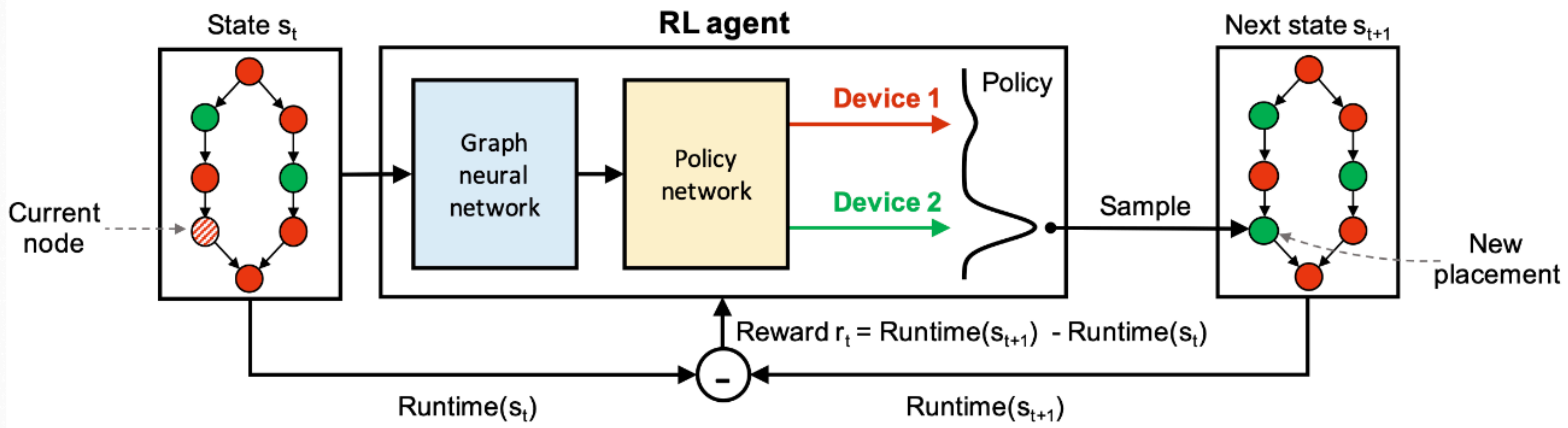
Alternative Graph Embeddings for Placeto

Zak Singh, 29/11/21

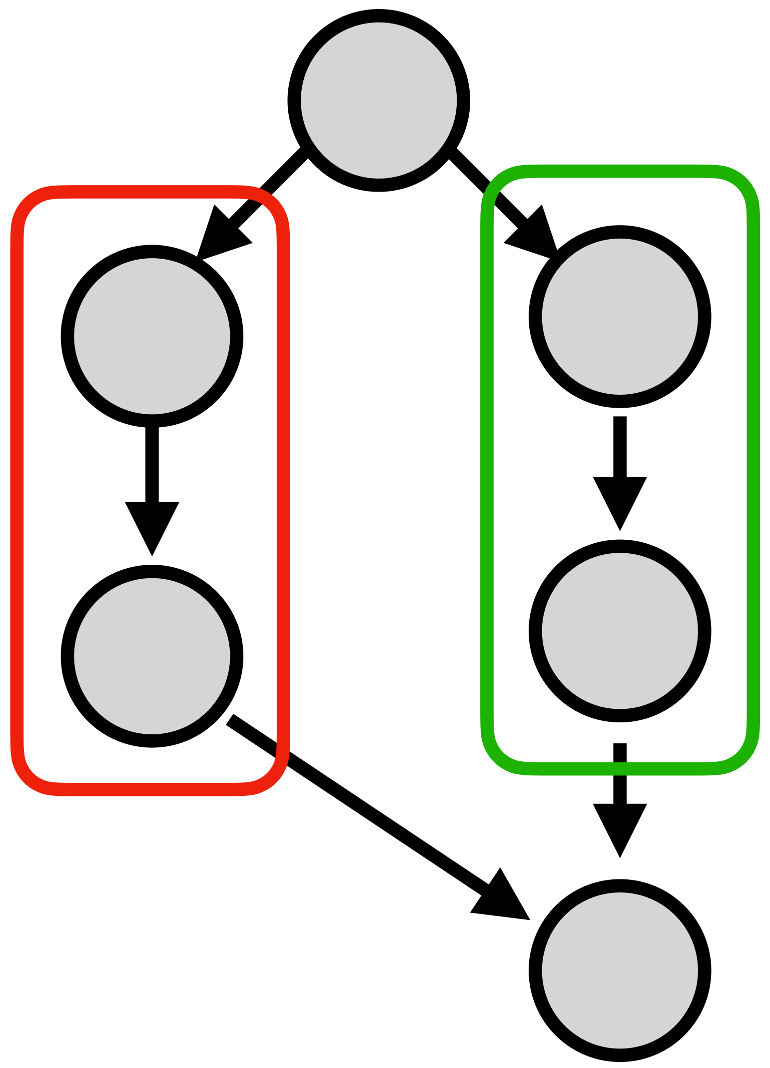
What is Placeto?

- Placeto is a method for device placement (assigning operations in a computation graph to devices)
- Why does it exist?
    - Existing RL-based placement solvers must be **trained for each computation graph individually**
        - In some cases upwards of 24hrs
    - **Placeto's main goal**: generalize to unseen computation graphs

Placeto's Iterative Placement



Co-location heuristics



6 -> 4 placements

- Problem: Tensorflow graphs can have tens of thousands of nodes, would take too long to process each operator iteratively
- Solution: group operators via heuristics
  - If operation A is only used by operation B, they are co-located
  - All operations in an LSTM "step" are co-located
- This shrinks problem space: no longer finding placement for ALL nodes; we only have to solve placement for each group
- Required to make training time reasonable

Model	#operations	#groups
RNNLM	8943	188
NMT	22097	280
Inception-V3	31180	83

Table 1. Model statistics.

(Slide adapted from my prev. presentation on Mirhoseini et al.'s work)

Graph Embedding

- Map each “group” of operators to a representation vector which encodes its **neighborhood** information
- Goal: groups of operators from similar graphs get mapped to similar representations
  - **Generalizability!**
- Implemented via traditional bidirectional messaging passing, plus…
- Each node gets “pooled attributes” appended to its representation to capture **regional** information
  - Set of all upstream nodes
  - Set of all downstream nodes
  - Set of unreachable nodes

Limits of Generalizability

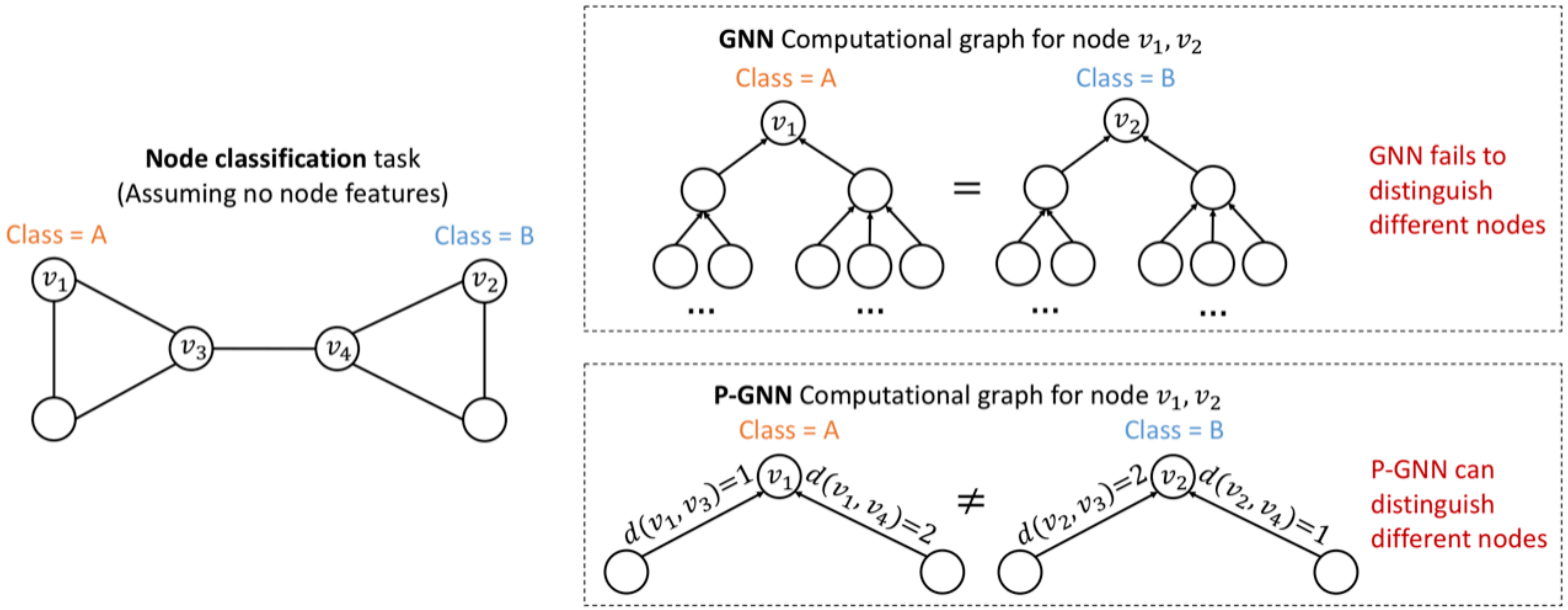
- Placeto's authors only show that the learned policy can generalize to "computation graphs from the same family as the training set"
- Meaning: if the policy is trained on convolutional networks, it can only place other CNNs
- Questions remain:
  - Why can't the policy be trained on a set consisting of multiple types of networks (CNNs + Transformers + MLPs etc…)?
  - How limiting is this? No benchmarks on "cross-family" placement are provided
  - Is it caused by the graph embedding procedure?

Proposed: Alternative graph embeddings

- “Pooled attributes” are one of many solutions to encode regional graph information into node representations
- GNN literature has papers dedicated solely to this problem domain
  - Example: Position-aware GNNs
- Proposed work: extend Placeto with these alternative graph embedding procedures, benchmark vs. “pooled attributes” approach
- Understand the value of contextual information in operator placement decisions
  - Is it the limiting factor in Placeto’s generalizability issues?

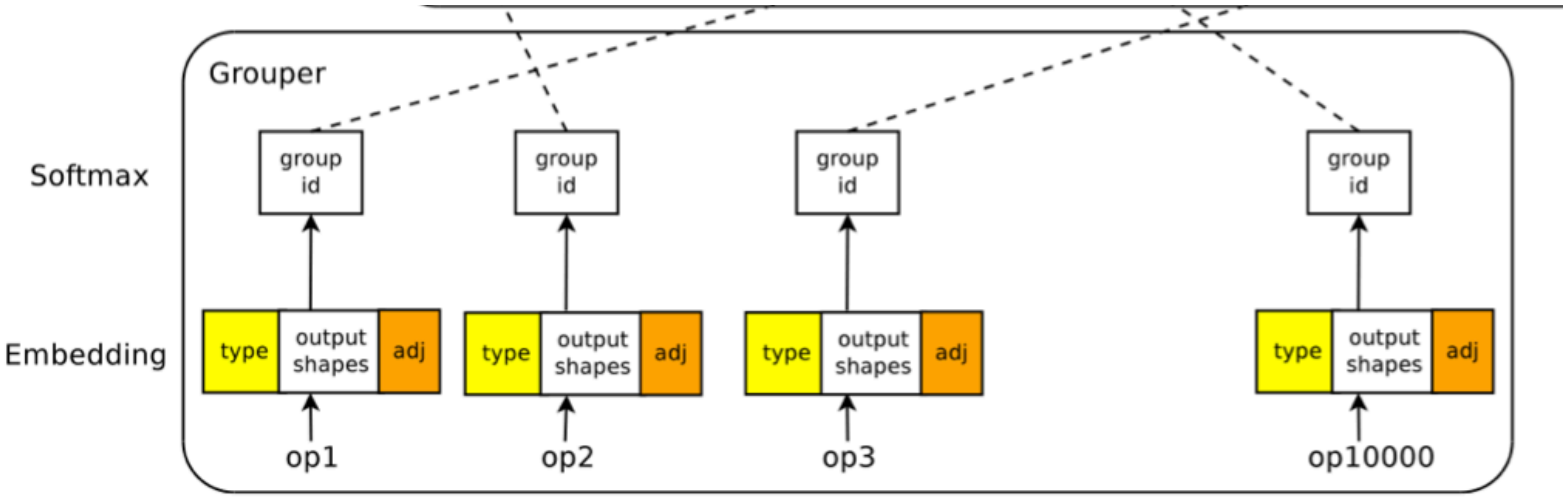
Position-aware GNNs

- Goal: learn position in broader graph structure
- Node position can be captured by quantifying the distance between each node and a set of “anchor sets”
- Anchor sets are chosen randomly
- Process can be repeated multiple times (similar to message passing)



Extension: Automatic Grouping

- Placeto's colocation heuristics are manual
- Could implement the network used by Mirhoseini et al. (discussed last week) to learn these groups instead of relying on heuristics
- Would fully automate placement and make this scalable to any computation graph



Challenges so far

- Placeto's published GitHub repo is missing some files, code doesn't compile as-is…
  - Mostly utilities they used during development (simplified graphs to test on, benchmarking code, etc)
  - Looks like they selectively published files and omitted ones they didn't think were needed for reproducing their results, probably an honest mistake

Timeline

- 29/11-3/12: Finish repairing codebase, replicate results from the paper
- 6/12-10/12: Implement alternative graph embeddings (Directed Acyclic GNNs, Position-aware GNNs)
- 13/12-17/12: Benchmark alternative graph embeddings, develop automatic grouping if time permits
- 17/12-deadline: Draft report

Questions?