Graph neural networks:

Report:

* Small vs large graphs
* Link vs node prediction, if both then graph prediction
* Node-, Link-, Graph-Prediction and Graph Generation
* Making myself familiar with libraries: geometric\_torch, …
* Making myself familiar with the Optimal Power Flow Problem: Generators, Slack node, …
* Reading the paper about GNNs and how they make the OPF problem more efficiently solvable, imitation learning/supervised, unsupervised solver, unsupervised solver with log barrier function
* Thinking about what to compare the GNN to: FFNN, normal GNN, Newton–Raphson algorithm, AD/DC OPF algorithm
* Question about how to use resistance/km: The is data for resistance per km and resistance between nodes. But we only need one of those as Ohm is getting linearly more with increasing length R=rho\*L/A, L distance. So I can just take into account only the Ohm from the edges!
* Possibly combining multiple grids in training
* Training with sampling node-wise sampling neighborhoods
* Regression task for generator nodes
* Hop distance choosing
* GraphSAGE model
* Relational Graph CNN
* Heterogenius Graph
* Also think about Joules’ heating of the grid: Q=I^2\*R\*t, meaning as little ampere as needed should be taken, but this cannot really be reduced as U=R\*I?
* For augmentation of grids because accuracy has to be with mean and variance compared to all of them solved by newton Raphson: Random walk or moving average of demands in grids?

Links:

* GNN literature: <https://github.com/naganandy/graph-based-deep-learning-literature/>
* GNN library based on pytorch: <https://pytorch-geometric.readthedocs.io/en/latest/>
* GNN library and datasets: <https://docs.dgl.ai/api/python/dgl.DGLGraph.html>
* GNN survey: <https://arxiv.org/pdf/1901.00596.pdf>
* GNN explanation and playground article: <https://distill.pub/2021/gnn-intro/#gnn-playground>
* Geometric torch tutorial: <https://pytorch-geometric.readthedocs.io/en/latest/tutorial/gnn_design.html>
* DGL tutorial: <https://docs.dgl.ai/tutorials/blitz/index.html>
* Unsupervised log barrier GNN Paper: <https://arxiv.org/pdf/2210.09277.pdf>
* GNN OPF paper: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9053140>
* Graph neural solver OPF: <https://pscc-central.epfl.ch/repo/papers/2020/715.pdf>

Questions:

* Going through the model changes and clarifying if it makes sense
* Hyperparameters
* K=? Can there be too many K?
* Batch\_size=?
* Same input for all 3 L networks? As it could be seen as latent space
* Phi from/to/loop sum?
* Samples to go though in one run? 🡪 nr\_samples, batch\_size
* Backwarding after each batch or through all samples?   
  🡪 Considering as many good updates as possible should be done
* Sample augmentation is rather large because of common factor division to equal demand and consumption increases demand compared to original 🡪 Lambda is almost always >0.5
* Memory efficient loading into GPU of all BLG? 🡪 Faster learning in time
* Latent dim in CGN input with phi out\_dim=1?