishable, but representation might be too complex to find reliable information about class assignment
- **Last subgraph:** Opposite: More convolutional layers have been used
  - Li et al. (2018): Mix of features of nodes → makes them indistinguishable

Modifications: Node information score

Use a different node information score metric

- For the node information score, Zhang et al. (2019) use the Manhattan norm. We also tried out the Euclidean norm.

$$\mathbf{p} = \gamma(\mathcal{G}_i) = \|(\mathbf{I}_i^k - (\mathbf{D}_i^k)^{-1}\mathbf{A}_i^k)\mathbf{H}_i^k\|_1,$$

Motivation:

- Gao et al. (2022) state: E*uclidean distance as it represents a common similarity measure that is especially convenient for high-dimensional vectors that might be present in some graph data sets.*

Manhattan

$$|\boldsymbol{x}|_1 = \sum_{r=1}^{n} |x_r|$$

Euclidean

$$|\boldsymbol{x}|_2 = \sqrt{\sum_{r=1}^{n} |x_r|^2}$$

Modifications: Node selection

Different top-rank node selection (based on node information score)

- Essentially is about modifying the pooling ratio **r**
- We tried out *pooling ratio* **r** = 0.2, 0.5, 0.8

$$\begin{aligned} \text{idx} &= \text{top-rank}(\mathbf{p}, \lceil r * n_i^k \rceil) \\ \tilde{\mathbf{H}}_i^{k+1} &= \mathbf{H}_i^k(\text{idx}, :) \\ \mathbf{A}_i^{k+1} &= \mathbf{A}_i^k(\text{idx}, \text{idx}), \end{aligned}$$

Motivation:

- ***r** cannot be too small, otherwise most of the graph structure information will be lost during the pooling process* (Zhang et al., 2019).
- "*HGP-SL almost achieves the best performance across different datasets when setting K = 3, d = 128 and r = 0.8, respectively*"
- For D&D, ***r*** is only 0.3.

Training pipeline

1. Find the best hyperparameter values for the **HGP-SL model**
2. First, change only **one hyperparameter at a time** (using CV)
3. Then **grid search** (again using CV): Build model for every combination of hyperparameters specified and evaluate each model
    a. For layers K, we only trained on K = 2 and K = 3
4. Final Model: Identify best hyperparameters & train the model on the **whole training set** and report the performance on the **test set** (has not been touched so far)

Model Variations

	DistanceMetric	*Layers*	*PoolingRatio*	*Readout*	Validation Accuracy avg	min	max
(base)	Manhattan	2	0.3	[1,2]	76.0	74.0	79.8
(distance)	Euclidean				75.7 (-0.3)	71.6	77.8
(layers)		3			76.0 (=)	74.0	76.8
		4			76.7 (+0.7)	75.0	78.7
		5			75.5 (-0.5)	73.1	77.4
(pooling ratio)			0.8		76.9 (+0.9)	73.5	81.2
			0.5		77.0 (+1.0)	73.1	80.7
			0.2		75.0 (-1.0)	71.2	77.9
(readout)				[1]	76.1 (+0.1)	73.5	77.4
				[2]	76.1 (+0.1)	73.1	79.7
(grid search)	Manhattan	2	0.8	[1]	**77.2 (+1.2)**	-	-

Table 1: Modifications made to the HGP-SL architecture and their evaluation through K-fold cross-validation (K=5) on the D&D dataset.

Hyper Parameter Search*

Hyperparameter	Options	Selected Parameter
Layers	2, 3	2
Pooling Ratio	0.2, 0.5, 0.8	0.8
Distance Metric	Manhattan, Euclidean	Manhattan
Readout Layers	ALL, FIRST, LAST	LAST

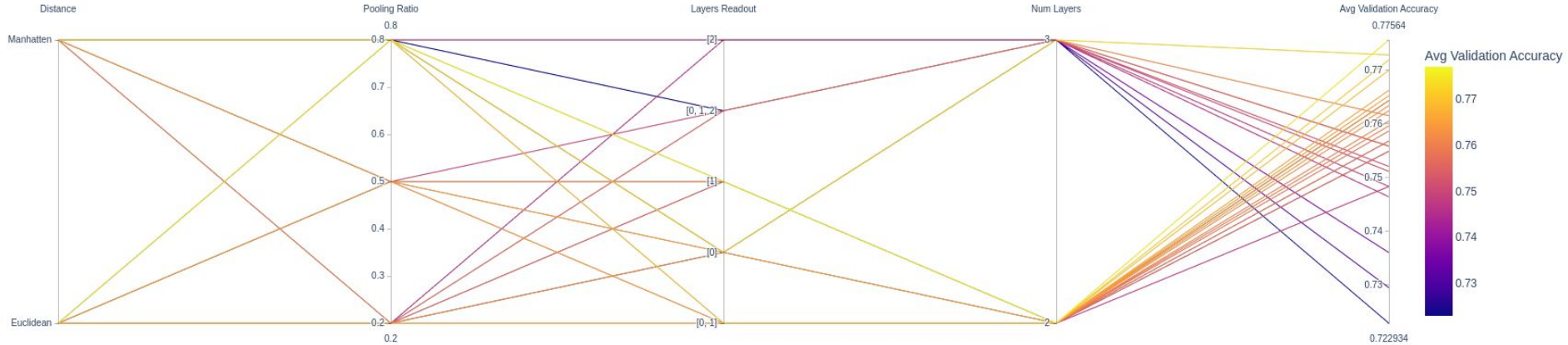
Table 2: Grid Search of Hyperparameters

Results

Model	Training	Test
Shortest Path Kernel	76.5	69.4
$HGP - SL_{original}$	84.4	78.6
Ours	85.6	78.6

Table 3: Comparison of training and test accuracies with respect to the shortest path baseline and the original $HGP-SL$ paper.

Hyper Parameter Analysis



Only 29/35 hyperparameter combinations could be trained because of limited compute!

Discussion

- Evaluation protocol (CV) has influence on accuracy data: 80.96% accuracy could not be beaten
- With 78.6% accuracy, our model has equal accuracy to HGP-SL baseline with CV. Shortest Path Kernel baseline was beaten.
- Grid search revealed that pooling ratio = 0.8 also for DD dataset is the best
  - When considering one change at time, r = 0.8 is better than r = 0.3 used by HGP-SL baseline → keeping structure information is important
- Deeper network did not bring any improvement → too much information is lost
- Contribution of node information score metric depends on combination
- Utilize some tool for metrics and training logs
- Slightly unbalanced data set split
- Better plan hyperparameter tuning: Informed tuning instead of extensive search

References

- Hierarchical Graph Pooling with Structure Learning (Zhang et al., 2019): https://arxiv.org/abs/1911.05954
- Hierarchical Multi-View Graph Pooling With Structure Learning (Zhang et al., 2023): https://ieeexplore.ieee.org/document/9460814
- D&D data set paper (Dobson and Doig, 2003): https://pubmed.ncbi.nlm.nih.gov/12850146/
- Shortest-path kernels on graphs (Borgwardt and Kriegel, 2005): https://www.dbs.ifi.lmu.de/~borgward/papers/BorKri05.pdf
- iPool—Information-Based Pooling in Hierarchical Graph Neural Networks (Gao et al., 2022): https://ieeexplore.ieee.org/document/9392315
- Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning (Li et al., 2018): https://arxiv.org/abs/1801.07606