* Intra Layer
  + Batch Norm = True
  + Dropout = False (weil GNNs meistens underfitten und nicht overfitten)
  + Activation = prelu
  + Aggregation = sum
* Inter Layer
  + Anzahl pre-, post-, und message passing layer hängt sehr vom Task ab
  + Skip layer mit concatenation waren nützlich
* Learning configurations
  + Adam Optimizer und grosse Anzahl training epochs (e.g. 400) sind besser
  + Anzahl Batch size und learning rate sind schwierig zum entscheiden
* Design Space ist abhängig vom Task. Man kann diese in zwei Gruppen unterteilen
  + Tasks rely on feature information
    - Classification tasks bei welchem input graphs high-dimensional features haben
    - Top Architekturen
      * 1-2 Pre Layers
      * 8 MP Layers
      * 2 Post layers
      * Skip-sum Connectivity
      * Sum Aggregation
  + Tasks rely on structural information
    - Nodes haben wenige features
    - Predictions hängen hauptsächlich von Graph Struktur ab
    - Top Architekturen
      * 2 Pre layers
      * 6 MP Layers
      * 2 Post layers
      * Skip-cat connectivity
      * Mean aggregation