Scaling Multi-Agent Epistemic Planning through GNN-Derived Heuristics

The 25th International Conference on Autonomous Agents and Multiagent Systems - AAMAS 2026 - Paphos

Giovanni Briglia[1,2], Francesco Fabiano[3] and Stefano Mariani[2]

[1]**University of Pisa**, Pisa, Italy
Department of Computer Science
[2]**University of Modena and Reggio Emilia**, Reggio Emilia, Italy
Department of Sciences and Methods for Engineering
[3]**University of Oxford**, Oxford, UK
Department of Computer Science

May 27, 2026

Overview

1. Introduction
2. Background MEP and DEL
3. Problem Formalization
4. Our Contribution
5. Experiments and Validation
6. Conclusions

Chapter 1

Introduction

Research Snapshot

- Multi-Agent Epistemic Planning (MEP) solvers face *exponential blind search* — no learned heuristics exist for Kripke-structured e-states.
- We train a *GNN over Kripke structures* to approximate the perfect heuristic (distance-to-goal), then plug it into informed search (HFS*).
- Result: consistently *fewer expanded nodes* than BFS across 6 benchmarks, and competitive with the best hand-crafted H-EFP heuristic.

Chapter 2

Background MEP and DEL

Dynamic Epistemic Logic and Kripke structures, briefly

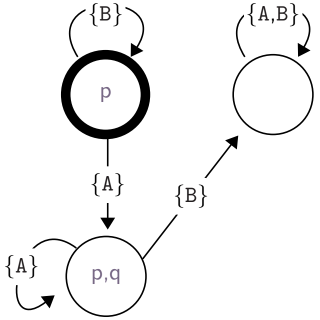
Dynamic Epistemic Logic

Dynamic Epistemic Logic is a modal logic that formalizes agents' beliefs, about the world's properties and other agents' beliefs, and how these beliefs evolve after the execution of an action.

Pointed Kripke structure

$(\langle S, \pi, \mathcal{B}_1, \ldots, \mathcal{B}_n \rangle,\ s_0)$ where:

- S: worlds; $s_0 \in S$: designated real world
- $\pi : S \to 2^{\mathcal{F}}$: interpretation of each world
- $\mathcal{B}_i \subseteq S \times S$: belief relation for agent i



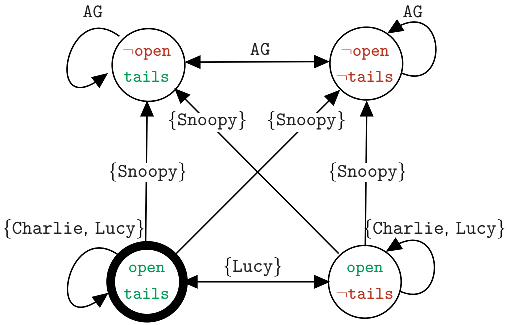
MEP in one example

Initial State

- The box is `closed`
- No one knows if the coin is `tails` up

Goal State

- `Charlie` knows that the coin is `tails` up
- `Lucy` knows that `Charlie` knows
- `Snoopy` is unaware of the plan execution



MEP problem, formally

Definition 2.2 (MEP Problem)

An MEP problem is a tuple $P = \langle D, I, \mathcal{G} \rangle$ where domain $D = \langle \mathcal{F}, \mathcal{AG}, \mathcal{A} \rangle$ consists of:

- $\mathcal{F}$: *fluents* (propositional variables)
- $\mathcal{AG}$: *agents*
- $\mathcal{A}$: *actions* — ontic, sensing, or announcement

I, $\mathcal{G}$ are sets of belief formulae specifying *initial* and *goal* conditions. A *plan* is a sequence of actions in $\mathcal{A}$ transforming the initial e-state into one satisfying $\mathcal{G}$.

Chapter 3

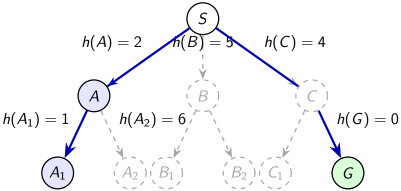
Problem Formalization

The bottleneck

- E-states are *Kripke structures* — directed labeled graphs that *grow unboundedly* with plan length.
- Existing approaches mostly rely on *blind BFS*.
- *Search-space explosion*, intractability on non-trivial instances.

Our Idea: Learn the Heuristic

- *Heuristic guidance* is what makes search tractable everywhere else in planning (e.g., AlphaGo's MCTS [Sil+16]).
- MEP lacks informed search approaches, since its fundamentally different epistemic-state structure makes existing heuristic search methods difficult to adapt effectively.
- We close that gap with a *data-driven GNN heuristic*.

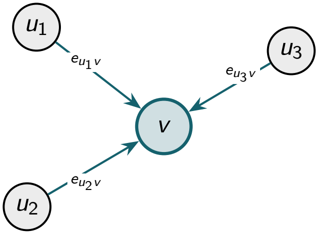


Chapter 4

Our Contribution

Graph Neural Networks (GNNs)

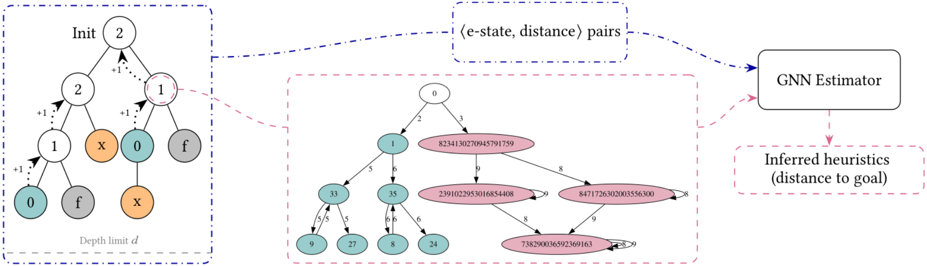
- **What:** iteratively updates each node's representation from its neighbors' — *message passing* along edges.
- **Why for MEP:** e-states are labeled directed graphs that grow unboundedly. Fixed-size encoders cannot ingest them; sequence models can lose relational structure.
  $\Rightarrow$ *GNNs are the natural fit.*
- **Which variant:** *GINEConv* [Hu+20] — uses both node features *and* edge attributes, essential for Kripke structures with heterogeneous edge types. More expressive than GCN or GAT.



$$h_v^{(\ell+1)} = \phi\big(h_v^{(\ell)}, \textstyle\sum_{u\in N(v)} \psi(h_u^{(\ell)}, e_{uv})\big)$$

GNN-Based Heuristics: Motivation and Training

E-states are graphs that grow unboundedly; fixed vector encoders don't fit. *GNNs do.*
Data representation matters: we evaluated map/*hash*/bitmask-based encodings of Kripke structures; hash wins on expanded nodes and wall-clock time at equal coverage.



1. DFS on *small instances* yields ⟨e-state, true distance⟩ pairs

2. GINEConv learns *structural patterns* of Kripke structures predicting distance to goal

3. Applied to *unseen* e-states as $h(s)$ in HFS*: $f(s) = h(s) + d(s)$

One design choice that mattered: HFS*

Search	Solved Inst.	Nodes	Time [ms]	Length
HFS	49/79 (62.03%)	18 ± 27	1290 ± 5528	8 ± 8
HFS*	**75/79 (94.94%)**	$\mathbf{17 \pm 42}$	$\mathbf{584 \pm 2276}$	$\mathbf{6 \pm 4}$

Comparison of HFS and HFS* (aggregate over all domains).

HFS* augments the learned heuristic $h(s)$ with the search depth $d(s)$, A*-style:

$$f(s) = h(s) + d(s)$$

Without this, the GNN estimate alone is too noisy to guide search — adding depth restores monotonic progress. **Admissibility** of the learned heuristic is not formally guaranteed. However, HFS* is **complete** under a bounded plan length: the MEP search space grows monotonically with depth, so a depth-bounded HFS* exhausts it in finite time.

Chapter 5

Experiments and Validation

GNN vs BFS across 6 domains

Domain	Solved (GNN)	Nodes (GNN)	Solved (BFS)	Nodes (BFS)	Red. (%)
AL	6/7 (85.7%)	10 ± 0	6/7 (85.7%)	14 ± 0	28.6
CC	18/18 (100%)	65 ± 250	18/18 (100%)	610 ± 1607	89.3
CB	3/3 (100%)	75 ± 860	3/3 (100%)	102 ± 1260	26.5
GR	8/12 (66.7%)	*157 ± 380*	10/12 (83.3%)	*448 ± 2068*	65.0
SC	19/20 (95.0%)	49 ± 357	19/20 (95.0%)	114 ± 372	57.0
SR	5/6 (83.3%)	6188 ± 17295	5/6 (83.3%)	7918 ± 22608	21.8
All	59/66 (89.4%)	64 ± 296	61/66 (92.4%)	242 ± 1080	73.6

Consistent reduction in expanded nodes; GR is the exception due to sparse solution density.

Cross-planner comparison is deliberately omitted: prior to the recent E-PDDL standardization, porting MEP instances across solvers is non-trivial, and multi-agent A* is not uniformly supported.

Versus hand-crafted heuristics on Generalization

Approach	# Solved	% Solved
GNN	**64/75**	**85.33%**
C_PG	37/75	49.33%
L_PG	54/75	72.00%
S_PG	62/75	82.67%
SUB	58/75	77.33%

CC-GR *against $\mathcal{H}$-EFP's individual heuristics on unseen domains (AL-CB-SC-SR).*

Competitive with the best H-EFP [Fab+24] heuristic (S_PG, 82.7%), and generalizes to unseen domains.

Chapter 6

Conclusions

Limitations (expected)

- **Runtime not yet competitive:** no batched CUDA inference. We report expanded nodes because it measures heuristic informativeness independent of engineering.
- **GR domain:** sparse valid plans $\rightarrow$ weak learning signal.
- **Deliberately simple baseline**: stronger architectures and training tricks were avoided on purpose: we wanted a clean answer to

  does this approach work at all?

  before optimizing.

Take-home

- *First learned heuristic for MEP*, integrated end-to-end in a solver (deep[1]).
- Competitive with hand-crafted heuristics; *generalizes across domains*.
- Foundational: opens MCTS and online RL for MEP; that's the next step.

[1] `github.com/FrancescoFabiano/deep`

Thank You
for the attention!

References I

[Fab+24] Francesco Fabiano et al. "$\mathscr{H}$-Efp: Bridging Efficiency in Multi-agent Epistemic Planning with Heuristics". In: *PRIMA 2024: Principles and Practice of Multi-Agent Systems - 25th International Conference, Kyoto, Japan, November 18-24, 2024, Proceedings*. Ed. by Ryuta Arisaka et al. Lecture Notes in Computer Science. Springer, 2024, pp. 81–86. DOI: 10.1007/978-3-031-77367-9_7. URL: https://doi.org/10.1007/978-3-031-77367-9_7.

[Hu+20] Weihua Hu et al. "Strategies for Pre-training Graph Neural Networks". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. Addis Ababa, Ethiopia: OpenReview.net, 2020. URL: https://openreview.net/forum?id=HJlWWJSFDH.

References II

[Sil+16] David Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nat.* 529.7587 (2016), pp. 484–489. DOI: 10.1038/NATURE16961. URL: https://doi.org/10.1038/nature16961.