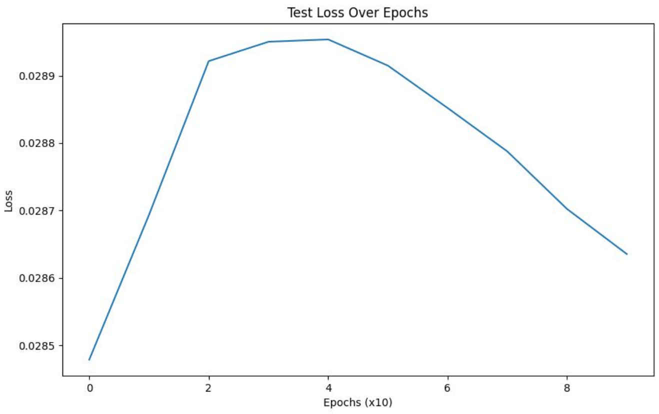
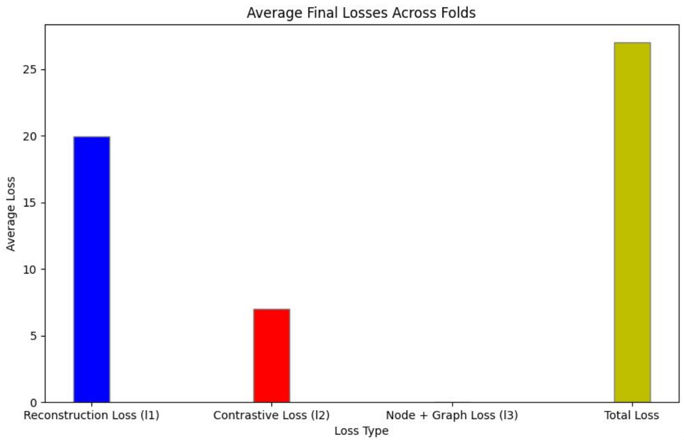
Last Week (cont.)

- Test loss going up while train loss goes down?
  - Most likely because test loss is computed only with l3 (which is negligible in the current context!)





Loss Function Reconfiguration

- Justification: Loss was getting overwhelmed by L3 in most attributed cases (with linear layers).
- L3 exclusion. In simple terms…
    - **Did not** lead to better results.
        - Pretty consistent with loss function without reconfigurations.

Loss Function Reconfiguration (cont.)

Datasets	GLADC	Original LG	LG - L3
MMP	**0.696 ± 0.042**	0.508 ± 0.138	0.547 ± 0.163
HSE	**0.618 ± 0.110**	0.551 ± 0.070	0.587 ± 0.067
p53	**0.649 ± 0.216**	0.497 ± 0.122	0.594 ± 0.099
BZR	**0.715 ± 0.067**	0.683 ± 0.045	0.661 ± 0.069
DHFR	**0.612 ± 0.041**	0.560 ± 0.053	0.448 ± 0.031
COX2	**0.615 ± 0.044**	0.595 ± 0.093	0.556 ± 0.082
ENZYMES	**0.583 ± 0.035**	0.529 ± 0.071	0.539 ± 0.038
IMDB	**0.656 ± 0.023**		
AIDS	0.993 ± 0.005	0.993 ± 0.005	0.993 ± 0.005
NCI1	0.683 ± 0.011	0.330 ± 0.016	0.318 ± 0.011

GC Layer Refactoring

- Tried running datasets previously tested on models with linear layers (as opposed to what was proposed in the paper) on refactored graph convolution ones.
    - The datasets were the attributed: AIDS, ENZYMES, DHFR, BZR and COX2.
        - As well as NCI1 (plain).
- Let’s take a look at the results…

GC Layer Refactoring (cont.)

Already implemented
with GC layers

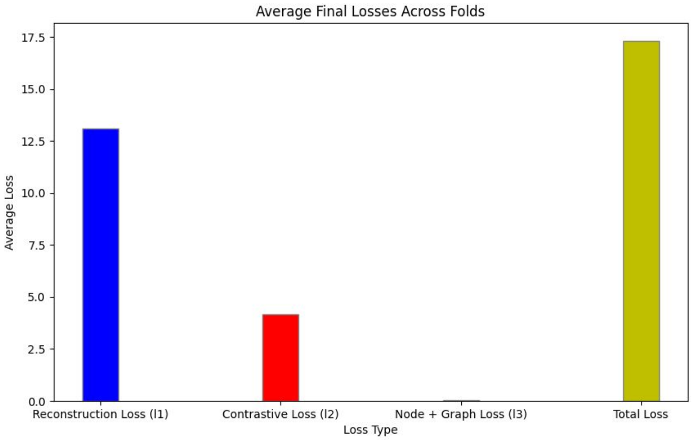
Datasets	GLADC	Linear Layers	GC Layers
MMP	**0.696 ± 0.042**		
HSE	**0.618 ± 0.110**		
p53	**0.649 ± 0.216**		
BZR	**0.715 ± 0.067**	0.683 ± 0.045	0.456 ± 0.037
DHFR	**0.612 ± 0.041**	0.560 ± 0.053	0.548 ± 0.013
COX2	**0.615 ± 0.044**	0.595 ± 0.093	0.488 ± 0.095
ENZYMES	**0.583 ± 0.035**	0.529 ± 0.071	0.509 ± 0.113
IMDB	**0.656 ± 0.023**		
AIDS	0.993 ± 0.005	0.993 ± 0.005	0.087 ± 0.038
NCI1	0.683 ± 0.011	0.330 ± 0.016	0.668 ± 0.034

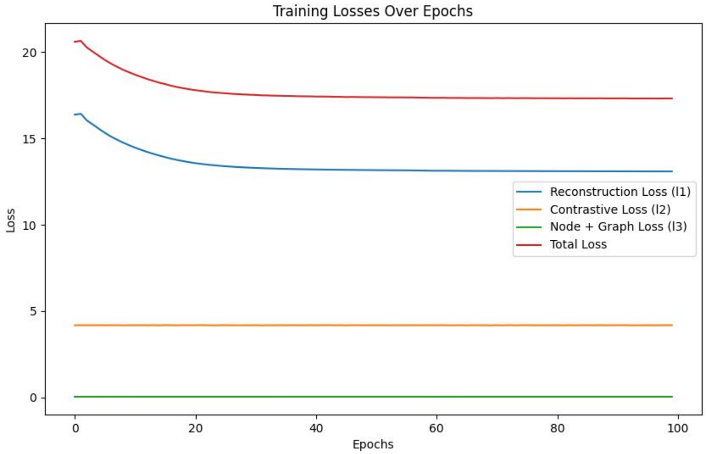
GC Layer Refactoring (cont.)

- Model performance appears to decrease.
- Let’s take a closer look at the loss…

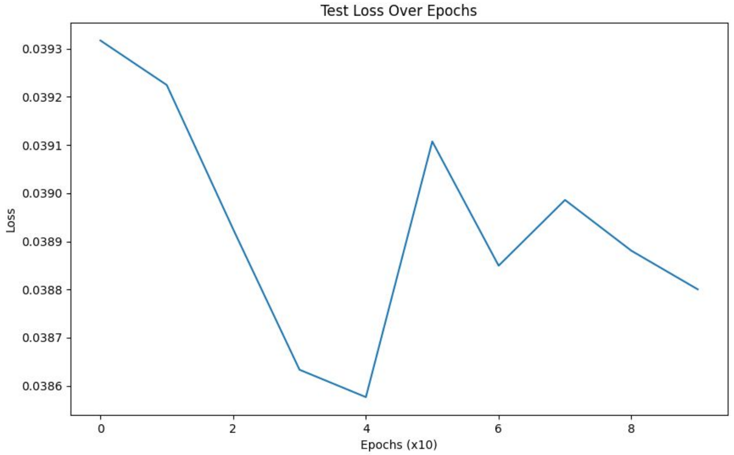
Loss Evolution (BZR)

- GC Layer Refactoring



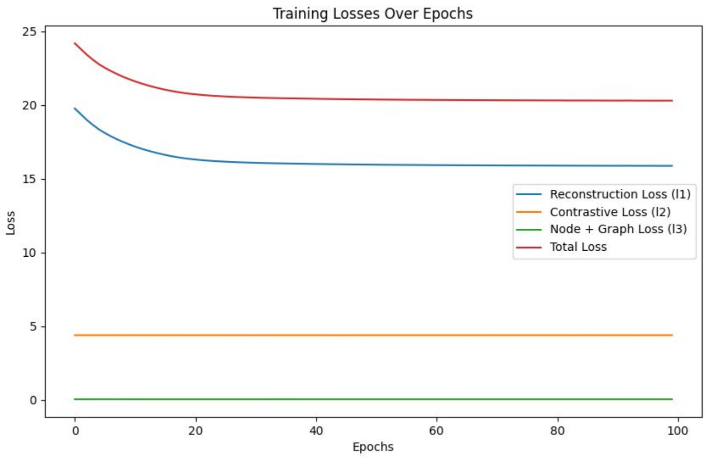


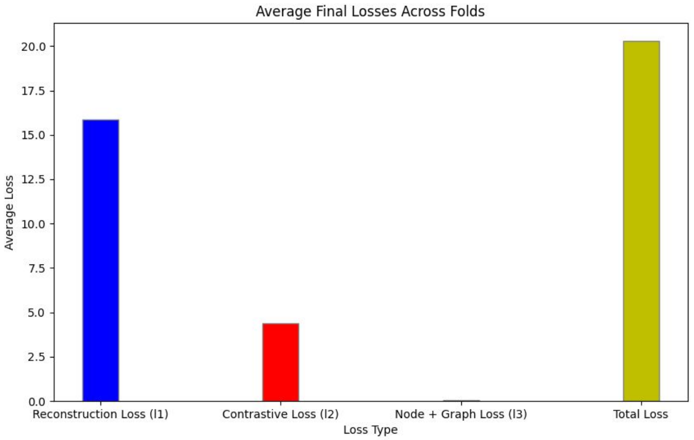
Test loss computed only from L3 for every DS



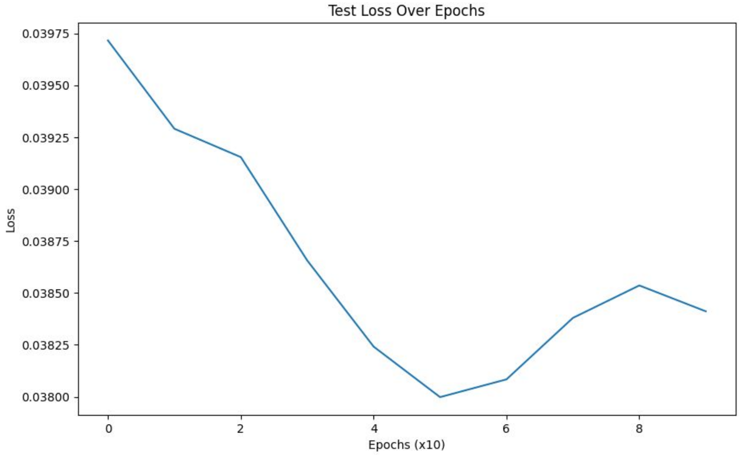
Loss Evolution (COX2)

- GC Layer Refactoring



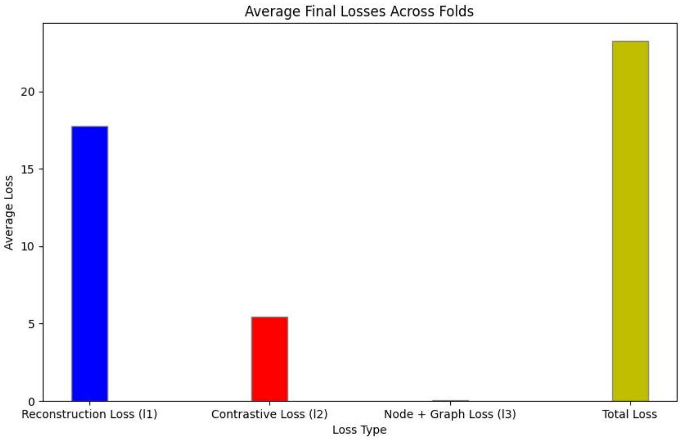


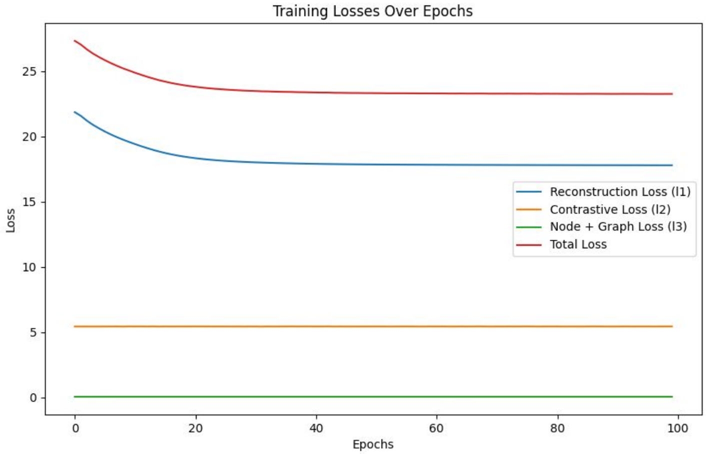
Test loss computed only from L3 for every DS



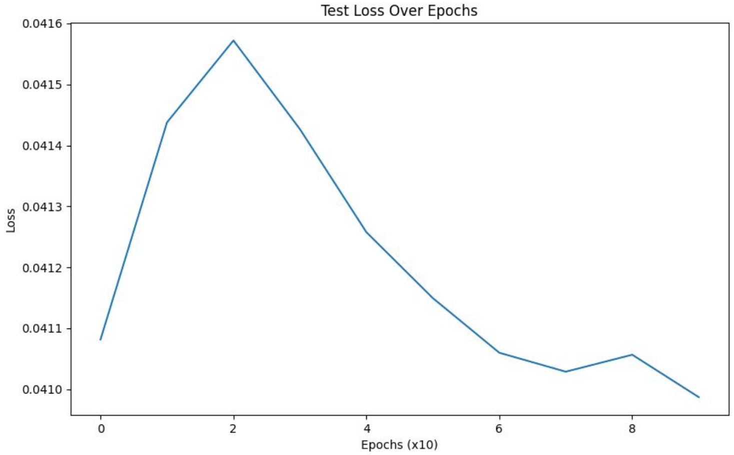
Loss Evolution (DHFR)

- GC Layer Refactoring



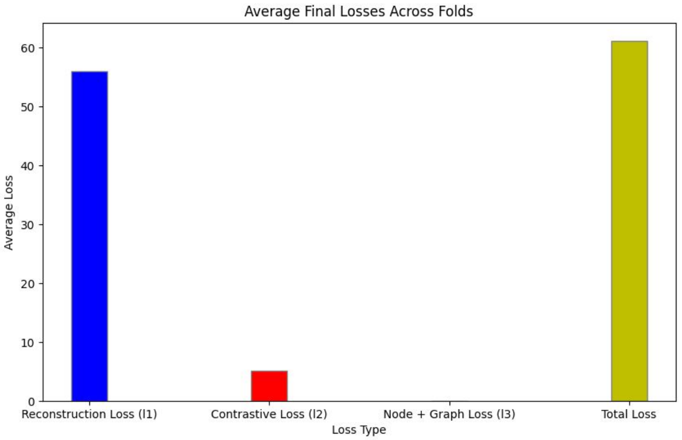


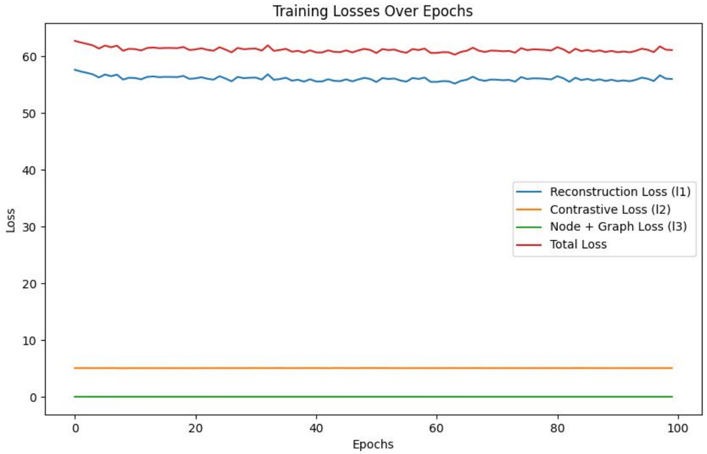
Test loss computed only from L3 for every DS



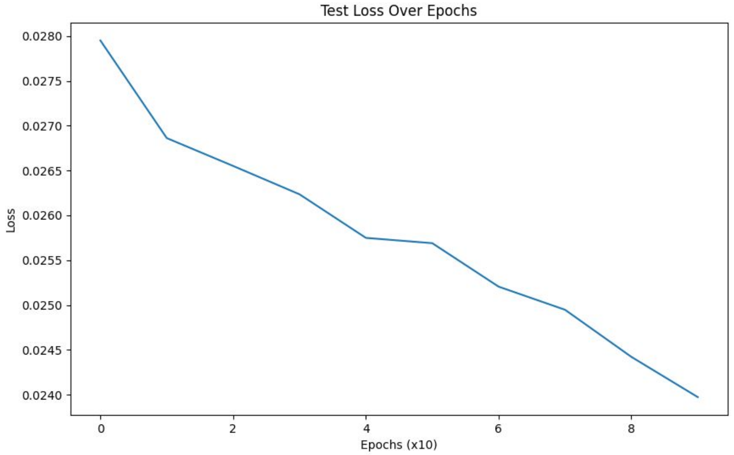
Loss Evolution (ENZYMES)

- GC Layer Refactoring



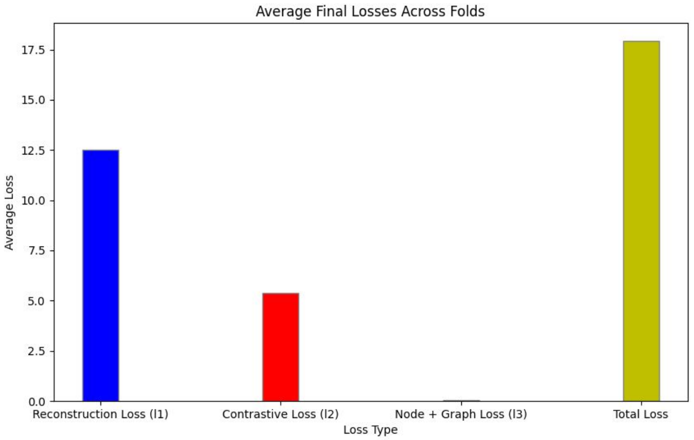


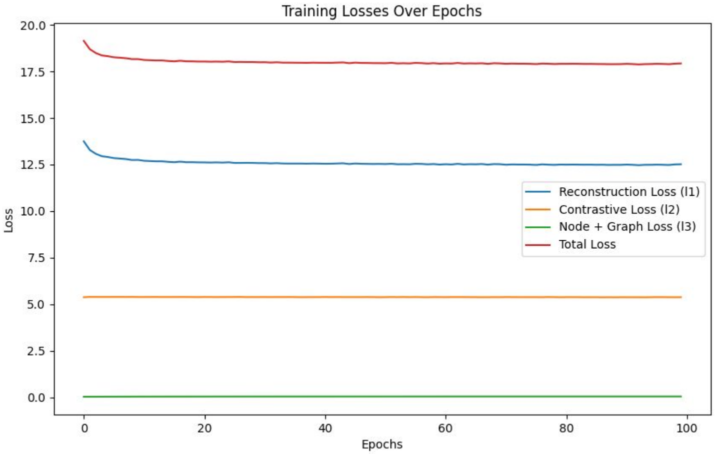
Test loss computed only from L3 for every DS



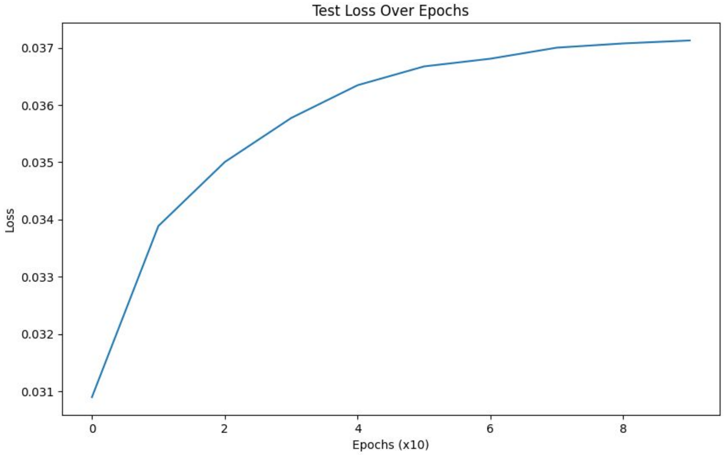
Loss Evolution (AIDS)

- GC Layer Refactoring



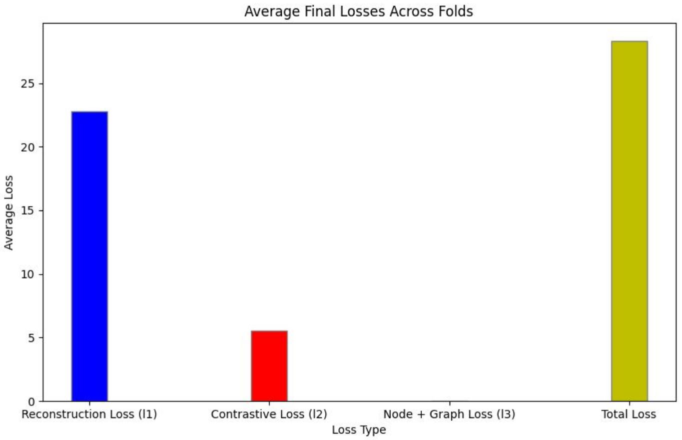


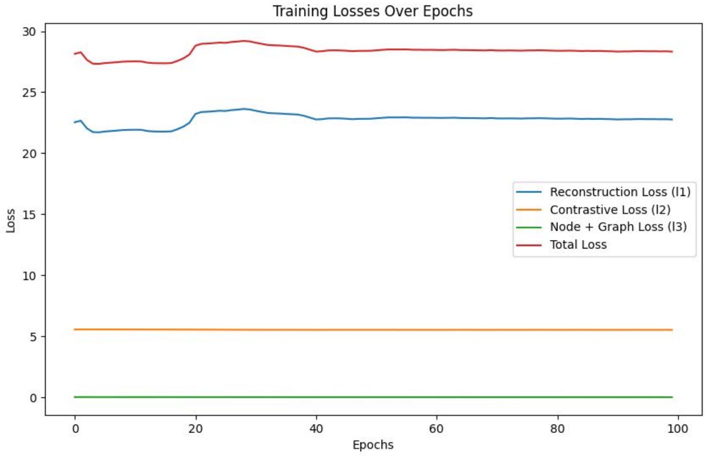
Test loss computed only from L3 for every DS



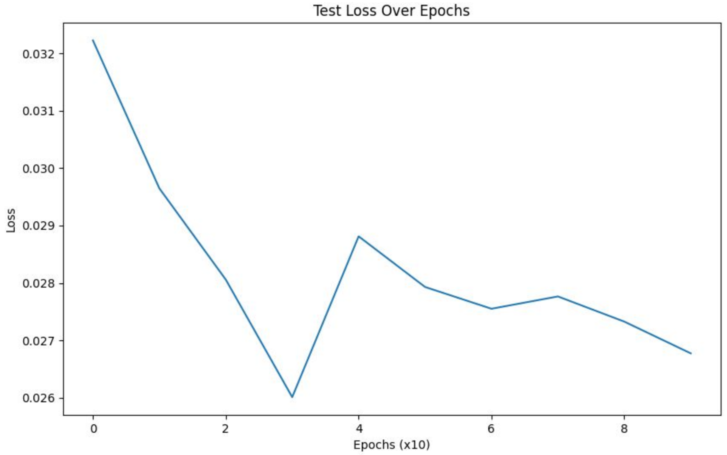
Loss Evolution (NCI1)

- GC Layer Refactoring





Test loss computed only from L3 for every DS



Discussion

- Peculiar trend: Models that use graph convolutions utterly minimize L3 loss (which makes sense as the embeddings are much richer).
- NCI1, a plain graph dataset, greatly increased its AUC score with new model implementation.
    - This results align with what was shown in the paper (so we can deduce that we were training the incorrect model for this particular dataset).
- This suggests that attributed graph datasets perform better with linear layers while plain graph perform better with graph convolutions.

Brainstorm

- Why is node degree not considered for attributed graphs?
  - Yes, we learn the predefined features, but isn’t the number of edges an important measure for detecting graph anomalies in an node basis?
- Why do graph convolutions fail (or at least get outperformed by linear layers) at handling attributed graphs?
  - Following up the previous point, can we combine both approaches (models with both linear and gc layers)?
- Graph attributes look really similar to node coordinates (x, y, z), at least for these datasets.
  - Can we apply translation invariant tactics (ie. EGNNS) for this problem?
-

Final thoughts

- The reason why they included two different types of models in their code is now clear:
  - Linear: attributed graphs
  - GC: plain graphs
- Perhaps asking the question (the one we discussed before) is not necessary anymore:
  - However it is still weird that they did not mention this in the paper…