Learning Domain-Independent Heuristics for Grounded and Lifted Planning

Dillon Z. Chen[1,2] Sylvie Thiébaux[1,2] Felipe Trevizan[2]

[1]LAAS-CNRS, Université de Toulouse, [2]Australian National University

{dillon.chen, sylvie.thiebaux}@laas.fr, felipe.trevizan@anu.edu.au

Background

- **Classical planning**:
  - find a sequence of actions from an initial state to a goal state in a huge implicitly defined transition system
- **State of the art**:
  - at the core, heuristic search (A*, GBFS etc.)
- **Domain-independent heuristics**:
  - solve a relaxation of the planning task

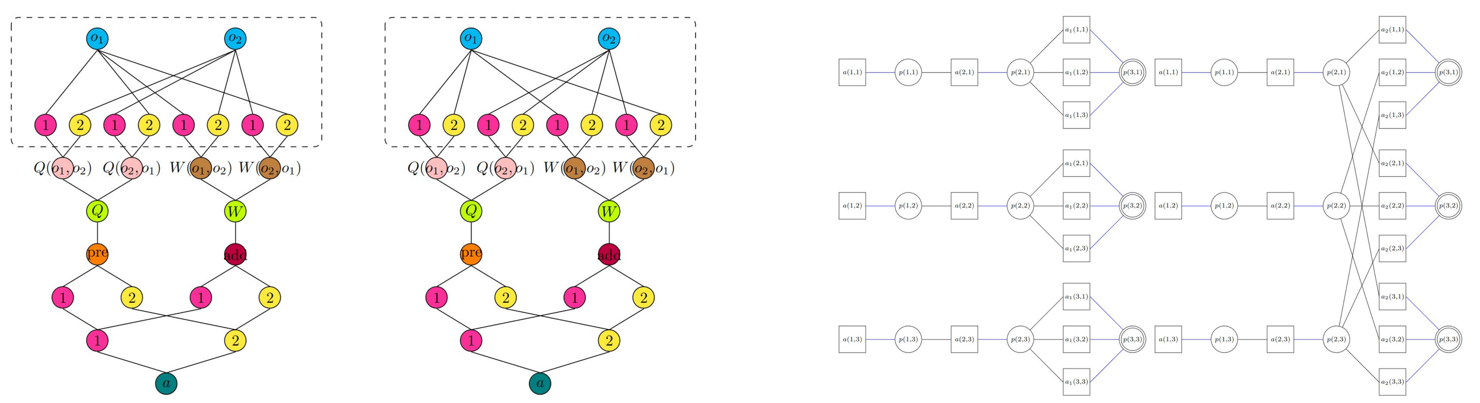
Contributions

1. first <u>graph representation of lifted planning tasks</u> for learning <u>domain-independent</u> heuristics
2. <u>theoretical expressivity results</u> for learning domain-independent heuristics
3. <u>large scale training</u> of domain-independent heuristics on IPC dataset, consisting of 30000 states

Theoretical Expressivity Results

GNNs

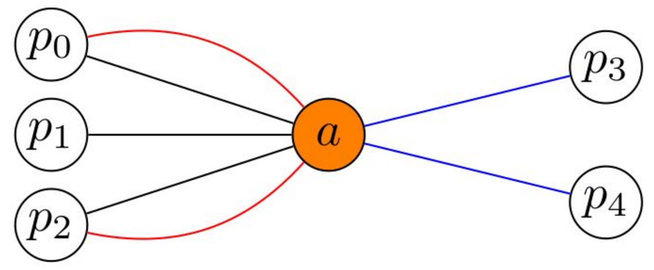
+ can learn h^{add}/h^{max} on grounded graphs [Thm. 4.1]
  - Pf: encode VI into GNNs + universal approximation theorem
- cannot learn h^{add}/h^{max} on lifted graphs [Thm. 4.3]
- cannot learn h^+ and h^* [Thm. 4.4]
- cannot learn an approximation of h^+ and h^* [Thm. 4.5]
  - Pf: counterexample tasks
- *note:* these are worst case scenarios; it is possible to learn h^+ or h^* on subclasses of planning problems



Domain-Independent Graphs for Planning Tasks

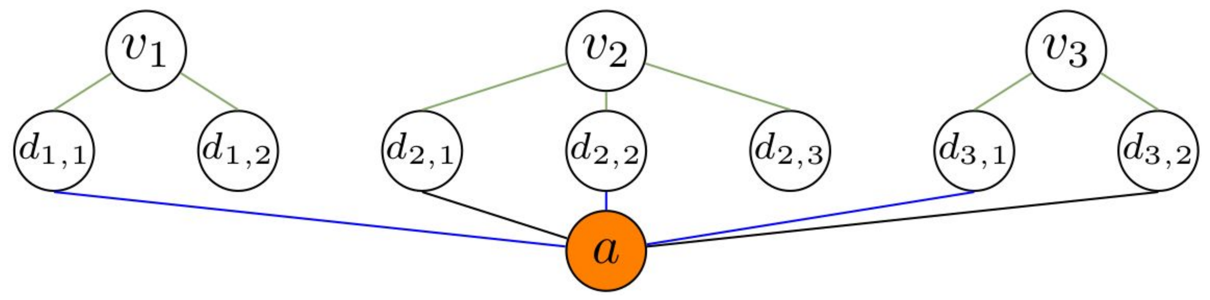
STRIPS Learning Graph (SLG)

- $\langle P, A, s_0, G\rangle$



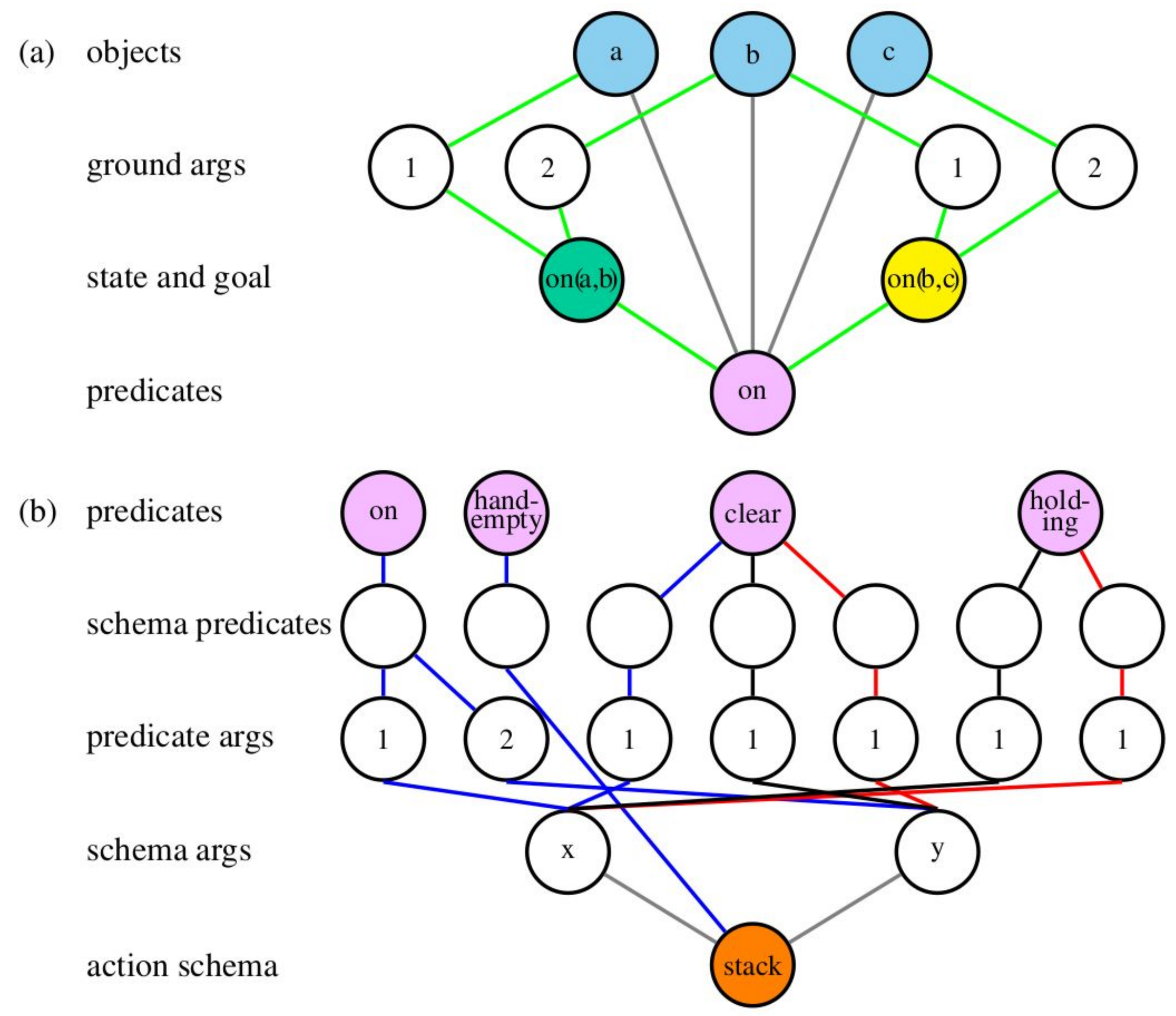
FDR Learning Graph (FLG)

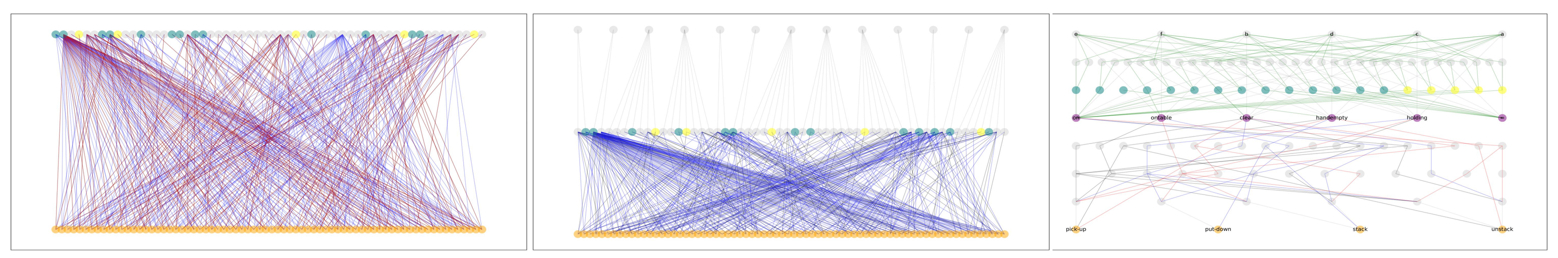
- $\langle \mathcal{V}, A, s_0, s_\star\rangle$



Lifted Learning Graph (LLG)

- $\langle \mathcal{P}, \mathcal{O}, \mathcal{A}, s_0, G\rangle$





Grounded vs Lifted Graphs

Grounded graphs: SLG, FLG

+ more informative for domain-independent learning
- large and slow to construct and evaluate
- requires grounded representation of planning tasks

Lifted graphs: LLG

+ small and quick to evaluate
+ can be used with planners which do not ground
- less informative for domain-independent learning

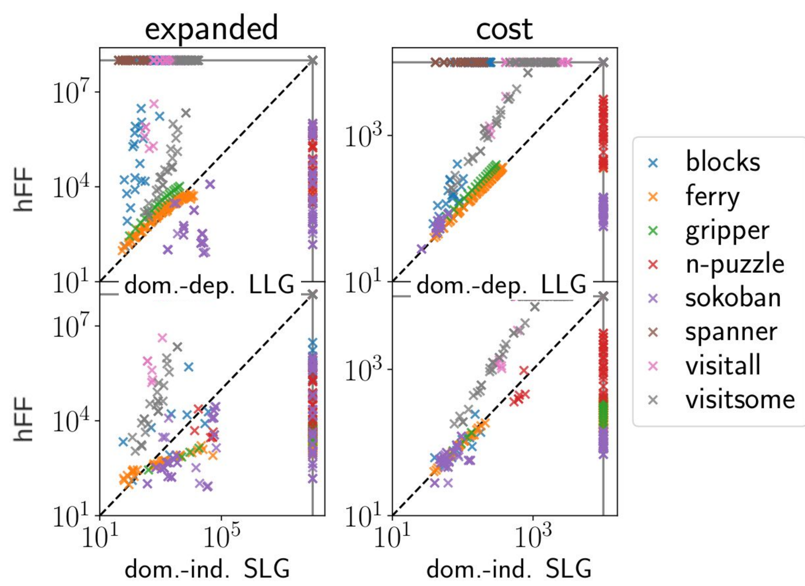
Search Guidance Performance

Training setting: given a planning domain δ

- **domain-dependent (DD)**: train on small tasks from δ
- **domain-independent (DI)**: train on tasks not from δ

Testing setting:

- eager GBFS, GPUs w/ batch evaluation, 600s timeout
- 8 message passing layers, mean aggregator



	baselines			domain-dep.			domain-ind.		
	blind	h^{FF}	HGN	SLG	FLG	LLG	SLG	FLG	LLG
blocks (90)	-	19	-	-	6	**62**	9	8	6
ferry (90)	-	**90**	-	32	33	88	28	22	2
gripper (18)	1	**18**	5	9	6	**18**	5	3	9
n-puzzle (50)	-	**36**	-	10	10	-	6	3	-
sokoban (90)	74	**90**	10	31	29	34	45	40	15
spanner (90)	-	-	-	-	-	**60**	-	-	-
visitall (90)	-	6	25	46	**50**	44	16	41	-
visitsome (90)	3	26	33	72	39	65	**73**	65	15

Effect of GNN hyperparameters

- mean > max aggregator for DI training
  - **contrary to literature of GNNs for CO**
- more layers ~ worse performance
  - GNN oversmoothing
  - slower evaluation

		domain-dep.			domain-ind.		
aggr.	L	SLG	FLG	LLG	SLG	FLG	LLG
mean	4	0.40	0.43	0.94	0.19	0.15	0.18
	8	**0.53**	0.40	**1.00**	**0.38**	0.32	0.33
	12	0.44	0.37	0.85	0.37	0.32	0.21
	16	0.31	0.18	0.75	0.36	**0.32**	0.12
max	4	0.46	**0.50**	0.89	0.33	0.29	0.30
	8	0.41	0.43	0.88	0.36	0.30	**0.52**
	12	0.36	0.43	0.80	0.12	0.24	0.39
	16	0.41	0.36	0.53	0.06	0.24	0.20