- Deadline: 28th March, 2021, 11:59PM
- Submission Material: PDF of the report.
  - A report detailing your
    - experimental framework: Parameters, Aggregation functions, dimensions, etc.
    - Explanation and **Analysis** of your experiments
    - A concluding discussion containing the key take-away points
    - In general, this is analogous to the experiments section of any paper.
    - Use the KDD 2021 latex template
  - In-class presentation covering all of the important points from your report.
  - Code base: Upload to GitHub and include the link in both your report and presentation. Try to have automated scripts for every experiment so that someone else can re-use your code as easily as possible
- Datasets
  - Brightkite: https://snap.stanford.edu/data/loc-brightkite.html
  - PPI: https://github.com/JiaxuanYou/P-GNN/blob/master/data/ppi.zip
  - Protein: https://github.com/JiaxuanYou/P-GNN/tree/master/data/PROTEINS_full

- Experiments
  - Quality
    - Metric: Average Precision, Recall, F1 score over 5-fold cross validation
    - Parameters: You can tune parameters and find the best choices.
    - Problem 1: Pair-wise node classification: Predict if two nodes are of same class label. For more information, read page 8 of https://arxiv.org/pdf/1906.04817.pdf
      - Dataset: Protein
    - Problem 2: Link Prediction
      - Dataset: PPI, Brightkite
      - Also do training time vs Quality
        - X-axis: training time. Use [5 min, 30 min, 2 hour, 6 hour, 12 hour]
        - Y-axis: Quality
    - Problem 3: Multi-class node classification
      - Dataset: PPI
        - Use inductive setting for GCN, GraphSage, GAT, and PGNN. You can use cross-entropy error as the loss function. Transductive for the remaining GNNs. Here you use the learned embeddings to learn a second-level classifier.

  - Interpretability, Design Choices, Parameters
    - The goal of this section is to understand the model embeddings better, verify how parameters affect performance, and possible alternatives to the design choices made by the authors. You will have to drive this component and come up with your own benchmarking ideas. You can use any dataset for this component.
    - Examples:
      - Deepwalk vs node2vec and GCN Vs GraphSage: when to use which one?

- Other Vertex invariants in Struc2Vec, GCNs.
- Aggregation functions in GCNs, GAT, PGNN
- Activation functions
- Layers
- Embedding Dimensions
- Verification against embedded ground truth data
- Adversarial cases
- Heuristics to improve quality or scalability