Graph Neural Networks and Graph Kernels For Learning Heuristics: Is there a difference?

Dillon Z. Chen[1,2] **Felipe Trevizan**[2] **Sylvie Thiébaux**[1,2]
[1]LAAS-CNRS, [2]Australian National University

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
- **Domain-independent heuristics**:
  - solve a relaxation of the planning task
- **Learning heuristics**:
  - use an ML model to learn heuristics from training data, usually in a domain-dependent fashion
  - recent works focused on deep learning methods
  - learners still not competitive with classical planners

Contributions

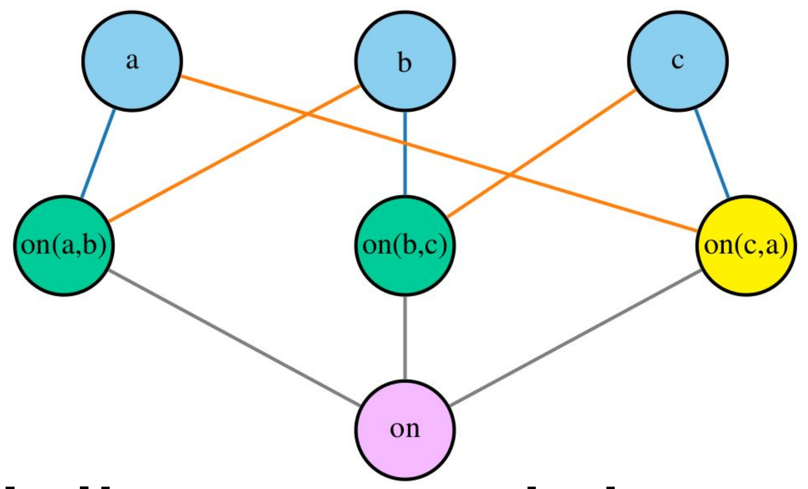
- novel statistical machine learning method for learning heuristics
- state of the art results for learning heuristics for planning
- discussion and experimental comparison between GNNs and GKs for learning heuristics

Future Work

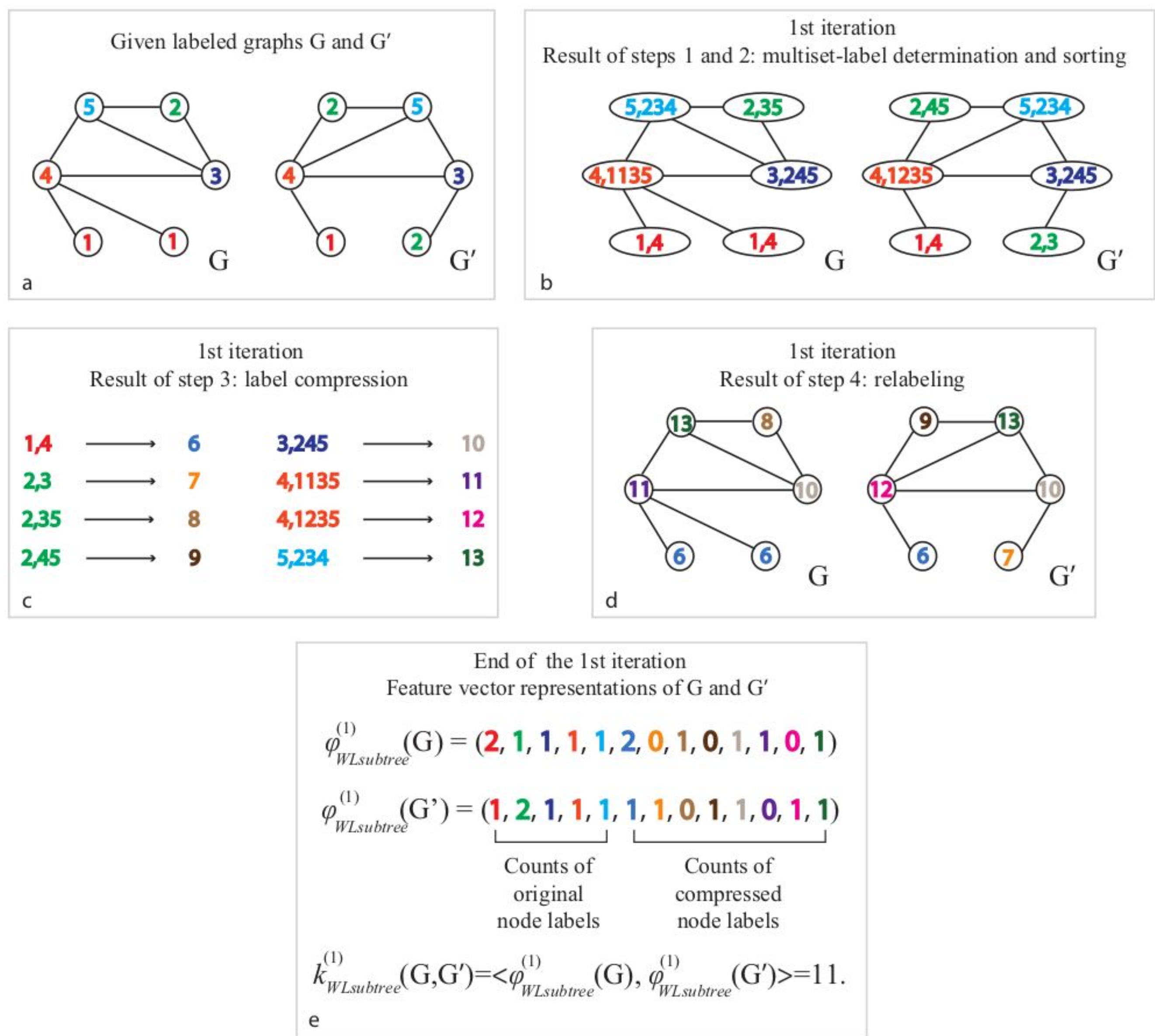
- theoretical connections with description logic features [Martin and Geffner, Appl. Intell. 04] and previous GNN networks [Ståhlberg and Bonet and Geffner, ICAPS 22]
- explore statistical machine learning methods (kernelised and/or Bayesian linear regression … Gaussian Processes)
- learn robust heuristics or policies with WL features

Graph Kernels for Learning Heuristics

Instance Learning Graph (ILG)



WL linear graph kernel



(figure from [Shervashidze et al., JMLR 2011])

- based on the WL algorithm for approximating the graph isomorphism problem

Algorithm 1: WL algorithm

Data: A graph $G = \langle V, E, c \rangle$, hash function f, and number of WL iterations h.
Result: Multiset of colours.

$c^{(0)}(v) \leftarrow c(v), \quad \forall v \in V$
for $j = 1, \ldots, h$ **do** $c^{(j)}(v) \leftarrow f\left(c^{(j-1)}(v), \{\!\{c^{(j-1)}(u) \mid u \in \mathcal{N}(v)\}\!\}\right), \quad \forall v \in V$;
return $\bigcup_{j=0,\ldots,h} \{\!\{c^{(j)}(v) \mid v \in V\}\!\}$

Search Guidance Performance

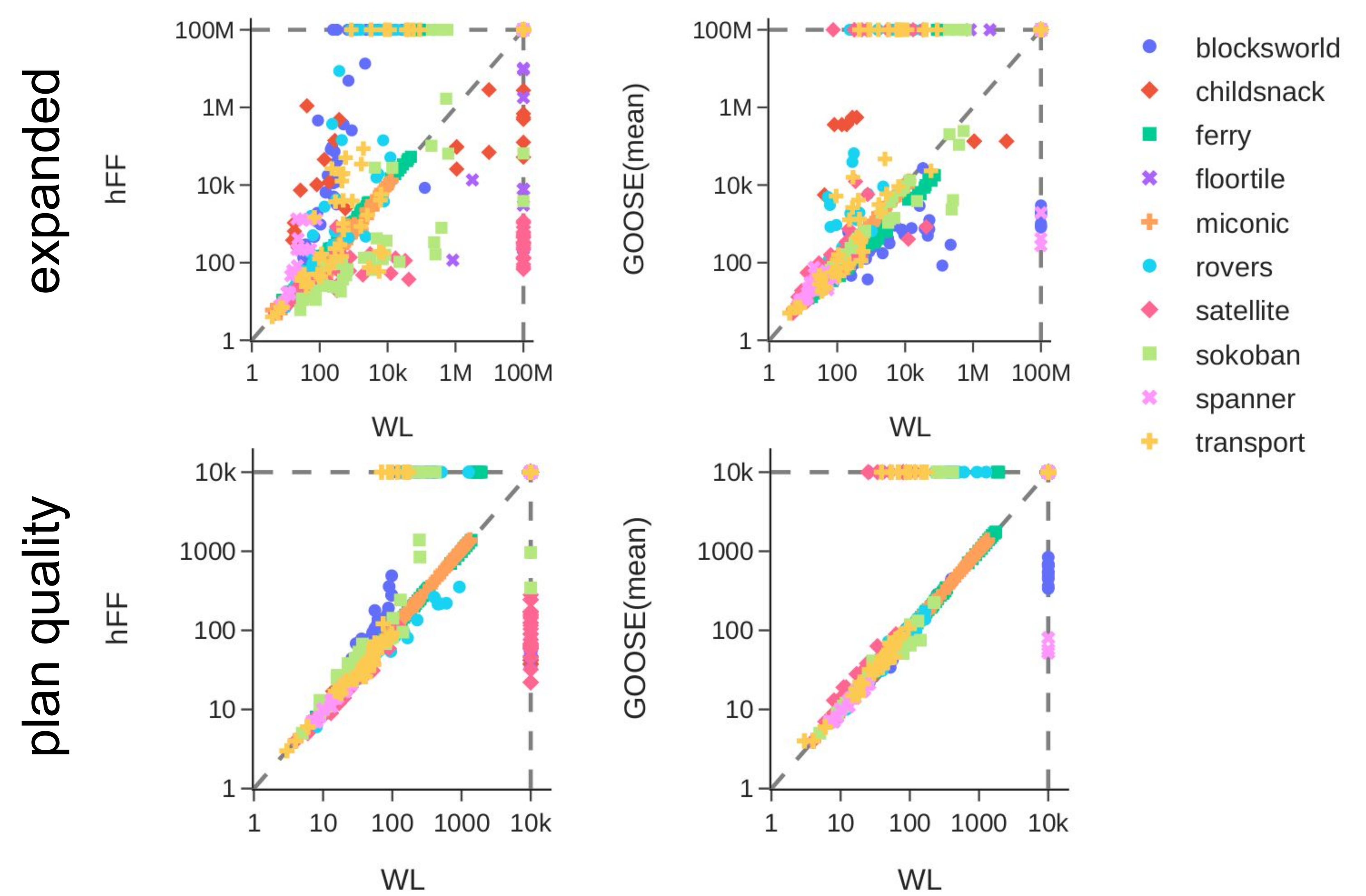
- **Benchmarks**: IPC 2023 Learning Track setup
- **Baselines:** blind search, h^{FF}, Muninn (IPC 2023 competitor), GNN-GOOSE
- **New model**: WL-GOOSE
- **Setup**: 8Gb memory, 30 minutes timeout

coverage

				GOOSE		
Domain	blind	h^{FF}	Muninn	GNN(max)	GNN(mean)	**WL**
blocksworld	8	28	40	49	**58**	49
childsnack	9	**26**	11	19	20	20
ferry	10	68	46	64	72	**74**
floortile	2	**12**	-	-	-	2
miconic	30	**90**	30	**90**	**90**	**90**
rovers	15	34	15	25	29	**45**
satellite	12	**65**	18	31	29	37
sokoban	27	36	26	32	33	**37**
spanner	30	30	32	30	**33**	30
transport	9	41	17	38	35	**49**
sum	152	430	235	378	399	**433**

competition score

				GOOSE		
Domain	blind	h^{FF}	Muninn	GNN(max)	GNN(mean)	**WL**
blocksworld	8.0	14.1	40.0	45.6	**52.7**	44.4
childsnack	9.0	**20.1**	11.0	17.6	19.9	18.9
ferry	10.0	67.6	46.0	63.9	71.9	**73.6**
floortile	2.0	**11.2**	-	-	-	1.8
miconic	30.0	88.5	30.0	**89.2**	89.1	89.0
rovers	15.0	32.7	14.2	19.7	24.8	**36.4**
satellite	12.0	**63.8**	18.0	24.6	21.6	32.9
sokoban	27.0	26.3	24.3	28.3	30.0	**33.1**
spanner	30.0	30.0	32.0	30.0	**33.0**	27.6
transport	9.0	39.3	17.0	35.6	31.6	**46.3**
sum	152.0	393.5	232.4	354.5	374.6	**404.0**



(top left triangles = better for WL-GOOSE)