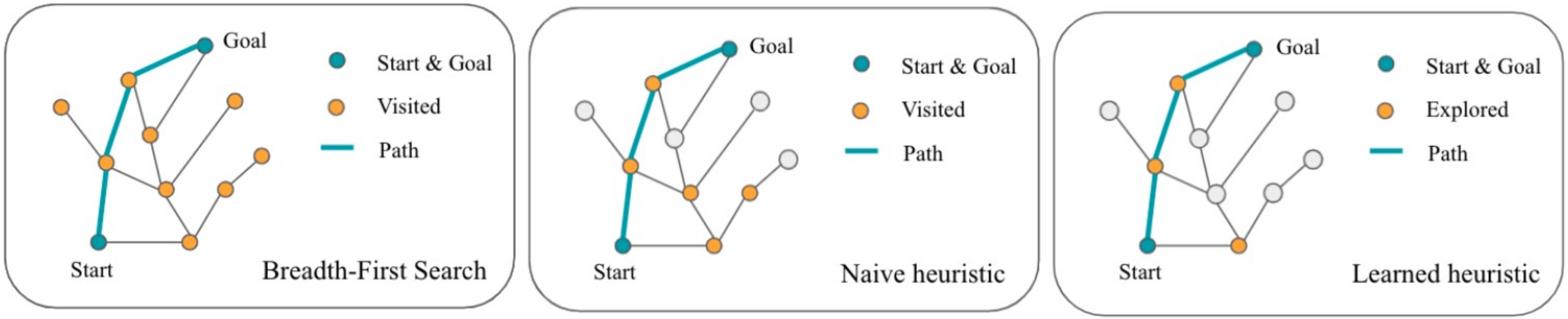
Learning Graph Search Heuristics

LoG 2022 submission

Michal Pándy, Weikang Qiu, Gabriele Corso, Petar Veličković, Rex Ying, Jure Leskovec, Pietro Lio

Motivation

- Search heurisitics: robotics, AI, biology, chemistry, …

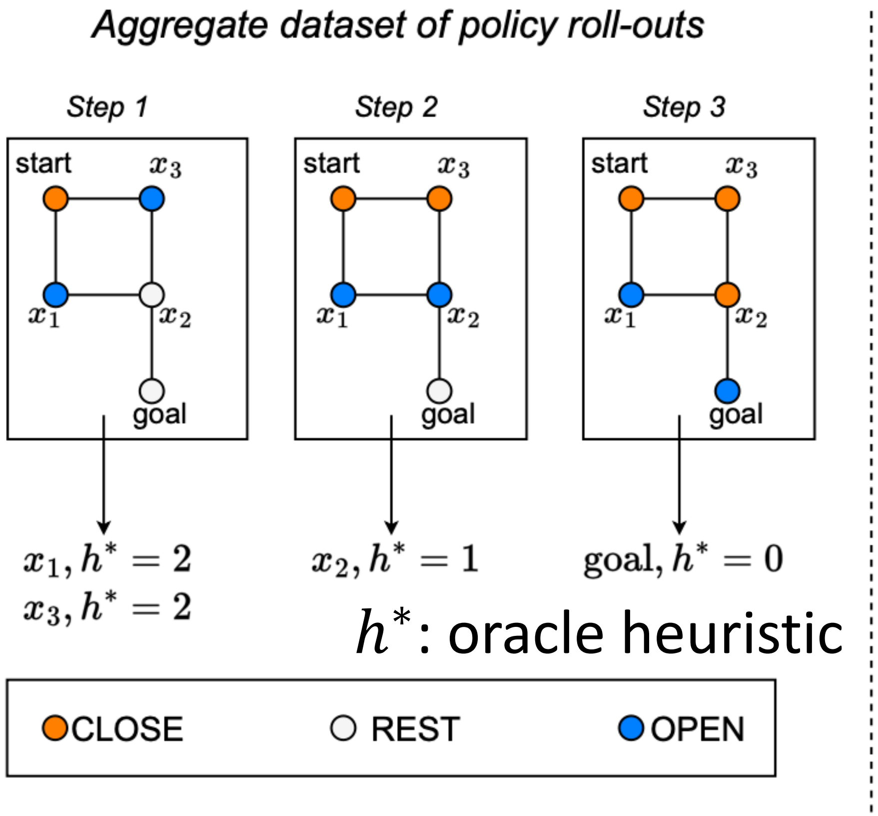


- Traditional methods
  - BFS: should explore many nodes
  - Heuristic function: need domain expretise and manual effort
- Learning-based methods:
  - Non-i.i.d data
  - Efficiency
  - Reinforcement learning-based heuristic perform poorly as graph sizes increase

PHIL: Path Heuristic with Imitation Learning

Preliminaries: Graph Search

- CLOSE ∪ OPEN ∪ REST
- Greedy best-first search
  - Exapnd node that **minimizes** the heuristic function in OPEN
  - Each expansion **moves a node** from OPEN to CLOSE, and **adds the neighbors** of the chosen node from REST to OPEN



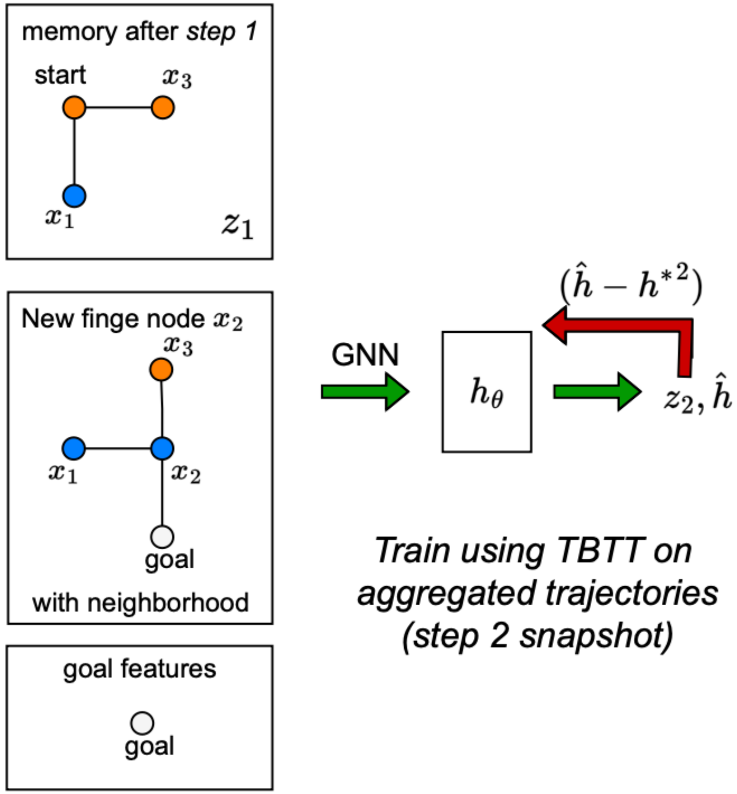
Preliminaries: Imitation Learning

- MDP (Partially observed, POMDP)
  - State: newly included nodes in OPEN set, also include memory history
  - Action: choose one node in OPEN set to expand

- Behavior cloning
  - Expert's actions as labels
- Learning oracle's cost-to-go function (Q)

$$\pi_\theta(s_t) = \underset{a_t \in \mathcal{A}}{\operatorname{argmin}} Q_\theta(s_t, a_t)$$

h_θ



Algorithm

- Training objective

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{\substack{\mathcal{G}\sim P_{\mathcal{G}}, \\ (v_s, v_g)\sim P_{v_{sg}} \\ t\sim\mathcal{U}(0,\ldots,T), \\ s,\psi\sim P_s}} \left[\frac{1}{|\mathrm{OPEN}|}\sum\nolimits_{v\in\mathrm{OPEN}} \left(h^*(s, v, v_g) - h_\theta(\psi, v, v_g)\right)^2\right]$$

Where

$h_\theta\colon \mathbb{R}^d\times\mathcal{O}\times\mathcal{V}\times\mathcal{V} \mapsto \mathbb{R}$ — learned heuristics

h^* — oracle heuristics

ψ — state-history

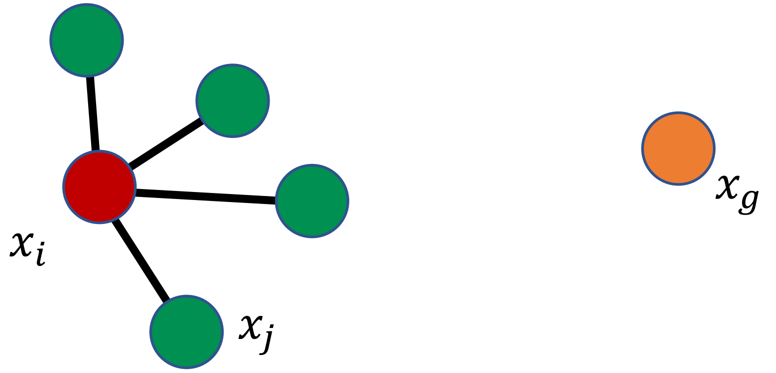
Algorithm: Imitation Learning

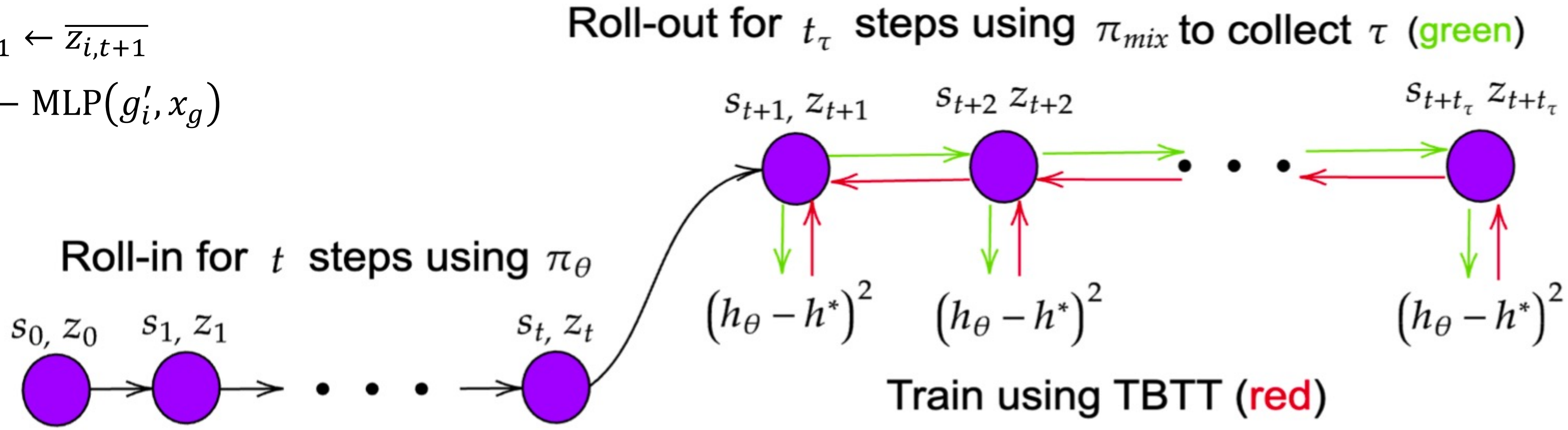
- **Main idea: collect trajectories**

- Set mixture policy: $\pi_{mix} \leftarrow (1-\beta_i)\pi_{\theta_i} + \beta_i \pi^*$
- Roll-in t timesteps (do not collect data)
- Roll-out trajectory τ_i, recording each timestep's state. i.e. $\tau_i \leftarrow \tau_i \cup \left\{\left(\mathcal{V}_{\text{new}}, h^*\left(s_{t+k}, \mathcal{V}_{\text{new}}, v_g\right)\right)\right\}$
- Update dataset with trajectory and initial memory (τ_i, z_t)
- Training using this dataset.

Algorithm: Model Architecture

- $x_i \leftarrow f\left(x_i, x_g, D_{EUC}(x_i, x_g), D_{COS}(x_i, x_g)\right)$
- $x_j \leftarrow f\left(x_j, x_g, D_{EUC}(x_j, x_g), D_{COS}(x_j, x_g)\right)$
- $g_i \leftarrow \phi\left(x_i, \bigoplus_{j \in \mathcal{N}_i} \gamma(x_i, x_j, e_{ij})\right)$
- $g_i', z_{i,t+1} \leftarrow \mathrm{GRU}(g_i, z_t)$
- $z_{t+1} \leftarrow \overline{z_{i,t+1}}$
- $\hat{h}_i \leftarrow \mathrm{MLP}(g_i', x_g)$



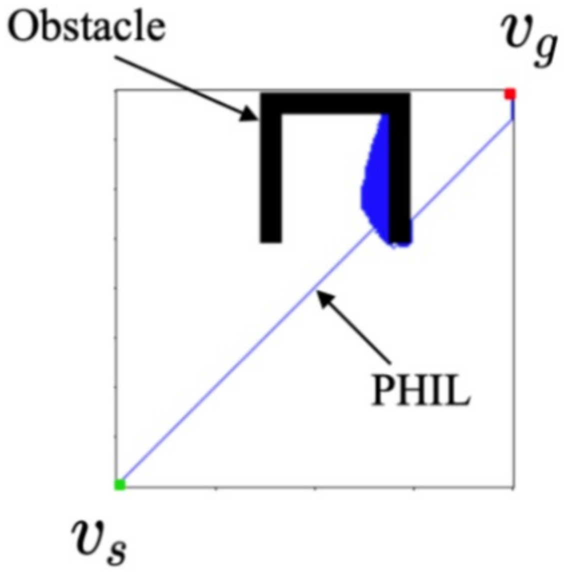


Experiments: Search in grids

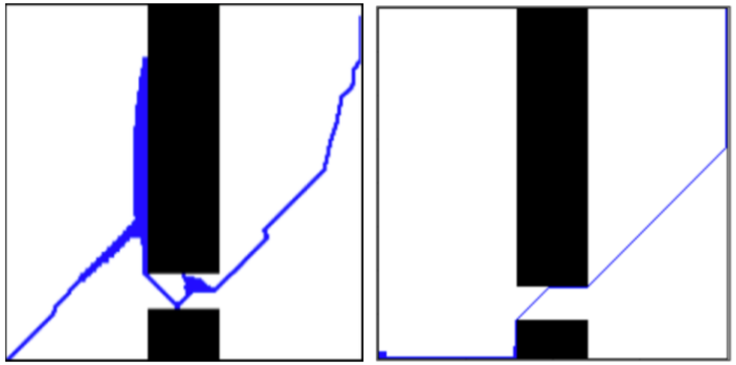
Dataset	Graph Examples	SAIL	SL	CEM	QL	h_{euc}	h_{man}	A*	MHA*	BFWS	Neural A*	PHIL
Alternating gaps		0.039	0.432	0.042	1.000	1.000	1.000	1.000	1.000	0.34	0.546	**0.024**
Single Bugtrap		0.158	0.214	0.057	1.000	0.184	0.192	1.000	0.286	0.099	0.394	**0.077**
Shifting gaps		0.104	0.464	1.000	1.000	0.506	0.589	1.000	0.804	0.206	0.563	**0.027**
Forest		0.036	0.043	0.048	0.121	0.041	0.043	1.000	0.075	0.039	0.399	**0.027**
Bugtrap+Forest		0.147	0.384	0.182	1.000	0.410	0.337	1.000	3.177	0.149	0.651	**0.135**
Gaps+Forest		0.221	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.401	0.580	**0.039**
Mazes		0.103	0.238	0.479	0.399	0.185	0.171	1.000	0.279	0.095	1.000	**0.069**
Multiple Bugtraps		0.479	0.480	1.000	0.835	0.648	0.617	1.000	0.876	0.169	0.331	**0.136**

- Comparing to SaIL
  - 58.5% reduction of explored nodes
  - 5x less data

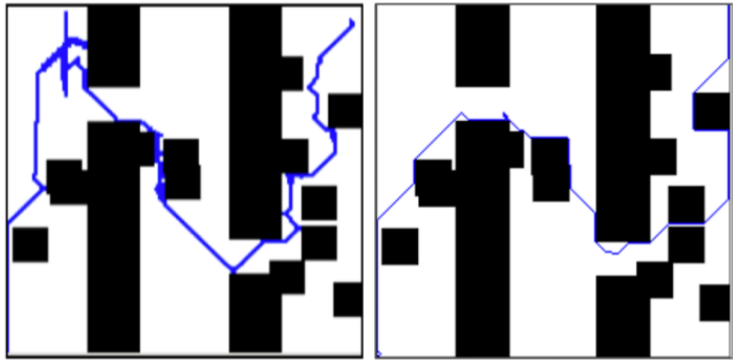
Experiments: Search in grids visualization



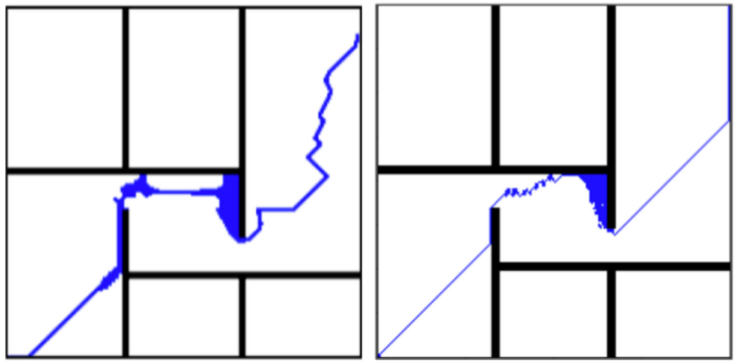
Escape local minima



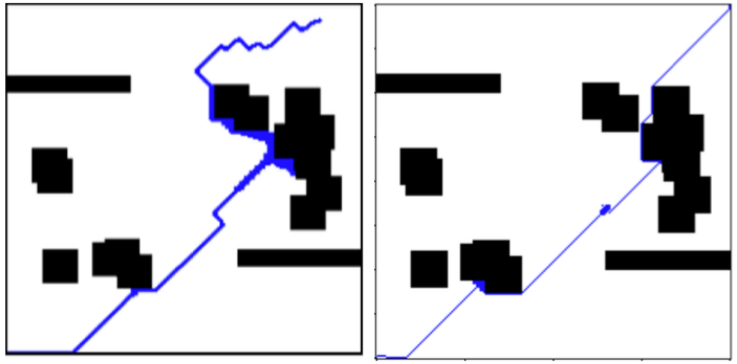
(a) Shifting gaps (SAIL left, PHIL right)



(b) Gaps+Forest (SAIL left, PHIL right)



(c) Mazes (SAIL left, PHIL right)



(d) Forest (SAIL left, PHIL right)

Reduce the redundancy of explored nodes

Experiments: Search in real-life graphs

	Dataset	$\lvert\mathcal{D}\rvert$	$\lvert\bar{\mathcal{V}}\rvert$	$\lvert\bar{\mathcal{E}}\rvert$	**SL**	**A***	h_{euc}	**BFS**	**SAIL**	**BFWS**	**PHIL**
Citation Networks	Cora (Sen *et al.* [49])	1	2,708	5,429	2.201	2.067	1.000	4.001	0.669	1.378	**0.475**
	PubMed (Sen *et al.* [49]))	1	19,717	44,338	2.157	2.983	1.000	3.853	1.196	1.000	**0.745**
	CiteSeer (Sen *et al.* [49]))	1	3,327	4,732	1.636	1.487	1.000	2.190	1.062	0.951	**0.599**
	Coauthor (cs) (Schur *et al.* [50])	1	18,333	81,894	1.571	1.069	1.000	2.820	1.941	1.026	**0.835**
	Coauthor (physics) (Schur *et al.* [50])	1	34,493	247,962	4.076	1.081	1.000	4.523	–	1.012	**0.964**
Biological Networks	OGBG-Molhiv (Hu *et al.* [48])	41,127	25.5	27.5	1.086	1.065	**1.000**	1.267	1.104	1.146	1.016
	PPI (Zitnik *et al.* [51])	24	2,372.67	34,113.16	0.772	0.831	1.000	5.618	1.746	3.941	**0.658**
	Proteins (Full) (Morris *et al.* [52])	1,113	39.06	72.82	0.995	0.997	1.000	2.645	0.891	0.966	**0.831**
	Enzymes (Morris *et al.* [52])	600	32.63	62.14	1.073	1.007	1.000	1.358	1.036	0.992	**0.757**
ASTs	OGBG-Code2 (Hu *et al.* [48])	452,741	125.2	124.2	1.196	1.013	1.000	1.267	1.029	**0.817**	1.219
Road Networks	OSMnx - Modena (Boeing [53])	1	29,324	38,309	2.904	3.085	1.000	3.493	1.182	0.997	**0.489**
	OSMnx - New York (Boeing [53])	1	54,128	89,618	39.424	36.529	1.000	63.352	1.583	1.013	**0.962**

- 13.4% improvement
- Use only 0.2% data

Experiments: Planning for drone flight

Dataset	SL	A*	h_{euc}	BFS	SAIL	BFWS	PHIL	Shortest path
Room simple	1.124	76.052	1.000	291.888	0.973	1.286	**0.785**	0.782
Room adversarial	2.022	67.215	1.000	238.768	0.944	1.583	**0.895**	0.853