Return to Tradition: Learning Reliable Heuristics with Classical Machine Learning

Dillon Z. Chen Felipe Trevizan Sylvie Thiébaux

Australian National University

LAAS CNRS

Tuples TRUSTWORTHY AI

icaps

Problem Statement: Learning for Planning

1. learn from **few** number of **small** problems
2. evaluate on **larger size** problems
3. do not use any IPC planner during evaluation

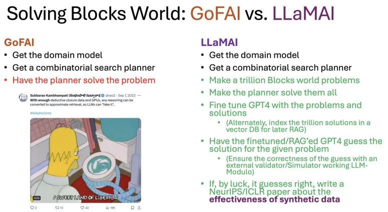


Image credits: Subbarao Kambhampati

✓ ok:

- learn heuristic
- learn policy

✗ not ok:

- learn transformation + LAMA
- learn $\emptyset$ + LAMA
- learn portfolios of IPC planners
- train and test on Blocksworld instances with 10 blocks

Question Time

Is deep learning the future for scaling up PDDL planning?

Related work 1/2. Deep Learning

- ✗ not explainable/interpretable
- ✗ data and computationally intensive; "profoundly uneconomical"[1]
- ✗ limited expressivity[2,3]
- ✗ evaluated on trivial problems, results for hard problems hidden
- ✗ **falls majorly behind classical planners**

---

[1] Michael Katz et al. *Planning with Language Models Through The Lens of Efficiency*. 2024. arXiv: 2404.11833.

[2] Simon Ståhlberg, Blai Bonet, and Hector Geffner. "Learning General Optimal Policies with Graph Neural Networks: Expressive Power, Transparency, and Limits". In: *ICAPS*. 2022.

[3] Dillon Ze Chen, Sylvie Thiébaux, and Felipe Trevizan. "Learning Domain-Independent Heuristics for Grounded and Lifted Planning". In: *AAAI*. 2024.

Related work 2/2. Symbolic Approaches

Description Logic Features[1] + policy rules[2]/sketches[3]

- $\uparrow$ robust
- $\uparrow$ explainable and interpretable
- $\downarrow$ expressivity restricted by (compilation to) binary predicates

---

[1] Mario Martín and Hector Geffner. "Learning Generalized Policies from Planning Examples Using Concept Languages". In: *Appl. Intell.* (2004).

[2] Guillem Francès, Blai Bonet, and Hector Geffner. "Learning General Planning Policies from Small Examples Without Supervision". In: *AAAI*. 2021.

[3] Dominik Drexler, Jendrik Seipp, and Hector Geffner. "Learning sketches for decomposing planning problems into subproblems of bounded width". In: *ICAPS*. 2022.

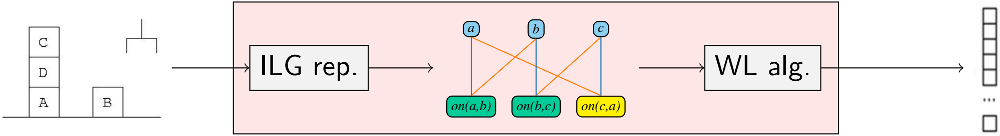
Contributions

1. *methodology*: new feature generation for planning
2. *theory*: provably more expressive than existing methods
3. *experiments*: competitive results on non-trivial benchmarks

1. Methodology: WL Features

New feature generator for planning states and problems

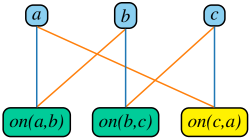
1. construct **Instance Learning Graph** (ILG) for a state
2. run modified Weisfeiler-Leman (WL) algorithm for generating features from ILGs



ILG representation

improved OA[1]/Muninn=Ståhlberg[2] graph representation

- nodes: objects, facts true in s_0, goal condition
  - colours: node type, predicate, goal information
  - *single* node for facts true in both state and goal
  - Muninn/OA graph unnecessarily *duplicates* such nodes
- edges: objects connected to facts
  - labels: location of instantiation of object



[1] Rostislav Horcik and Gustav Šír. "Expressiveness of Graph Neural Networks in Planning Domains". In: *ICAPS*. 2024.

[2] Simon Ståhlberg, Blai Bonet, and Hector Geffner. "Learning General Optimal Policies with Graph Neural Networks: Expressive Power, Transparency, and Limits". In: *ICAPS*. 2022.

WL algorithm

- iteratively refine colours of a graph based on message propagation
- hash function to compress neighbour colours

$$c^0(v) \leftarrow c(v), \forall v \in V$$
for $j = 1, \ldots, L$ **do for** $v \in V$ **do**
$$c^j(v) \leftarrow \mathrm{hash}\Big(c^{j-1}(v), \bigcup_{\iota \in \Sigma_E} \{\!\{(c^{j-1}(u), \iota) \mid u \in \mathcal{N}_\iota(v)\}\!\}\Big)$$
return $\bigcup_{j=0,\ldots,L} \{\!\{c^j(v) \mid v \in V\}\!\}$

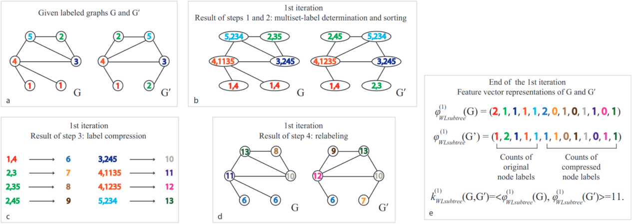
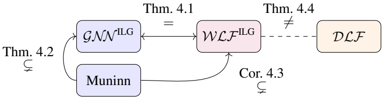


Image from [Shervashidze et al., JMLR-11]

$\rightsquigarrow$ we modify WL to support edge labelled graphs

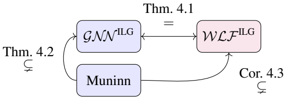
2. Theory: connections to related work



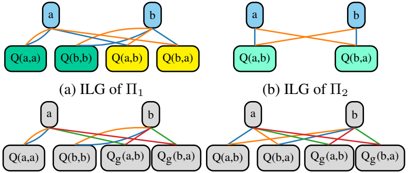
- Expressivity analysis
  - detect indistinguishable pairs[1]
- Compare to 2 streams of learning for planning research
  1. Graph Neural Network architectures ($\mathcal{GNN}^{\text{ILG}}$, Muninn)
  2. Description Logic Features ($\mathcal{DLF}$)

[1] Dillon Ze Chen, Sylvie Thiébaux, and Felipe Trevizan. "Learning Domain-Independent Heuristics for Grounded and Lifted Planning". In: *AAAI*. 2024.

Theory: WL Features vs GNNs



- Thm 4.1: WLFs upper bound GNNs on ILGs
  - Proof idea: WLFs upper bound GNNs
- Thm 4.2: WLFs/GNNs on ILGs are more expressive than Muninn[1]
  - Proof idea: Muninn cannot learn ”achieved goals”

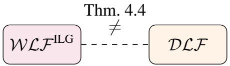


(a) ILG of Π_1 (b) ILG of Π_2

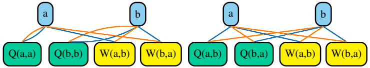
(c) Implicit Muninn graph of Π_1 (d) Implicit Muninn graph of Π_2

[1] Simon Ståhlberg, Blai Bonet, and Hector Geffner. “Learning General Optimal Policies with Graph Neural Networks: Expressive Power, Transparency, and Limits”. In: *ICAPS*. 2022.

Theory: WL Features vs DL Features



- Thm 4.4: WLFs and DLFs are incomparable
  - Proof ideas:
  - $\exists <$
    - $\mathcal{DLF}$ can distinguish some symmetric predicates
    - $\mathcal{WLF}$ cannot



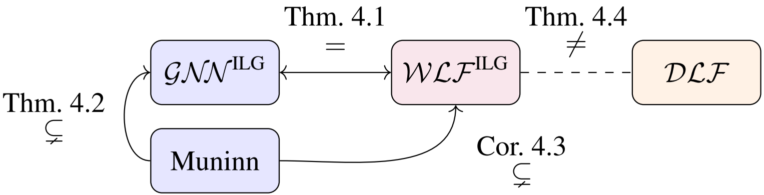
(a) ILG of Π_1 (b) ILG of Π_1

  - $\exists >$
    - $\mathcal{DLF}$ limited by compilation to binary predicates
    - $\mathcal{WLF}$ can distinguish some ternary predicates

$$s_0^1 = \{P(a,b,a), P(c,b,c), P(a,d,c), P(c,d,a)\}$$
$$s_0^2 = \{P(a,b,c), P(c,b,a), P(a,d,a), P(c,d,c)\}.$$

Theory: summary

- key takeaway: <u>WL Features ($\mathcal{WLF}$) most expressive, alongside $\mathcal{DLF}$</u>



- but all features still have *limited* expressivity

3. Experiments

Recall:

1. learn from **few** number of **small** problems
2. evaluate on **larger size** problems
3. do not use any IPC planner during evaluation

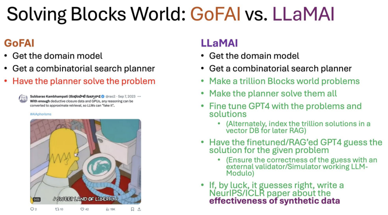


Image credits: Subbarao Kambhampati

✓ ok:

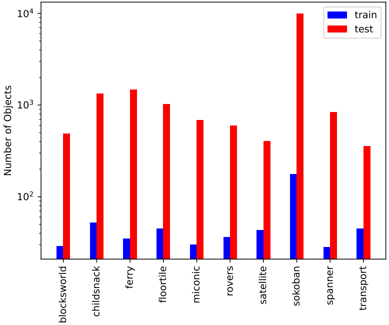
- learn heuristic
- learn policy

✗ not ok:

- learn transformation + LAMA
- learn $\emptyset$ + LAMA
- learn portfolios of IPC planners
- train and test on Blocksworld instances with 10 blocks

Experiments: setup

- IPC 2023 Learning Track (10 domains, 900 problems)
- 8GB memory; 1800s runtime
- GPU for NN models; single core CPU otherwise
- give all learners the same optimal training plans (< 99 per domain)



Train and problem sizes for each domain. Note the *log* scale.

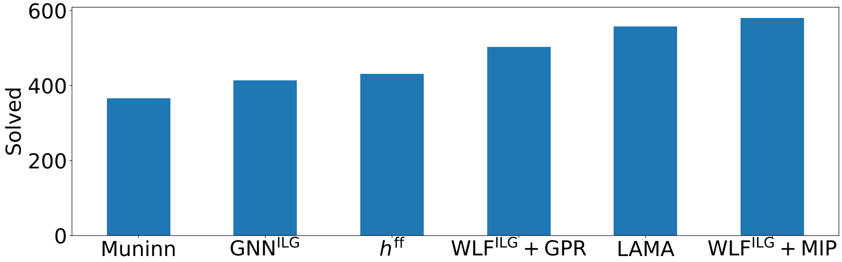
Experiments: baselines

- LAMA
- h^{FF}
- Muninn
- **new:** GNN on ILGs
- **new:** WLF on ILGs + Gaussian Process Regression

All single-queue GBFS except LAMA

Experiments: coverage results on IPC 2023 Learning Track

- higher = better ↑

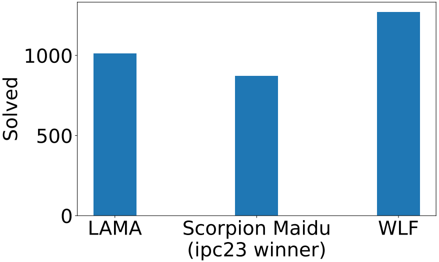


1. ILG encoding improves on Muninn encoding
2. WLF outperforms GNNs
3. WLF competitive with LAMA

theory matches practice

Experiments: real world planning problem

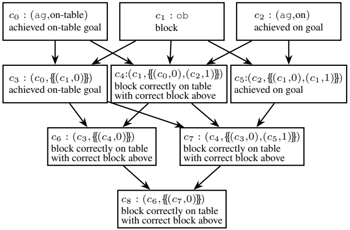
- Beluga Logistics Planning (Airbus)
- training data not chosen by me



Acknowledgements: Rebecca Eifler for PDDL encodings and training data

Many Other Results

- explainable features



- correlation results

Domain	*h* error				Expanded			
	easy	medium	hard	all	easy	medium	hard	all
blocksworld	**+0.93**	**+0.90**	**+0.94**	**+0.98**	*+0.32*	*+0.22*	*+0.33*	**+0.58**
childsnack	**+0.69**	**+0.93**	-	**+0.87**	**+0.59**	*+0.52*	-	*+0.20*
ferry	**+0.86**	**+0.98**	**+0.99**	**+1.00**	**+0.86**	**+0.87**	**+0.83**	**+0.93**
floortile	-	-	-	-	-	-	-	-
miconic	**+0.56**	**+0.67**	**+0.97**	**+0.96**	**+0.55**	**+0.81**	**+0.99**	**+0.99**
rovers	**+0.89**	**+0.86**	-	**+0.96**	*+0.26*	*+0.19*	-	**+0.53**
satellite	**+0.73**	**+0.95**	-	**+0.96**	*+0.09*	*+0.07*	-	*+0.18*
sokoban	*+0.27*	**+0.86**	-	**+0.96**	*+0.26*	**+0.76**	-	**+0.79**
spanner	**+0.36**	**+0.53**	**+0.96**	**+0.92**	**+0.43**	**+0.54**	**+0.96**	**+0.92**
transport	**+0.83**	-	-	**+0.83**	**+0.37**	-	-	**+0.35**

- super fast training

Domain	GNN		WLF			
	$\text{GOOSE}_{\text{max}}$	$\text{GOOSE}_{\text{mean}}$	SVR	SVR_{∞}	$\text{SVR}_{\text{2-LWL}}$	GPR
blocksworld	122.6	155.9	0.3	7.9	7.8	4.3
childsnack	36.9	46.0	0.0	3.1	0.2	0.1
ferry	56.1	112.2	0.2	6.5	3.4	1.5
floortile	122.2	146.0	0.5	28.3	OOM	14.9
miconic	46.5	50.2	0.1	95.3	0.6	0.5
rovers	100.4	88.4	0.3	18.5	7.2	7.5
satellite	49.7	97.6	0.3	13.1	7.7	1.8
sokoban	102.1	197.2	0.2	3.6	1.4	1.7
spanner	90.4	66.2	0.1	35.6	0.8	5.3
transport	44.7	41.9	0.1	0.6	2.1	0.5
all	77.2 ±33.7	100.2 ±52.6	0.2 ±0.1	21.3 ±28.4	3.1 ±3.2	3.8 ±4.6

- very small models

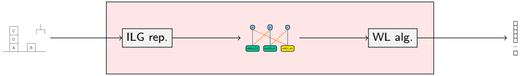
Domain	GNN	WL	improvement
blocksworld	54721	10444	5
childsnack	56257	251	224
ferry	54529	3228	17
floortile	55681	7616	7
miconic	54913	108	508
rovers	74561	23202	3
satellite	55297	22155	2
sokoban	70913	110	645
spanner	54913	350	157
transport	54721	3787	14

Return to Tradition: Learning Reliable Heuristics with Classical Machine Learning

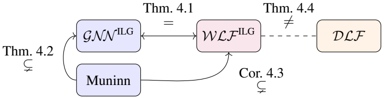
Dillon Z. Chen, Felipe Trevizan, Sylvie Thiébaux

New feature generation for planning, and SOTA results in learning for planning

(1) New feature generation method for planning tasks

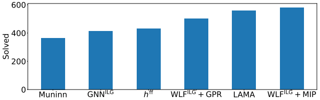


(2) Theoretical comparison to Graph Neural Networks and Description Logic Features for planning



(3) State-of-the-art results on competition benchmarks

- 2 orders of magnitude fewer parameters than GNN models
- 3 orders of magnitude faster training than GNN models
- 1st learned heuristics to outperform h^{FF} and match LAMA in a non-trivial competition setting



Code at `https://github.com/DillonZChen/goose`

(∗) WL Features available as a Python/C++ package

- `pip install wlplan`
- `https://github.com/DillonZChen/wlplan`

Thanks! Questions?