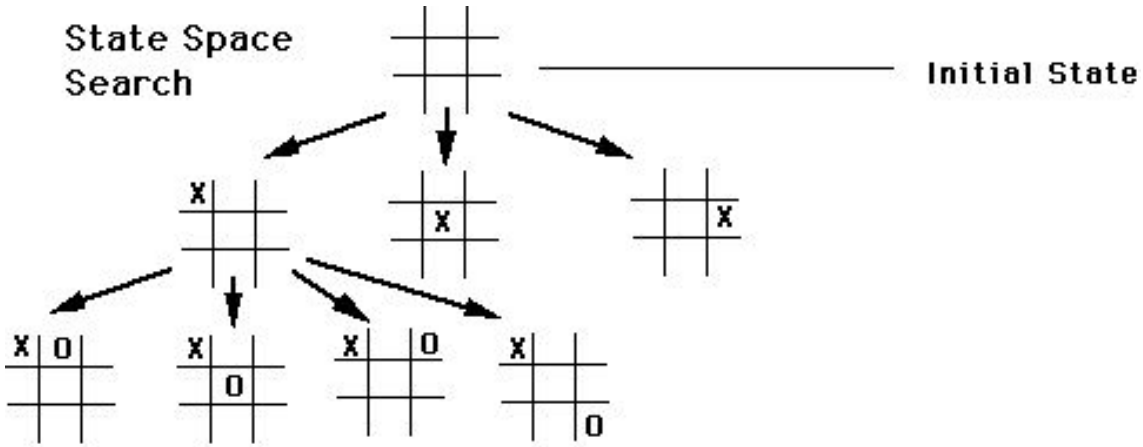
PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
GRADUATE PROGRAM IN COMPUTER SCIENCE

Learning Search Heuristics
by Graph Convolutional Networks

Pedro Ballester

Heuristics

- Searching is computationally expensive



Heuristics

- We try to solve this with heuristics

$$f(x) = g(x) + h(x)$$

Heuristics

- Finding a good heuristic can be quite demanding
- Tradeoff (inefficient x uninformative)

Heuristics

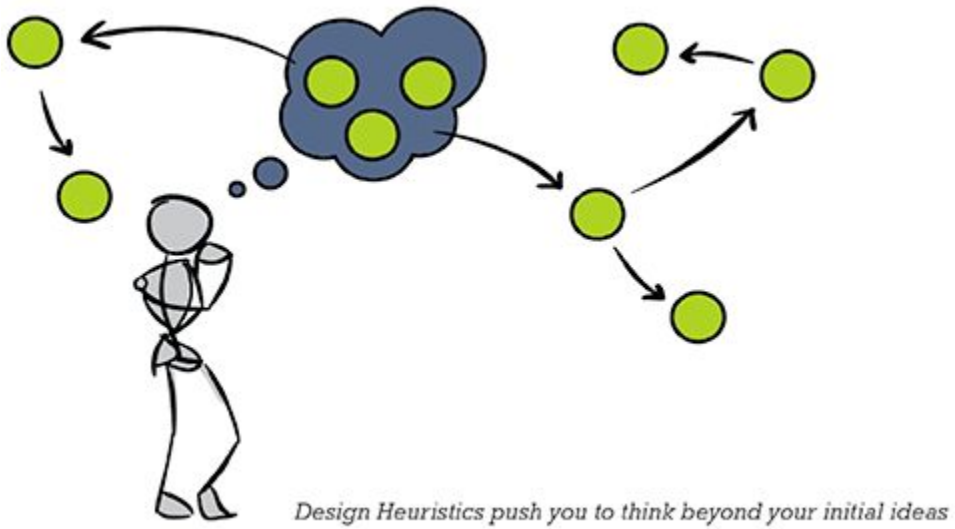
- Off-the-shelf
- Domain Specific

Heuristics

- Off-the-shelf

Heuristics

- Domain Specific



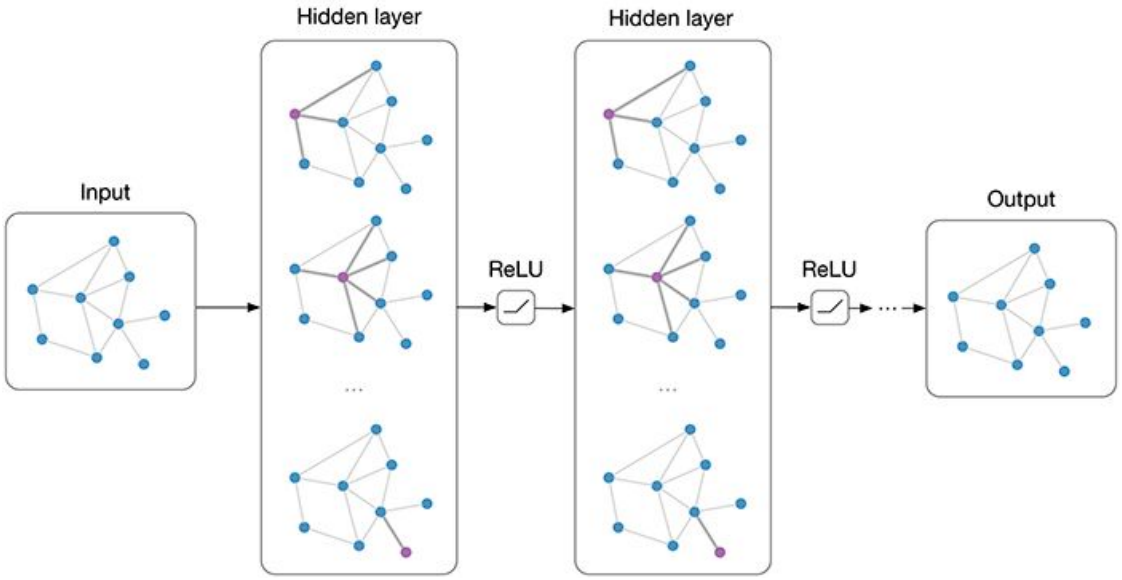
Design Heuristics push you to think beyond your initial ideas

Problem

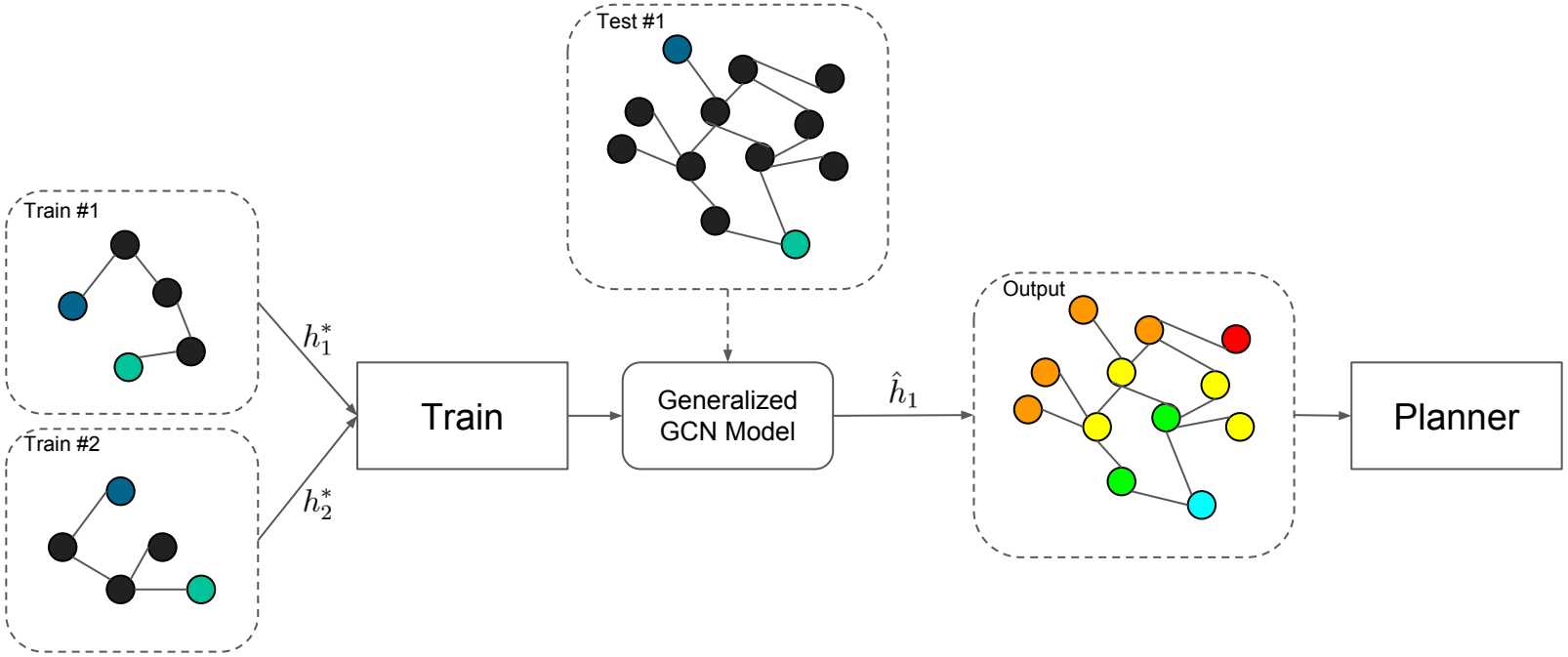
- How to find good and domain-specific heuristics without human knowledge?

Possible Solution

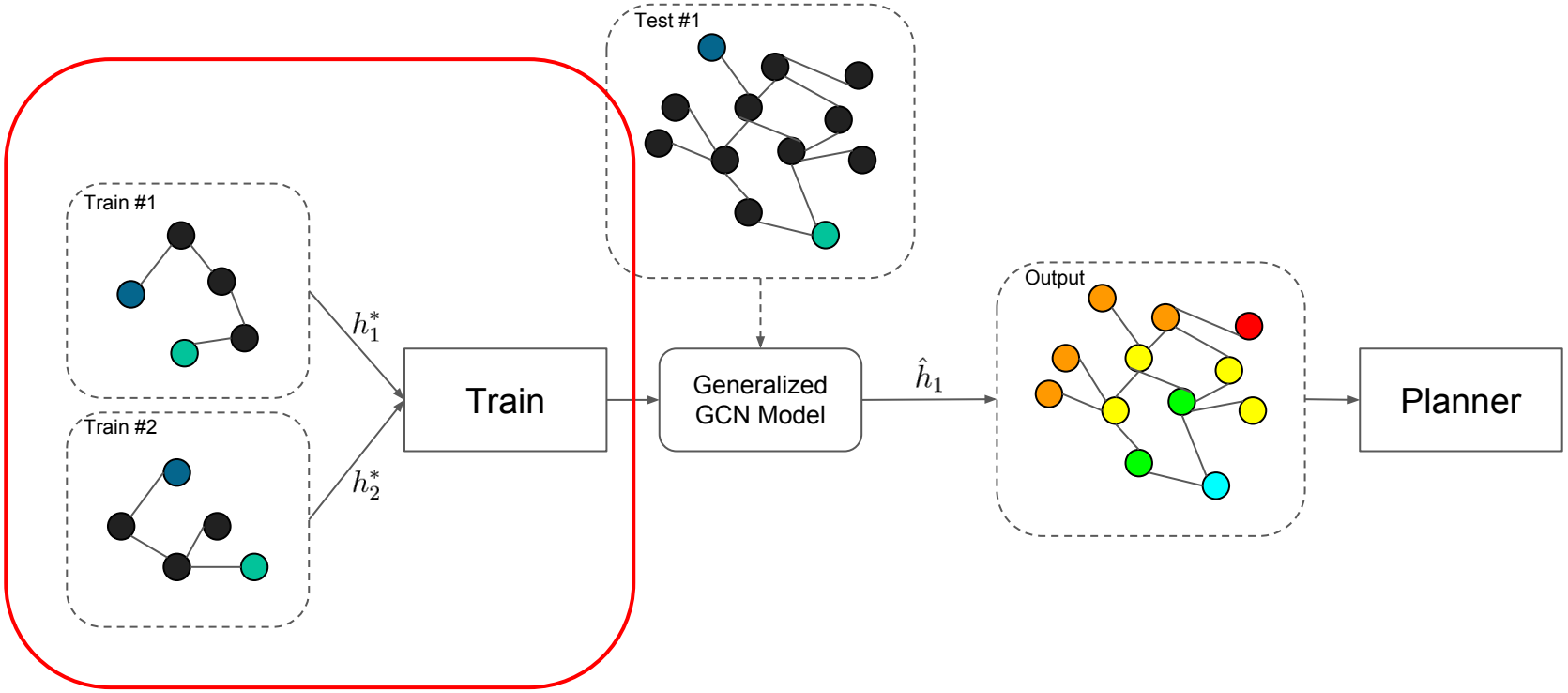
- Induce graph-based models with Deep Learning (GCN)
- The graph should output the heuristic value for an expanded frontier



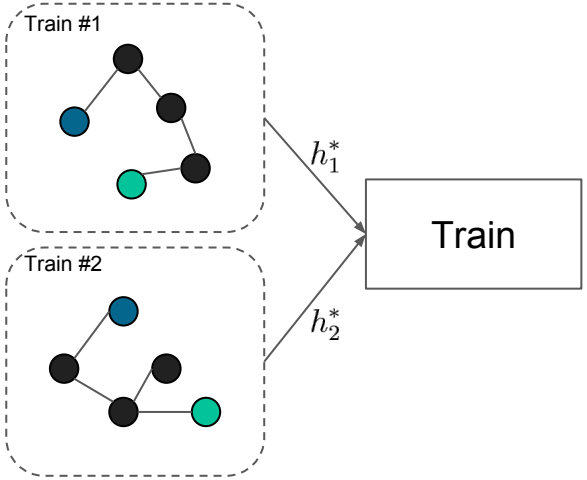
Proposed Method



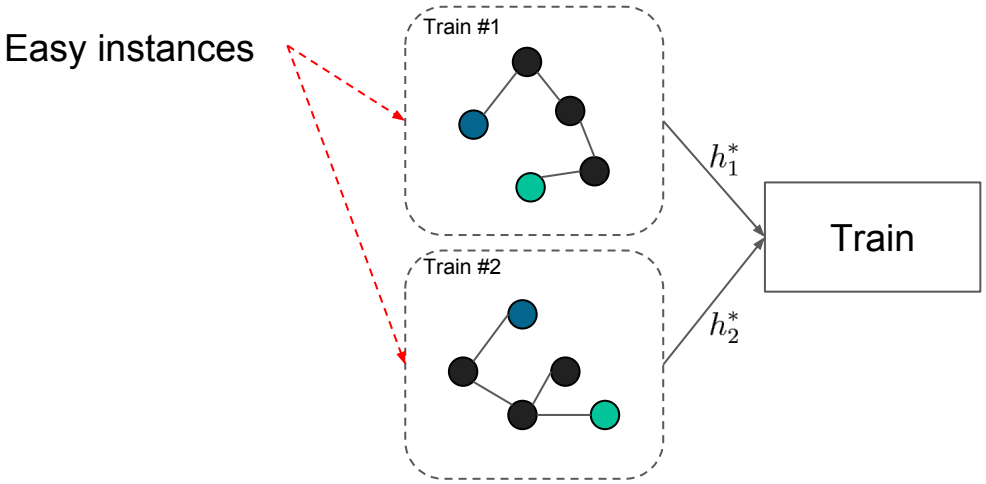
Training


Test #1
Train #1
Train #2
h_1^*
h_2^*
Train
Generalized
GCN Model
$\hat{h}_1$
Output
Planner

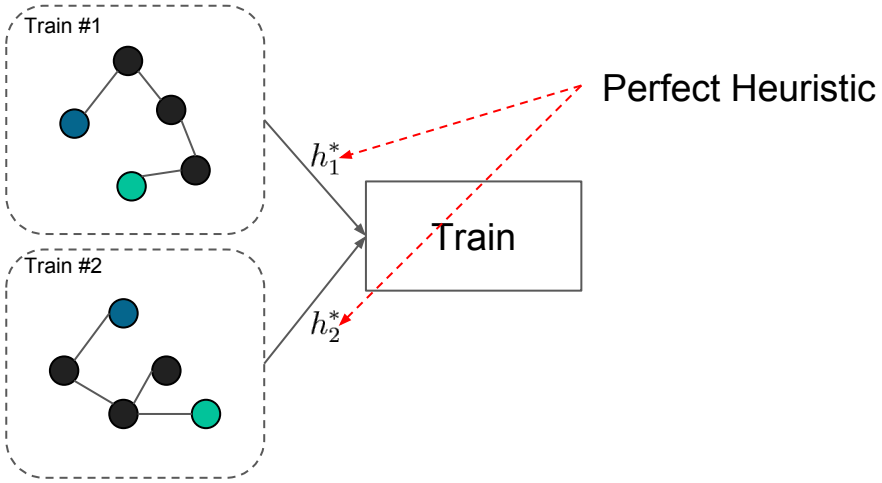
Training



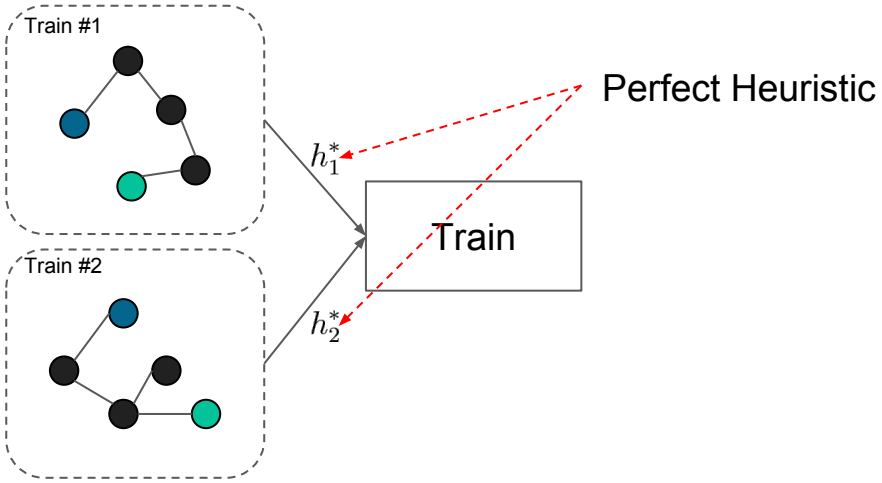
Training



Training



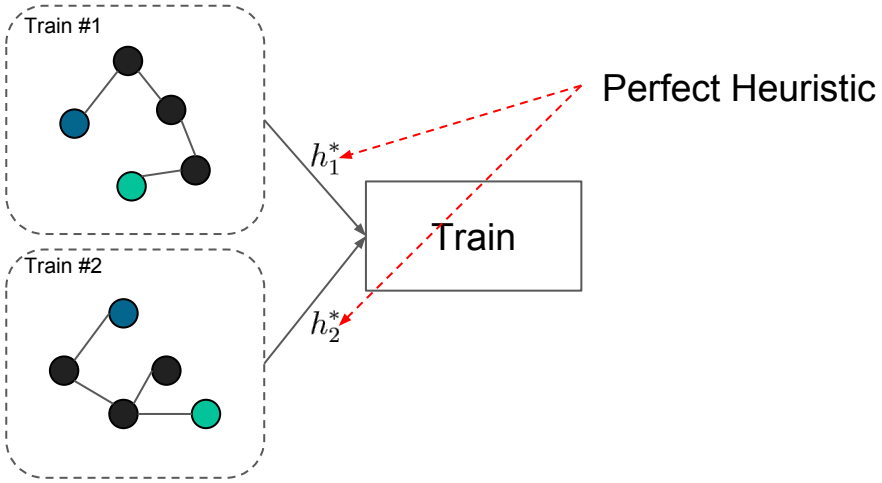
Training



$$h^*_{n,g} = |plan(n)|, \forall n \in g, \forall g \in G$$

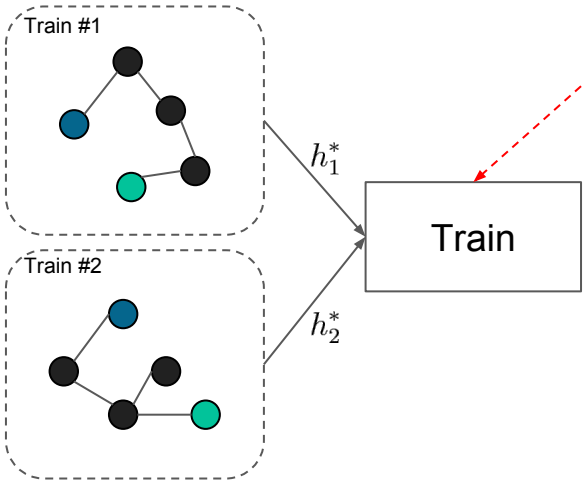
Training

Na prática:
1 task → múltiplos subgrafos



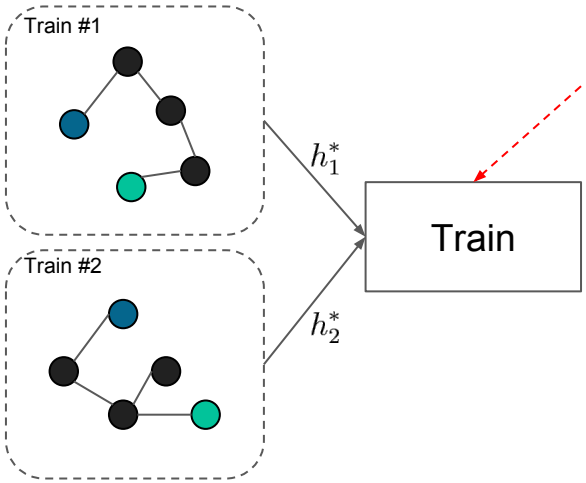
$$h^*_{n,g} = |plan(n)|, \forall n \in g, \forall g \in G$$

Training



$$L_h = \sum_{i=1}^{N} ||\hat{h}_i - h_i^*||_2$$

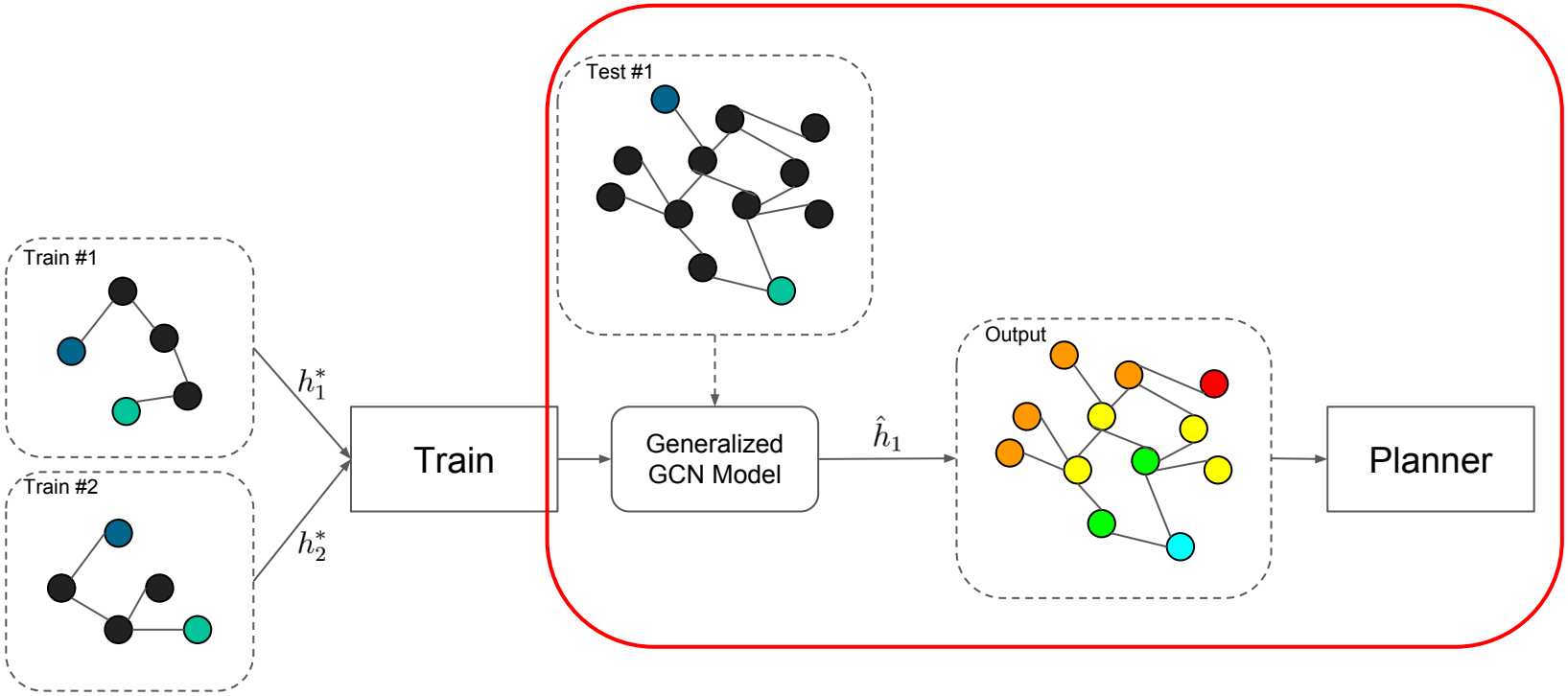
Training



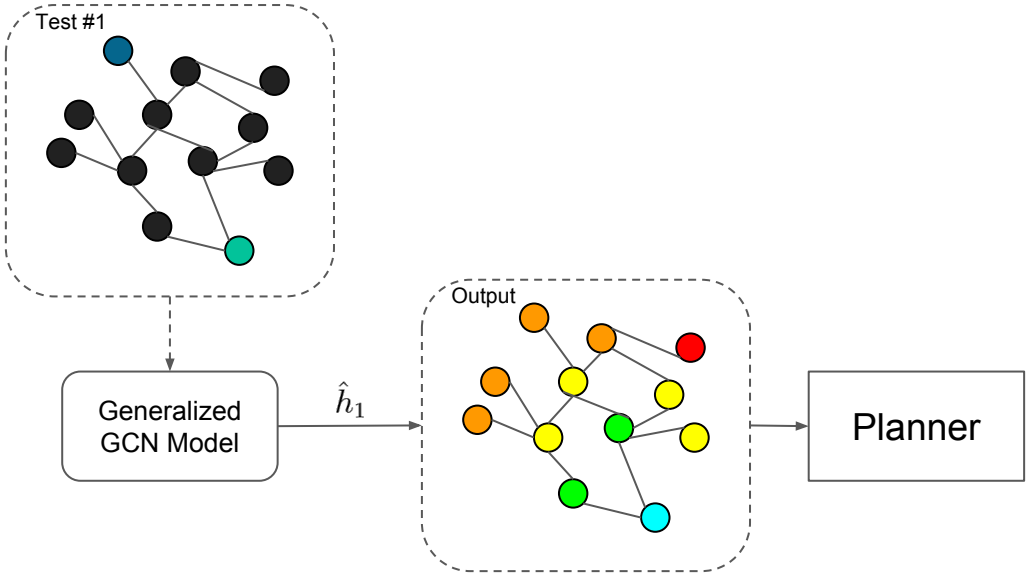
$$L_h = \sum_{i=} ||_i - h_i^*||_2$$

$$L_h(g) = \sum_{i=1}^{N} ||\hat{h}_{i,g} - h_{i,g}^*||_2 + \lambda||\Theta||_2$$

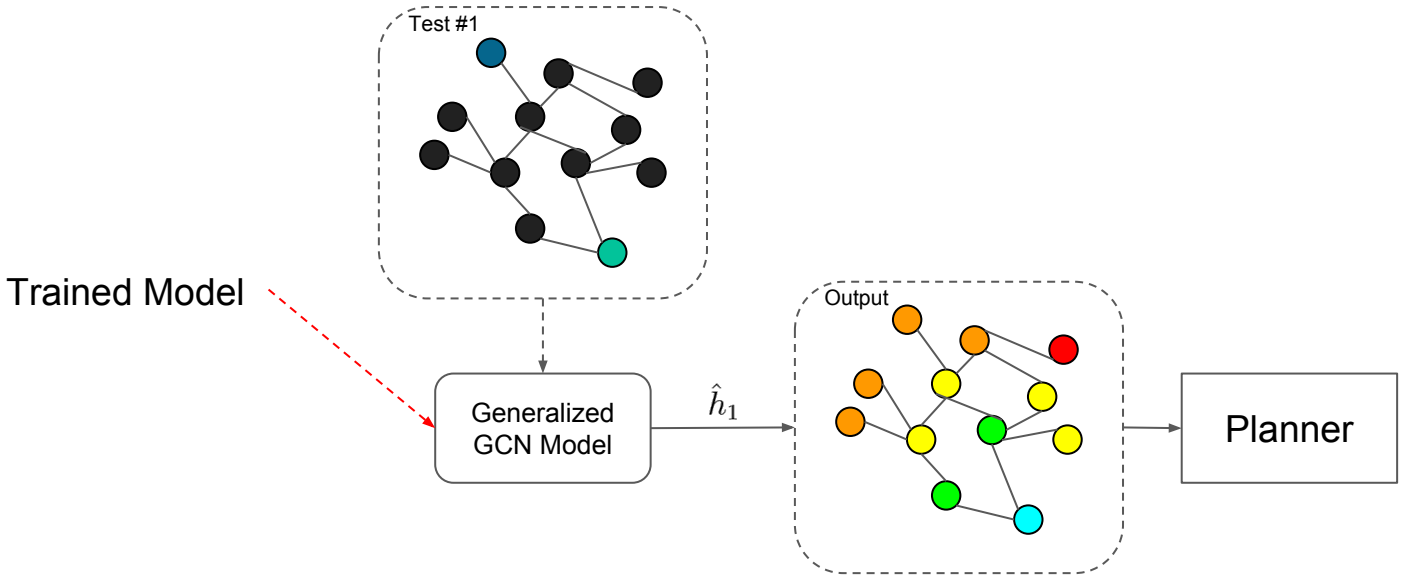
Testing


Test #1
Train #1
Train #2
h_1^*
h_2^*
Train
Generalized
GCN Model
$\hat{h}_1$
Output
Planner

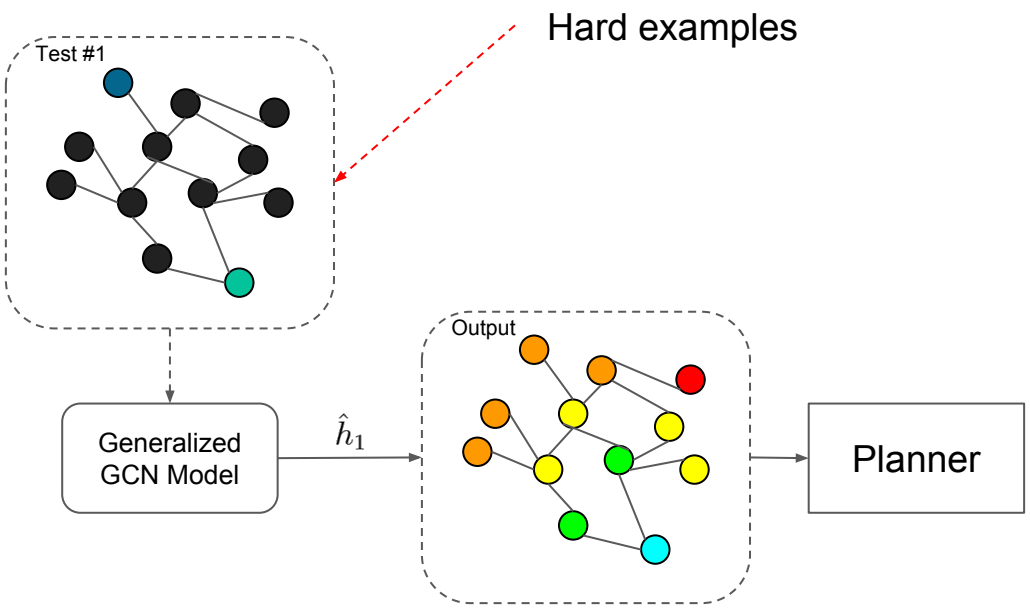
Testing



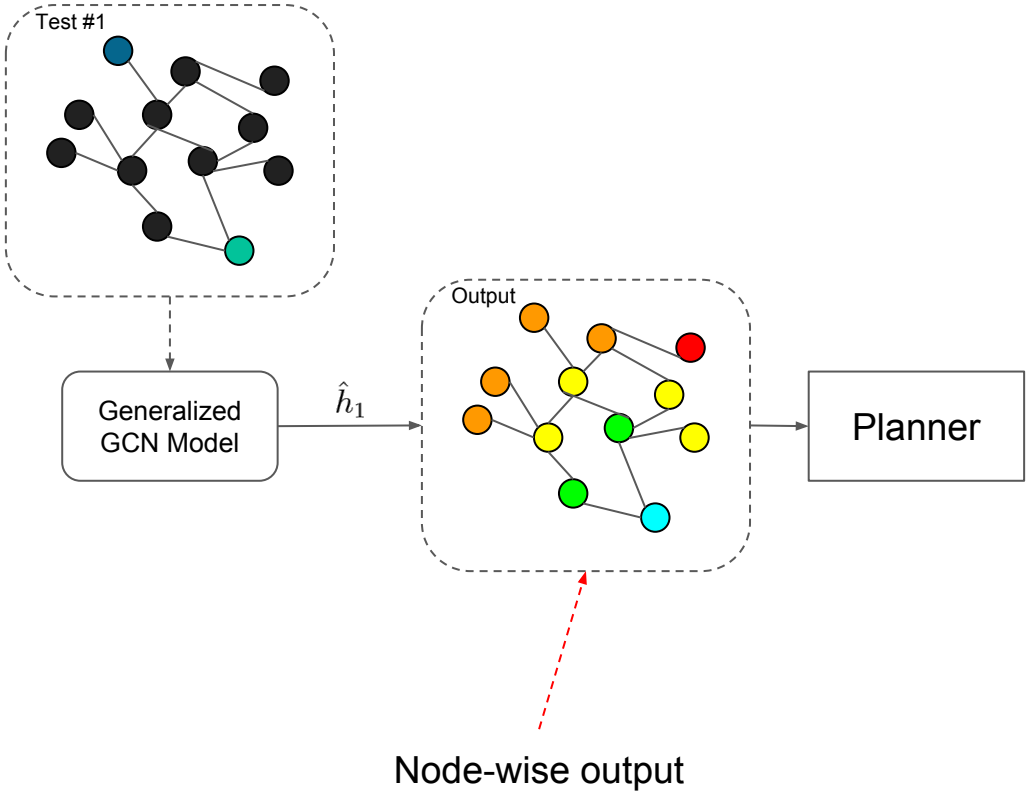
Testing


Test #1
Trained Model
Generalized
GCN Model
$\hat{h}_1$
Output
Planner

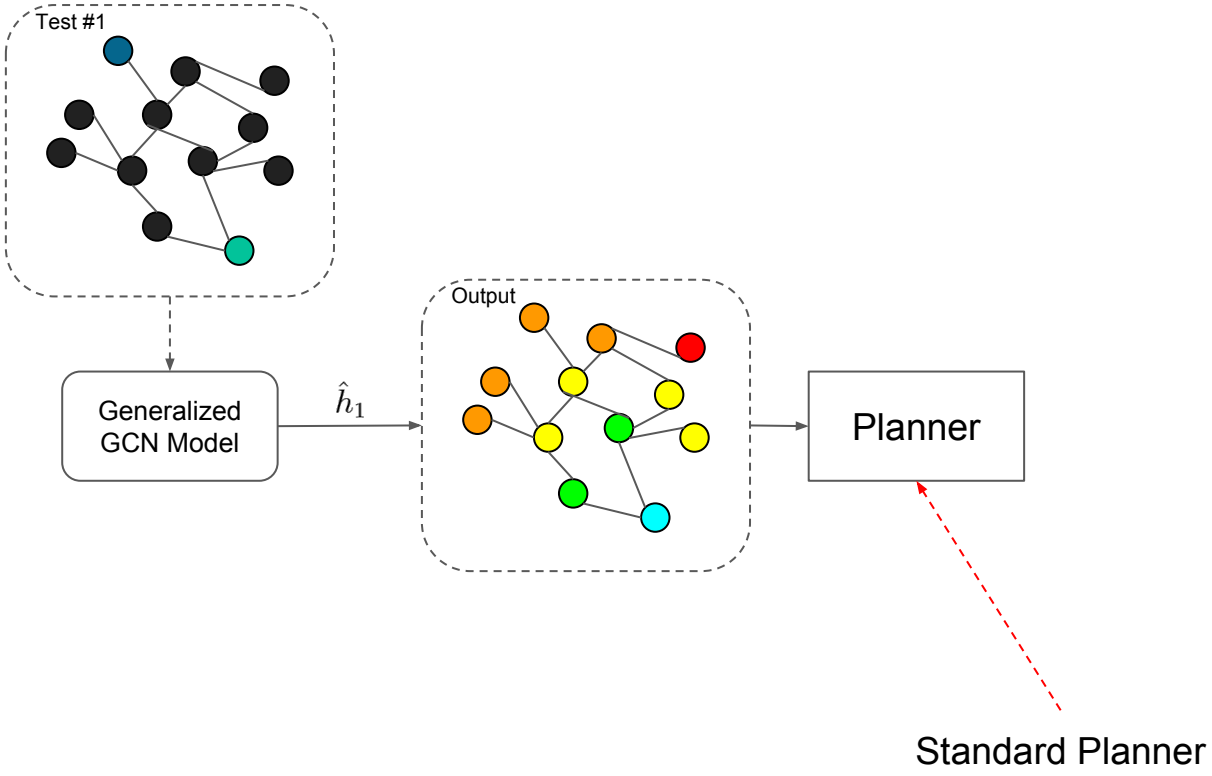
Testing


Hard examples
Test #1
Output
Generalized
GCN Model
$\hat{h}_1$
Planner

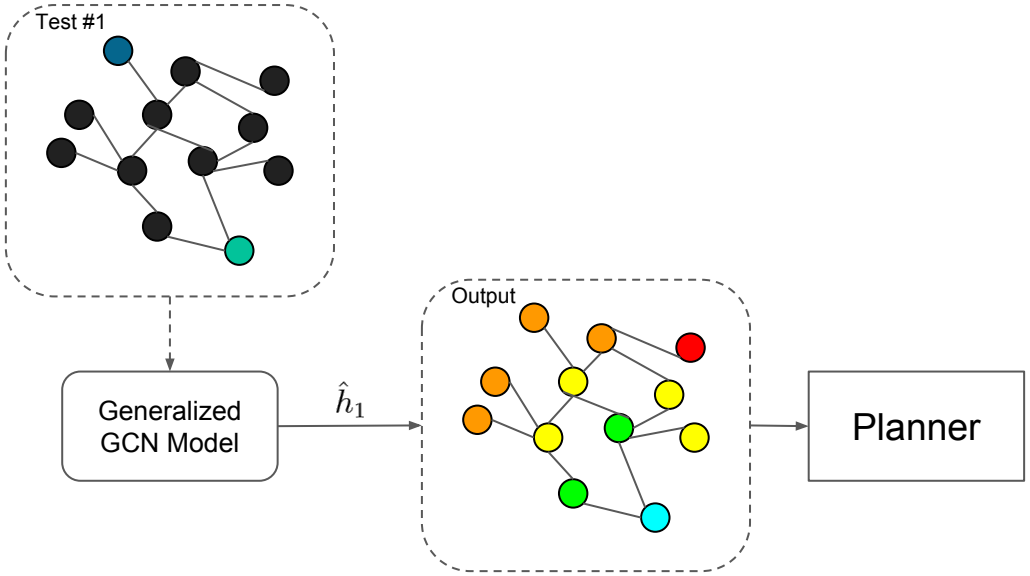
Testing


Test #1
Generalized
GCN Model
$\hat{h}_1$
Output
Planner
Node-wise output

Testing

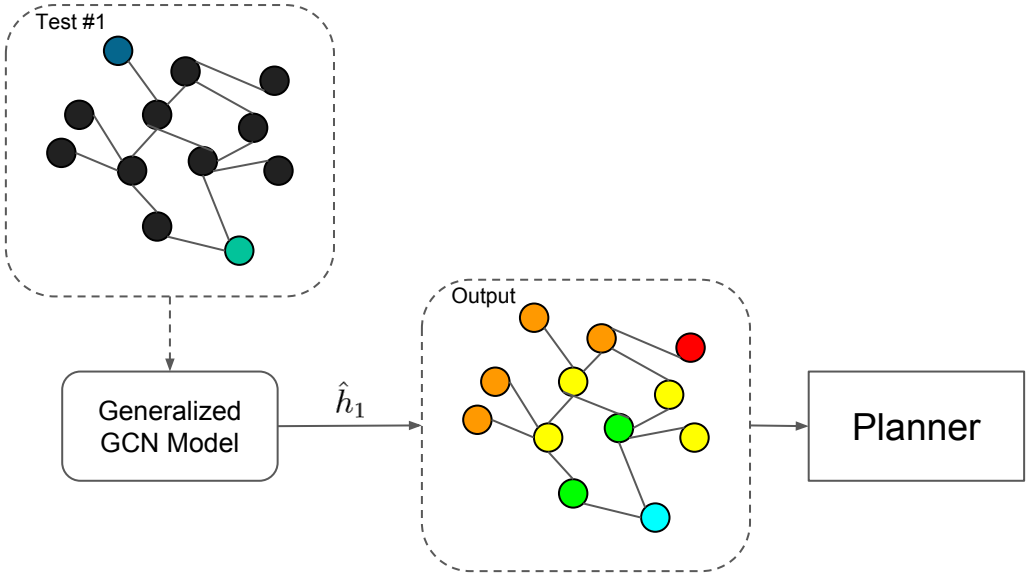

Test #1
Generalized
GCN Model
$\hat{h}_1$
Output
Planner
Standard Planner

Testing



Problema:
Input é um subgrafo.

Testing



Solução:
Expandir o nó para os seus vizinhos;
Vantagem do resultado servir para todos os nós expandidos.

Challenges

- Encode state information without exploding vector size;
- Classify the heuristic in planning definitions (Safe, Goal-aware, …)
- Understand heuristic generalization (becoming an off-the-shelf heuristic)

Challenges

- **Encode state information without exploding vector size;**
- Classify the heuristic in planning definitions (Safe, Goal-aware, …)
- Understand heuristic generalization (becoming an off-the-shelf heuristic)

Challenges

- Encode state information without exploding vector size;
- **Classify the heuristic in planning definitions (Safe, Goal-aware, …)**
- Understand heuristic generalization (becoming an off-the-shelf heuristic)

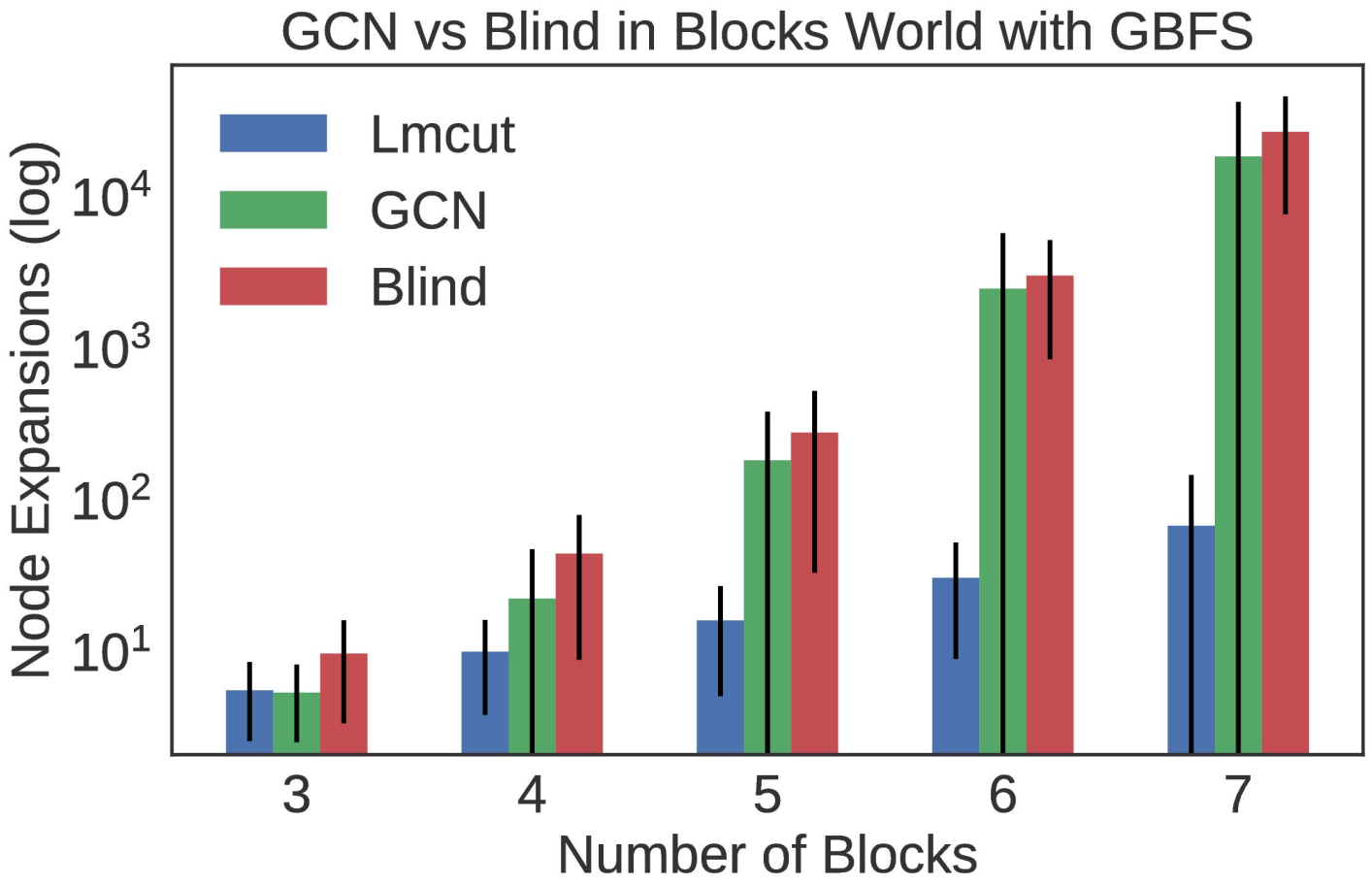
Challenges

- Encode state information without exploding vector size;
- Classify the heuristic in planning definitions (Safe, Goal-aware, …)
- **Understand heuristic generalization (becoming an off-the-shelf heuristic)**

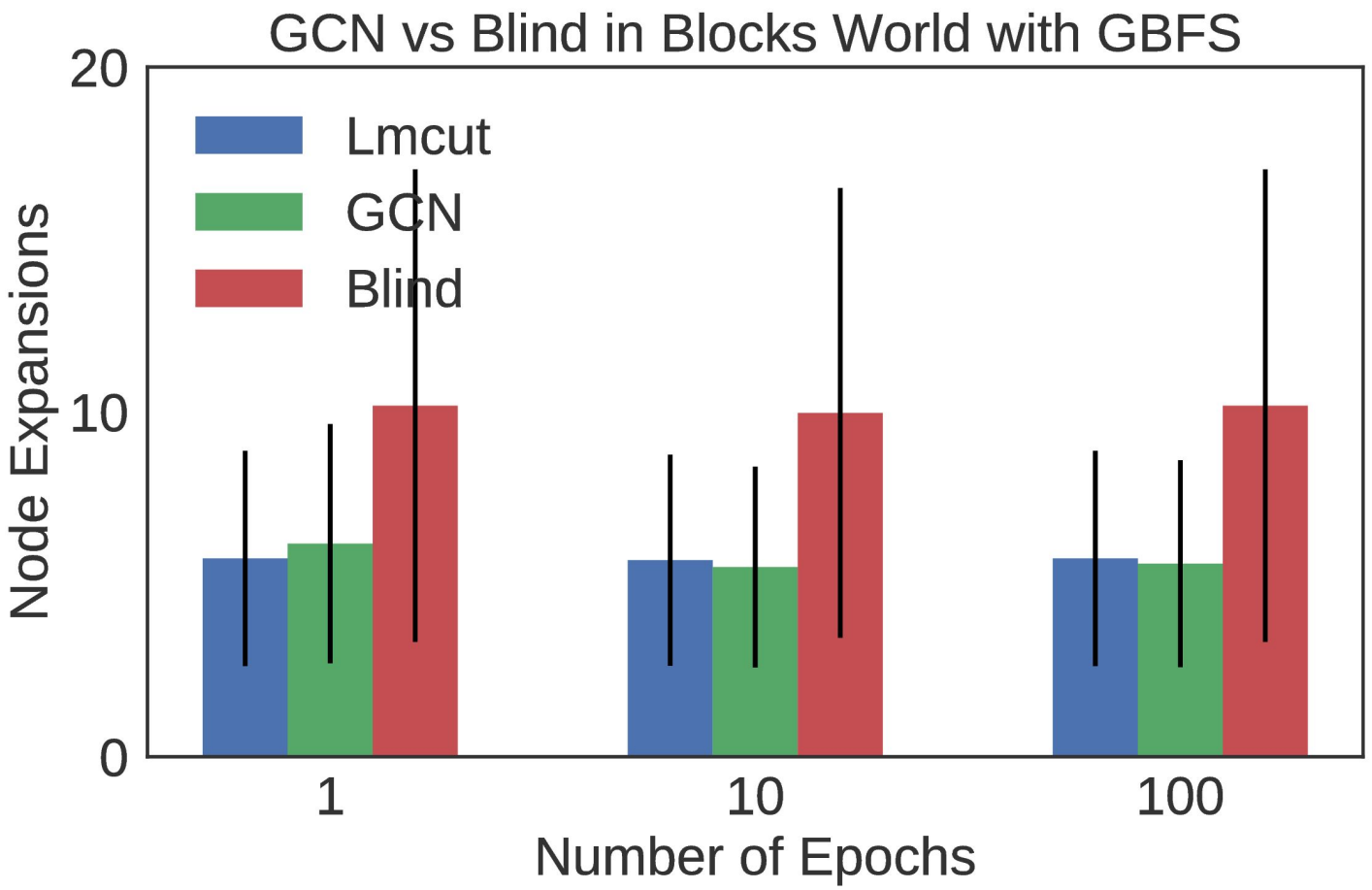
Results

- Number of nodes expanded by the heuristic;
- Poor implementation would lead to incorrect computational time comparison.

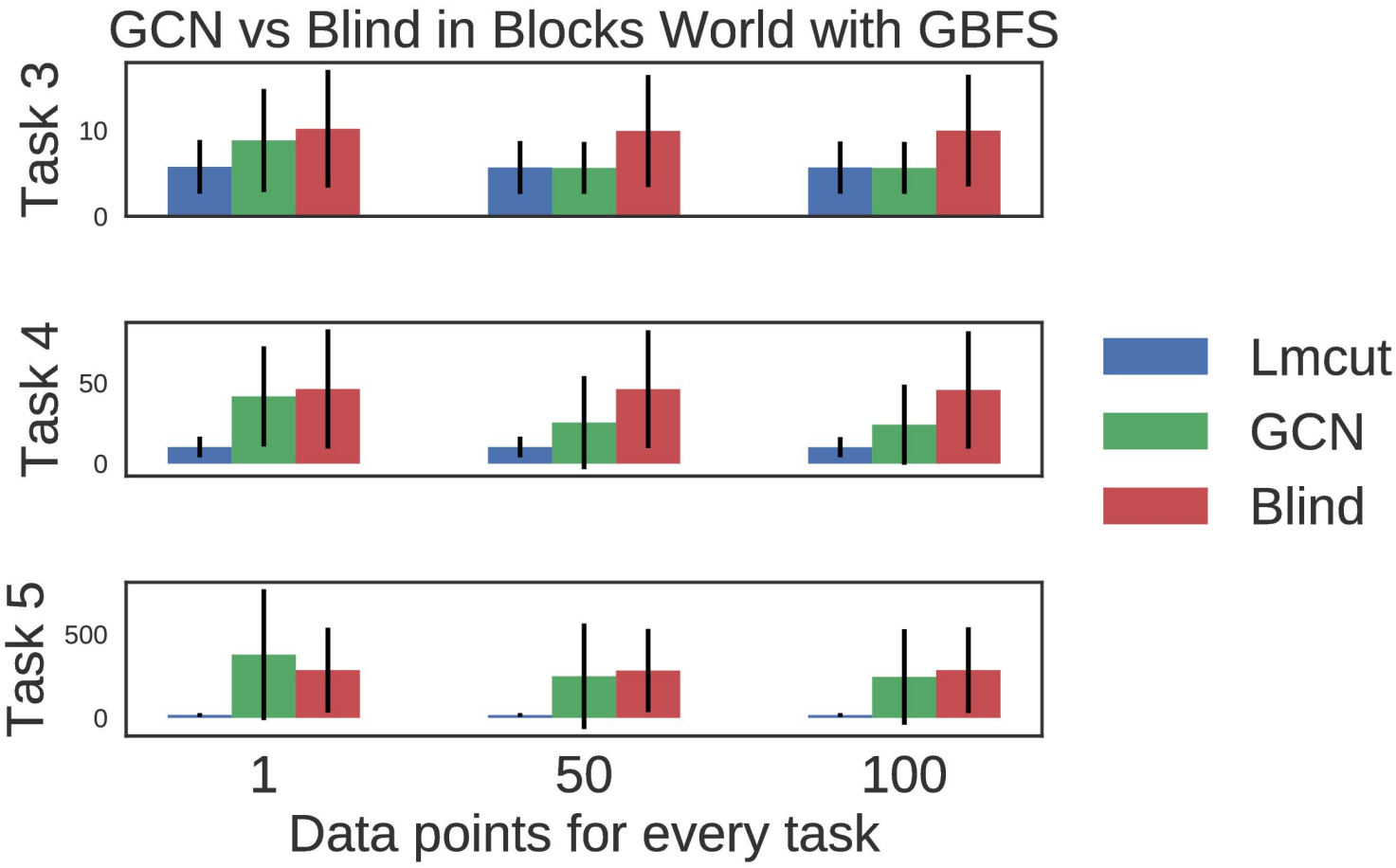
Results - Overall



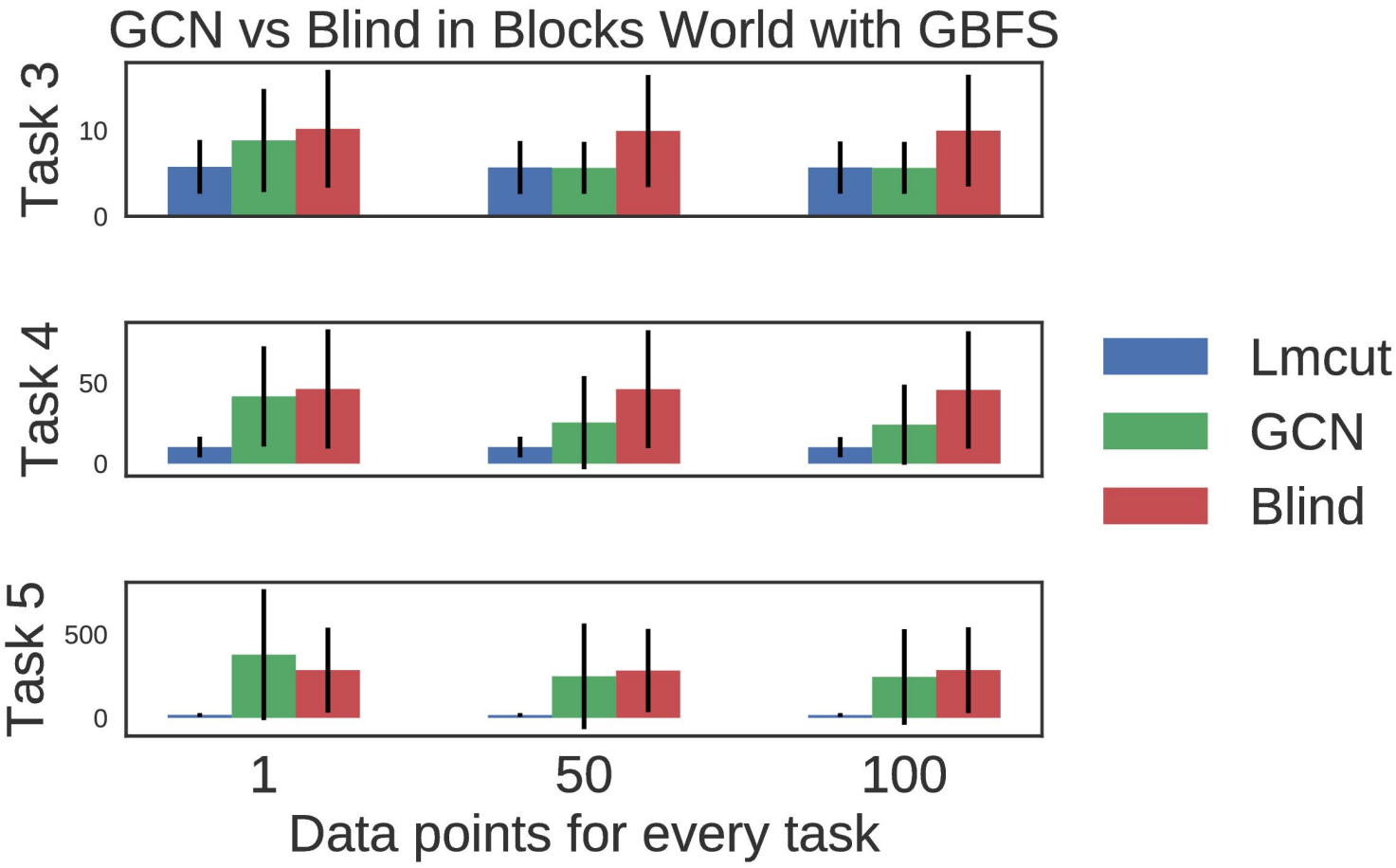
Results - Epochs



Results - Data Points



Results - Data Points



Results - Sokoban

Grid size	GCN	Blind	Lmcut
5	33.13 ± 5.72	208.85 ± 81.96	9.9 ± 1.5
6	864.57 ± 111.09	429.33 ± 208.83	12.4 ± 2.34