Learning Domain-Independent Heuristics over Hypergraphs

William Shen[1], Felipe Trevizan[2], Sylvie Thiébaux[2]

[1]william.w.y.shen@gmail.com [2]first.lastname@anu.edu.au

This poster was also presented at GenPlan20 at AAAI

Australian National University

Motivation and Overview

Existing approaches to learning heuristics:

- Use features derived from existing heuristics
- Learn domain-dependent heuristics
- Difficult to generalise across problems of different sizes

Our approach **STRIPS-HGN**:

- Learns heuristics completely from scratch
- Generalises across problems of different sizes
- Capable of learning domain-independent heuristics which generalise to domains they were not trained on

STRIPS-HGN approximates the shortest path over the *hypergraph* induced by the *delete-relaxation*.

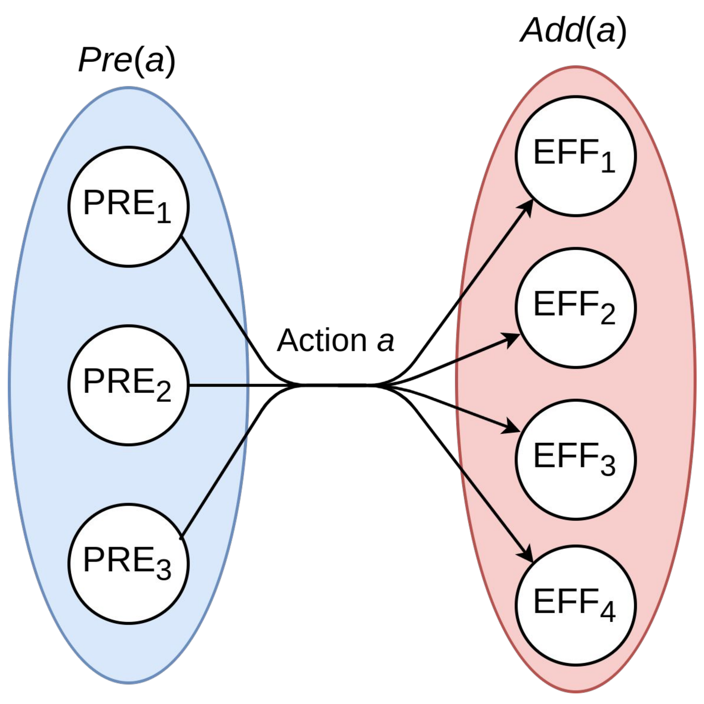
Delete-Relaxation Hypergraph

A **hypergraph** is a generalisation of a normal graph in which a **hyperedge** may connect any number of vertices together.

We consider the hypergraph induced by the **delete-relaxed problem** P^+

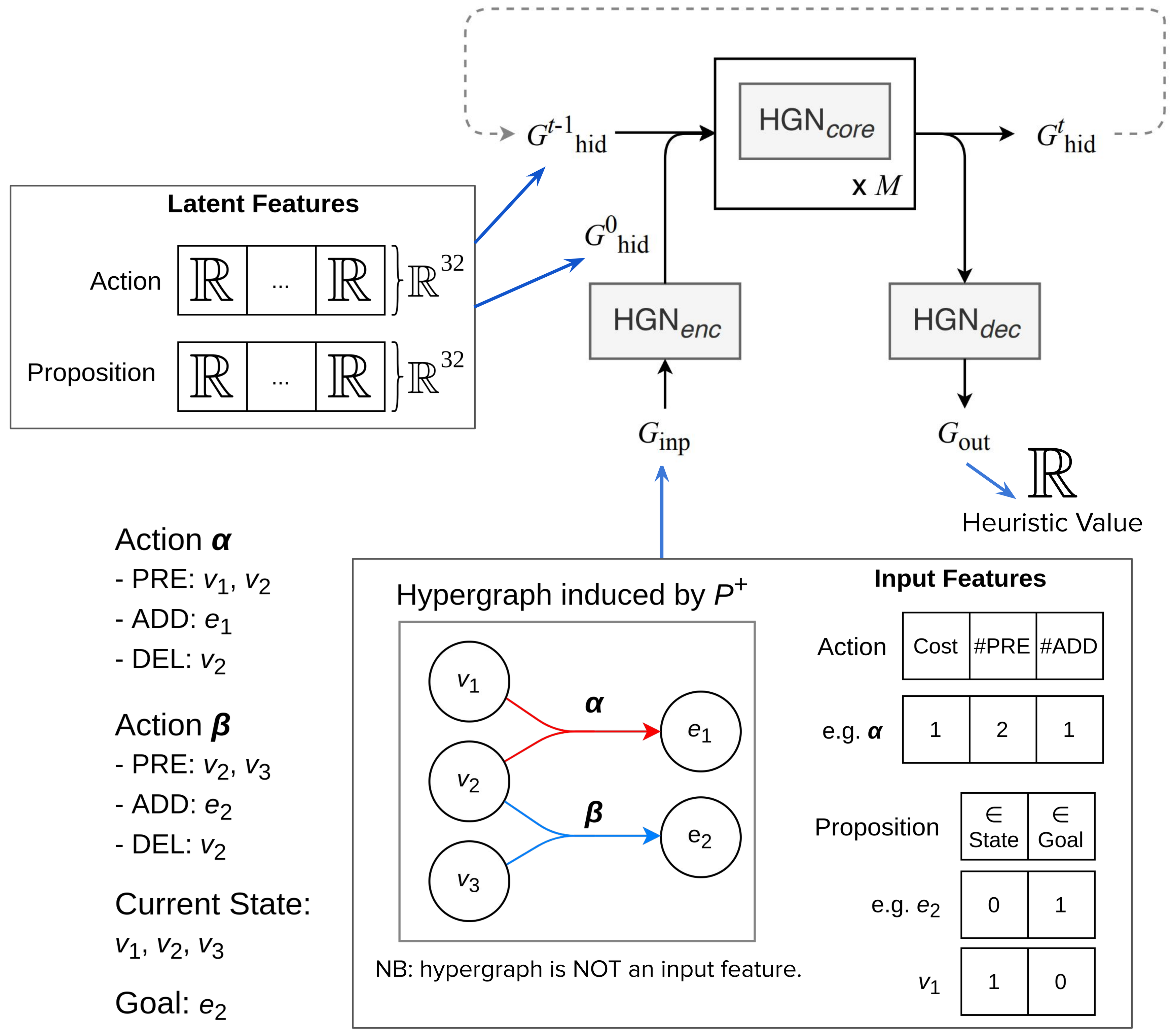
- **Delete-Relaxation**: Ignore negative effects of all actions
- Propositions are vertices
- Actions are hyperedges
  - Connect preconditions with positive effects

Used implicitly by h^{max} & h^{add}



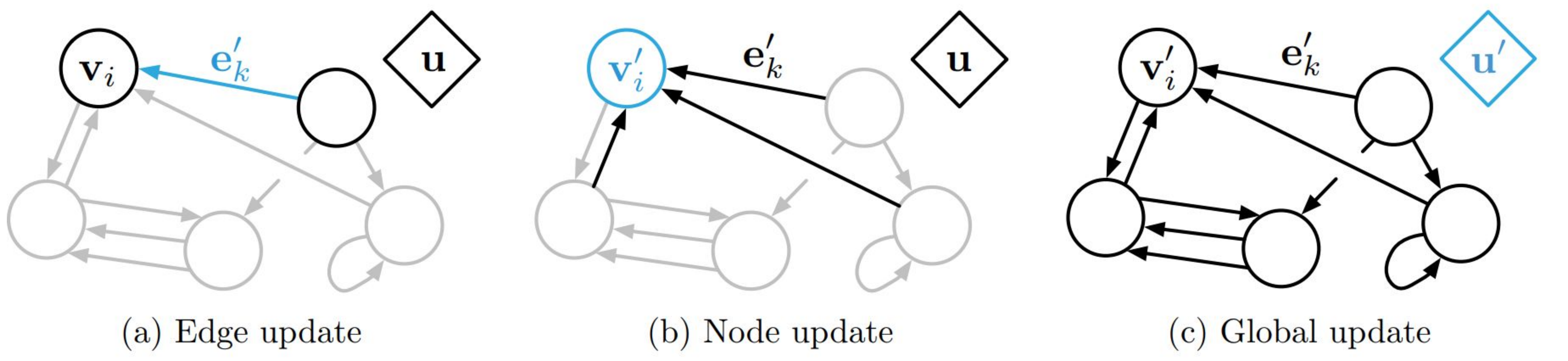
STRIPS-HGN

STRIPS-HGN uses a recurrent encode-process-decode architecture.



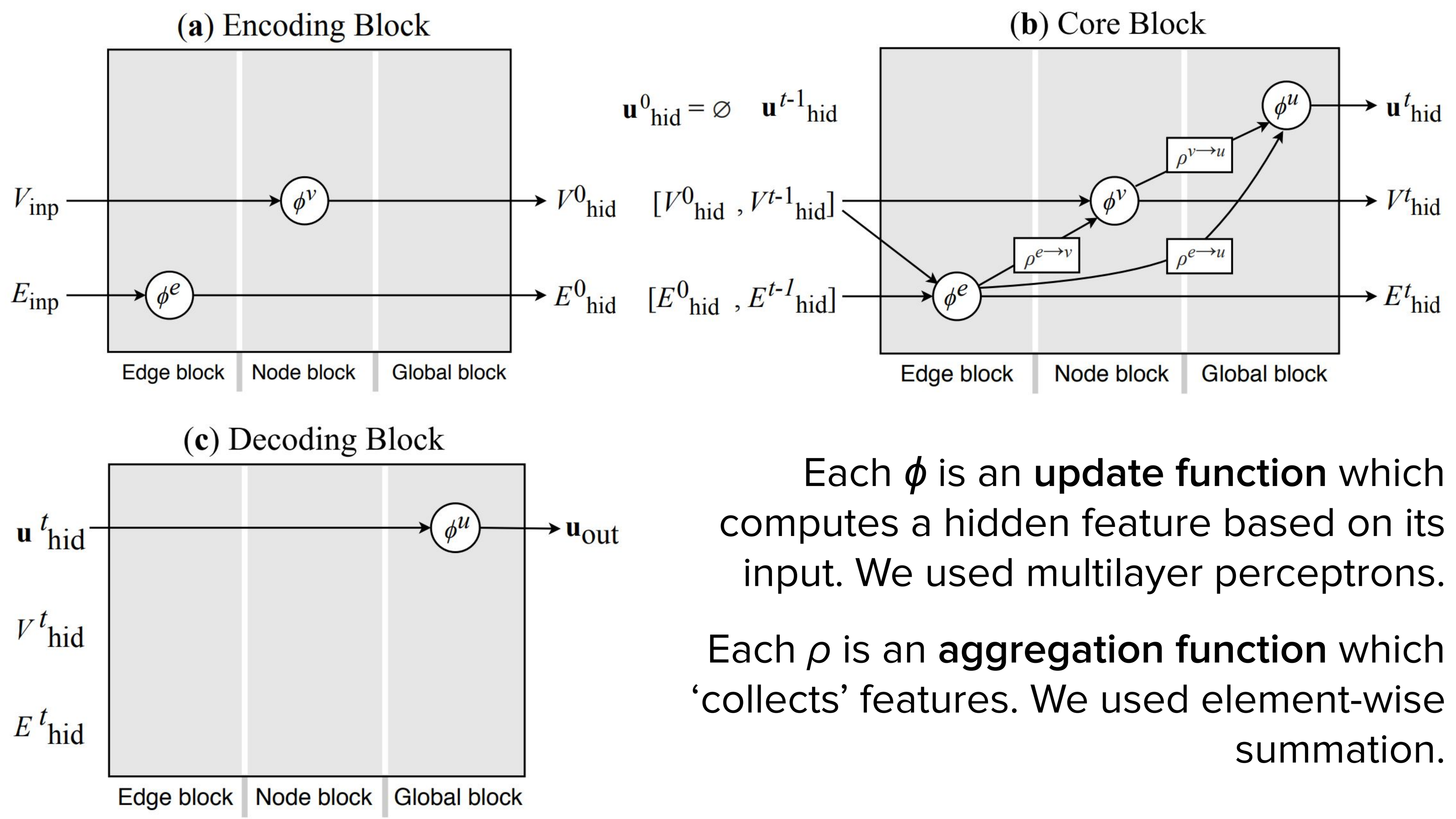
In each **core processing step**, we perform message passing:

- Update the latent features for each vertex/hyperedge
- Compute a global latent feature for the heuristic value
- Increased repetitions lead to deeper information propagation



[Battaglia et al., 2018]

STRIPS-HGN (cont.)



Each ϕ is an **update function** which computes a hidden feature based on its input. We used multilayer perceptrons.

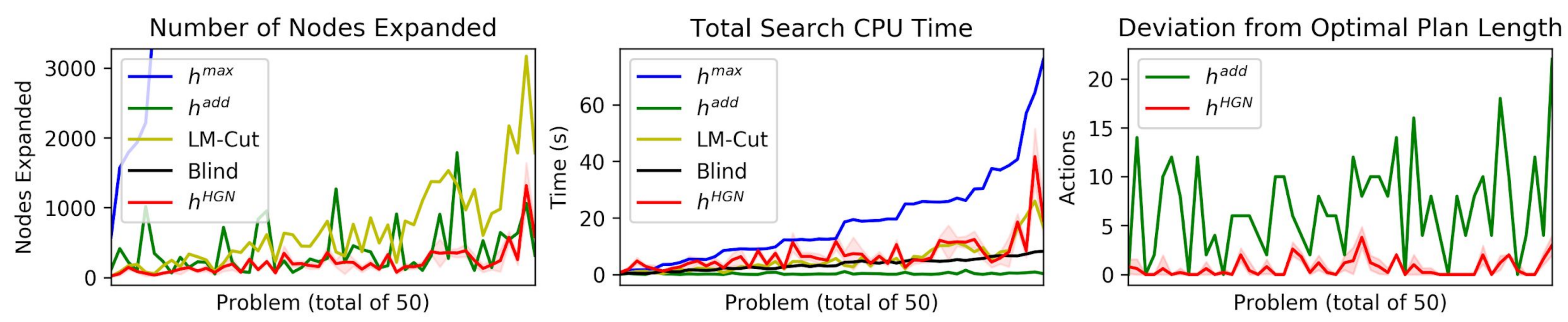
Each ρ is an **aggregation function** which 'collects' features. We used element-wise summation.

Training and Experiments

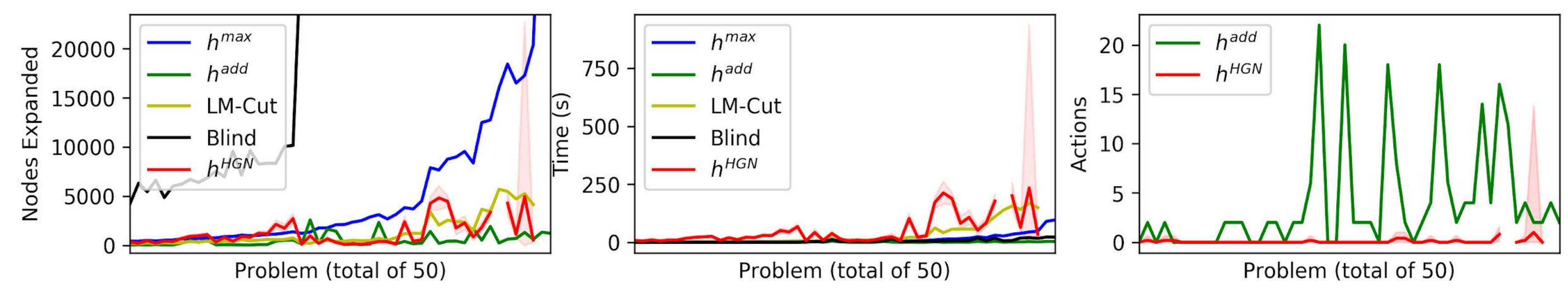
- **Training**: train on optimal heuristic values obtained from the optimal solution of the original problem (i.e., h^*)
  - Train on small problems, evaluate on larger problems
- **Baselines**: h^{add}, h^{max}, Landmark-cut and the blind heuristic
  - Have access to the same information (i.e., hypergraph)
- **Setting**: A* search with 5 minute timeout

Domain-Dependent Heuristics

8-puzzle: 10 training problems, 50 testing problems



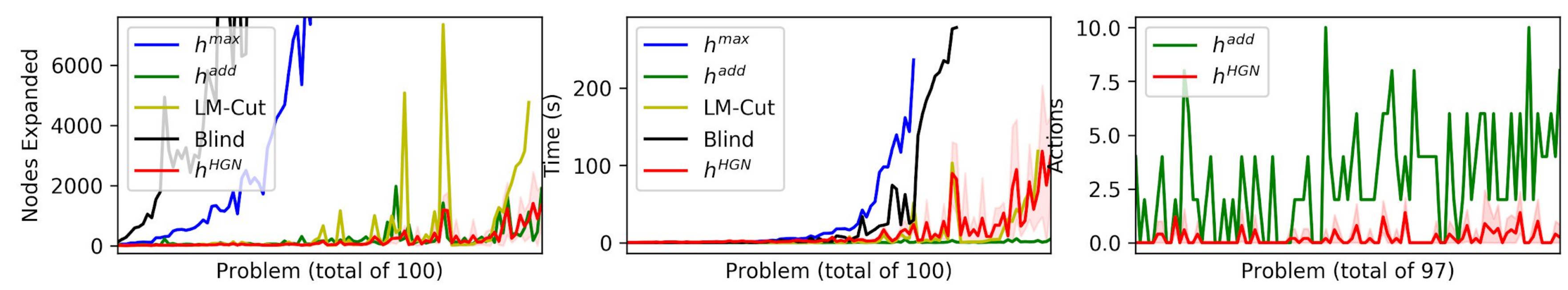
Sokoban: 20 train problems (grid size 5 & 7), 50 test problems (grid size 5, 7, 10)



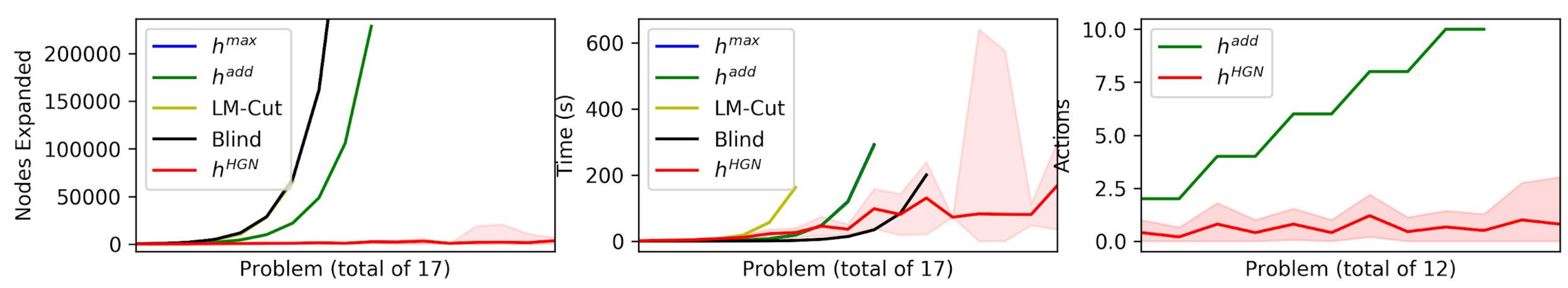
Multi-domain Heuristics

Train a single network on Blocksworld (10 problems with 4 to 5 blocks) + Zenotravel (10 small problems) + Gripper (3 problems with 1 to 3 balls)

Evaluate on **Blocksworld**: 100 test problems with 6 to 10 blocks



Evaluate on **Gripper**: 17 test problems with 4 to 20 balls



Domain-Independent Heuristics

Train a single network on Zenotravel (10 small problems) + Gripper (3 problems with 1 to 3 balls)

Evaluate on unseen **Blocksworld**: 50 test problems with 4 to 8 blocks