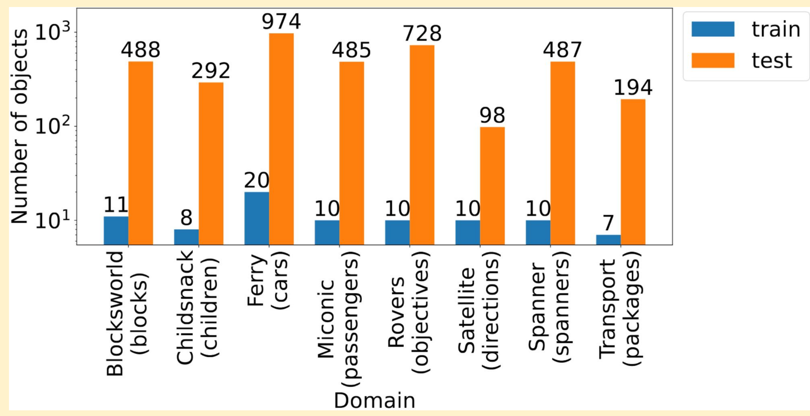
Graph Learning for Planning:
The Story Thus Far and Open Challenges

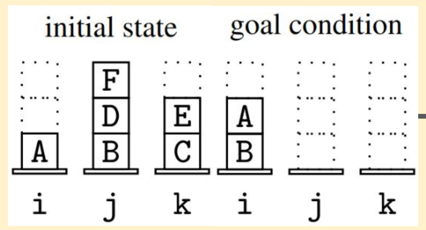
Dillon Z. Chen Mingyu Hao Sylvie Thiébaux Felipe Trevizan

Generalised Planning

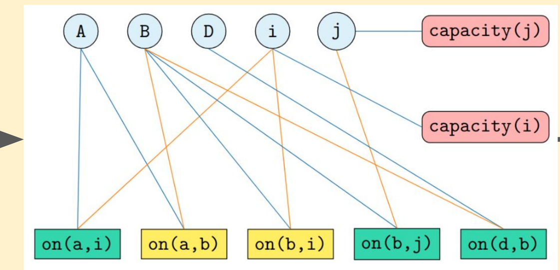
- Focus on generalisation across number of objects
- Train on small problems
- Test on large problems



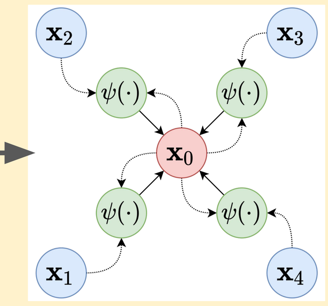
Graph Learning Architectures



convert task to graph



put in graph learning model



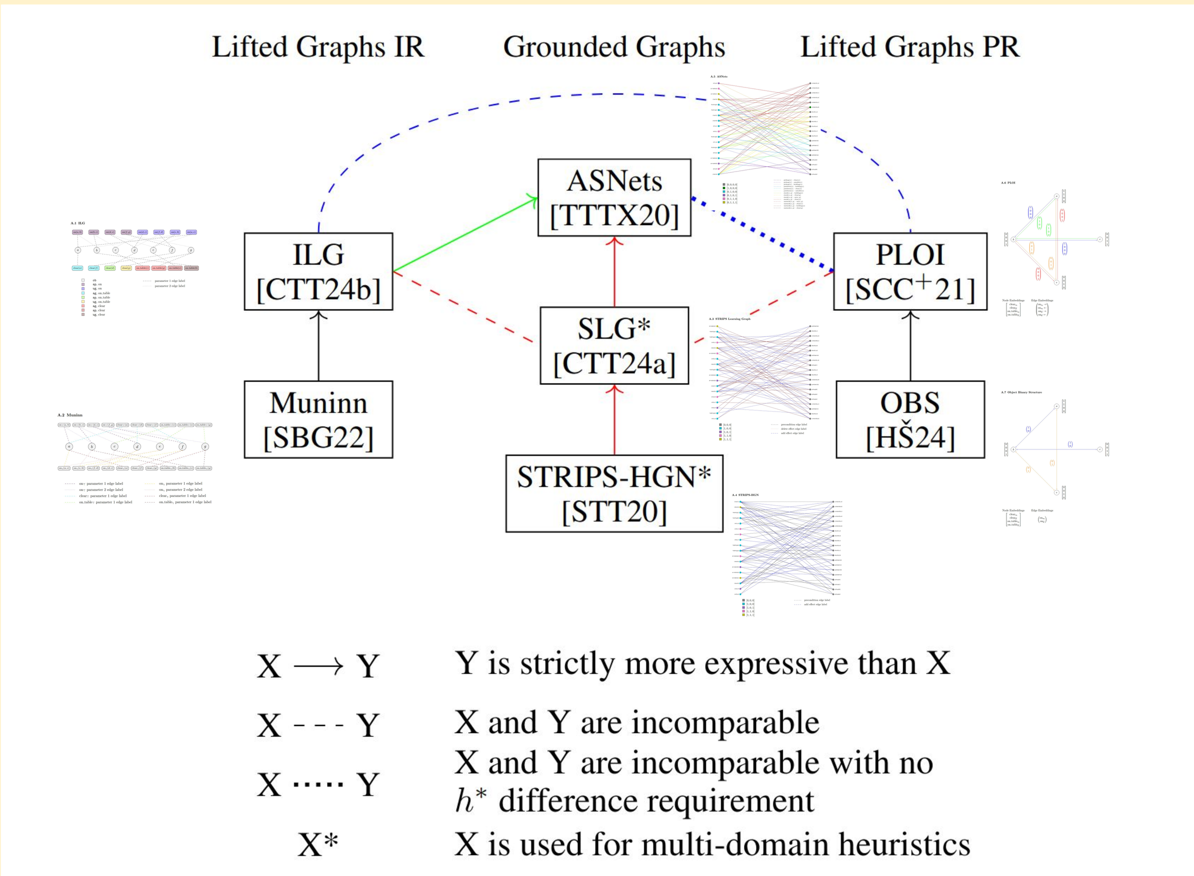
The Story Thus Far: 3 Key Insights

(1) Your Graph Learning Model Does Not Matter

- How to compare different architectures?
- architecture = representation + learning model
- Representation determines expressivity

In light of this, we

- Unify and taxonomise graph representations
- Theoretically compare representation power

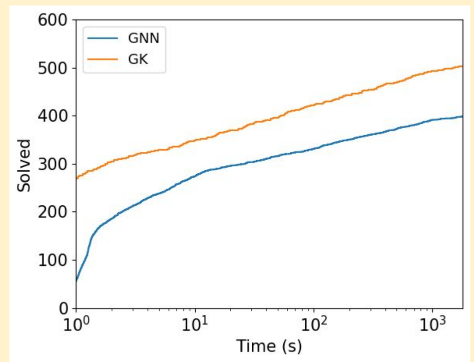


(2) Deep Learning is Most Overrated for Planning

- Idea: Graph Kernels (GKs) at least as expressive as Graph Neural Networks (GNNs) [Chen and Trevizan and Thiébaux, ICAPS-24]
- cheaper to train and evaluate = faster search

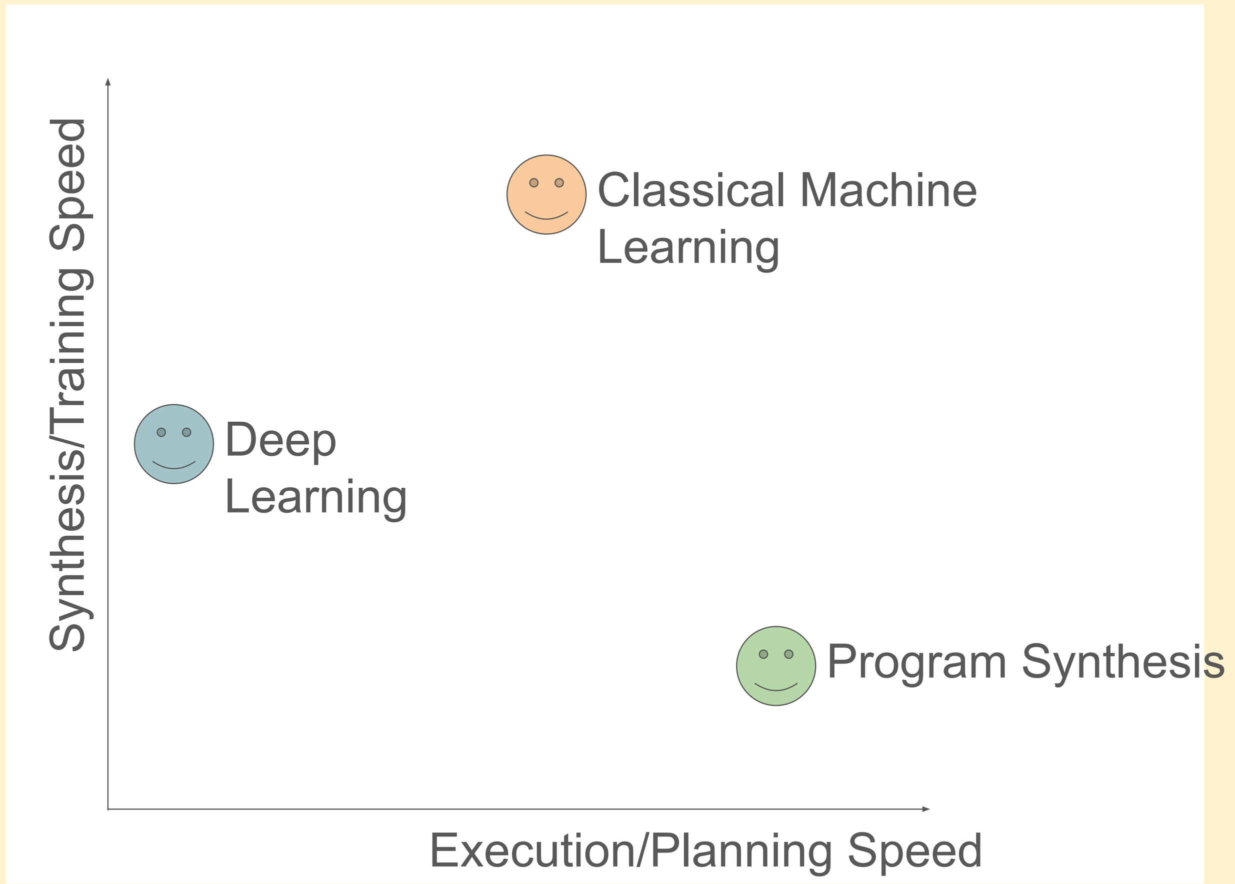
GNN vs GK train time (s)
Note the LOG scale

GNN vs GK parameters



Cumulative cov./time

GNNs also have access to GPUs

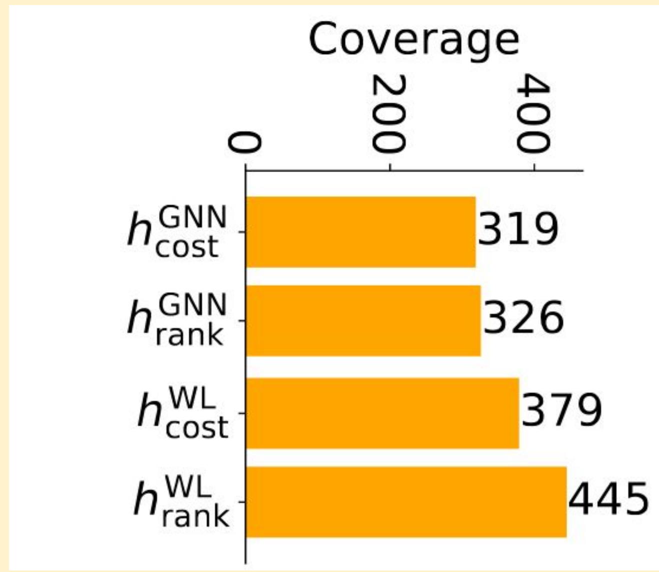


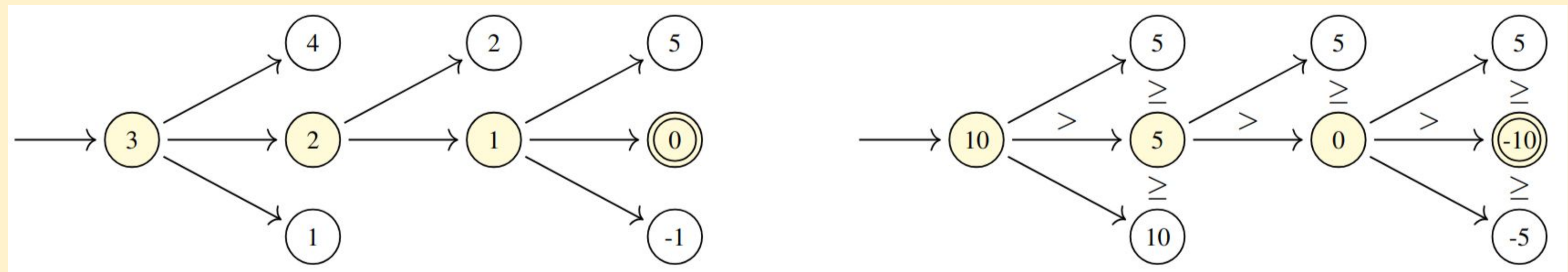
(3) Learn Rankings Instead of Hard Targets

- Supervised learning requires labels
- Usually learn from optimal heuristics or policies

Instead, learn to **compare** states

- Relax the optimisation criterion
  - → no longer learning an NP-hard target
- Get additional data from plan traces for free
- Reduce overfitting to targets vs. opt. heuristic learners





Open Challenges: 5 Problems From ML

(1) Expressivity

Expressivity determines what domains your model can solve.

(2) Generalisation

There is minimal generalisation theory with learning to plan.

(3) Optimisation

Choice of optimisation depends heavily on the domain.

(4) Collecting Data

When do we know we have collected enough data?

(5) Fair Comparisons

Model performance is not robust to training data and parameters