UNIVERSITY OF ABERDEEN

ROBUST NEURO-SYMBOLIC GOAL AND PLAN RECOGNITION

Leonardo Amado[1], Ramon Fraga Pereira, [2], Felipe Meneguzzi [3,1]

[1] Pontifícia Universidade Católica do Rio Grande do Sul, Brazil

[2] Sapienza University of Rome, Italy

[3] University of Aberdeen, Scotland

February 11, 2023

INTRODUCTION

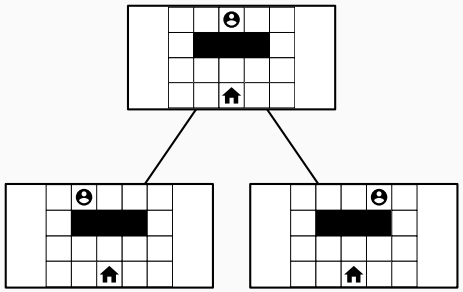
Definition (Goal Recognition Task)

A goal recognition task $\Pi^{\Omega}_{\mathcal{G}} = \langle \Xi, \mathcal{I}, \mathcal{G}, \Omega \rangle$ is a tuple composed of a domain definition Ξ, an initial state $\mathcal{I}$, a set of goal hypotheses $\mathcal{G}$, and a sequence of observations Ω.

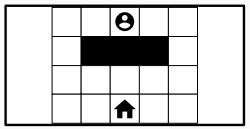
Background

Goal/Plan Recognition problems have three key ingredients

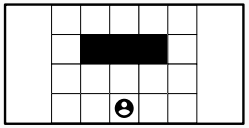
Domain Description



Initial State

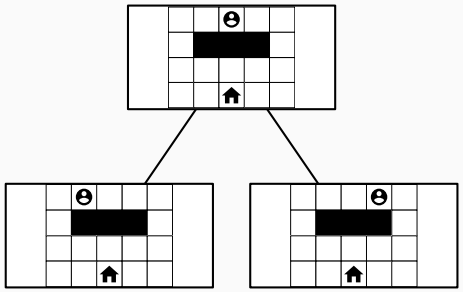


Goal State

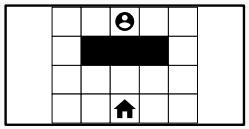


Goal/Plan Recognition problems have **four** key ingredients

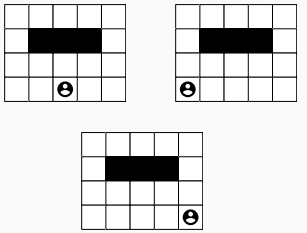
Domain Description



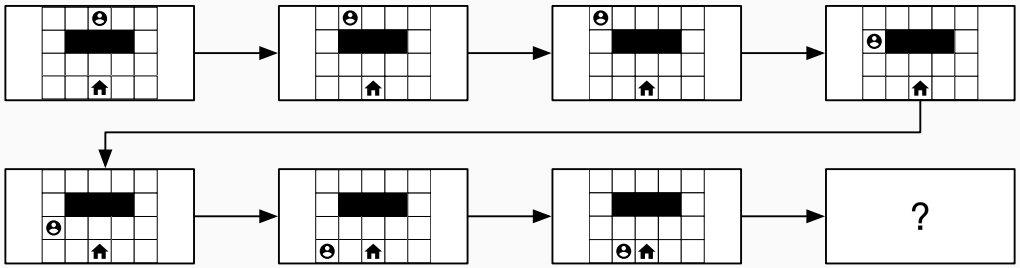
Initial State



Goal **Hypotheses**

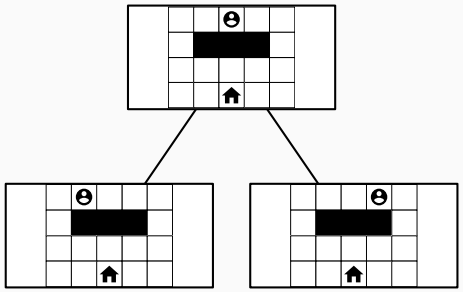


Observations

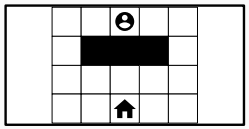


Goal/Plan Recognition problems have **four** key ingredients

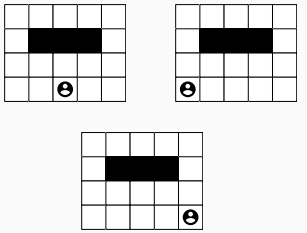
Domain Description



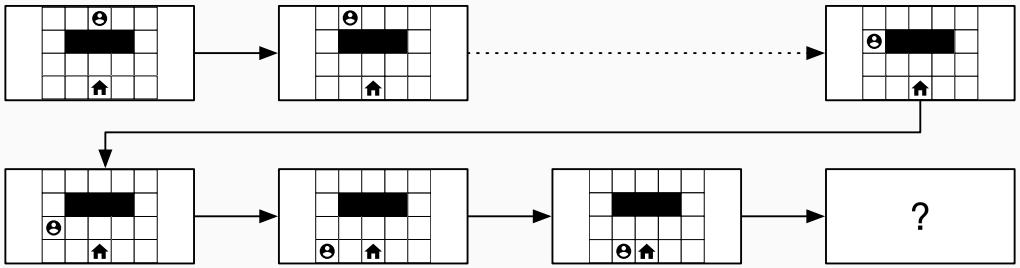
Initial State



Goal **Hypotheses**

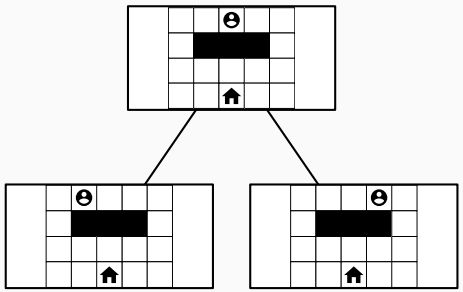


Observations

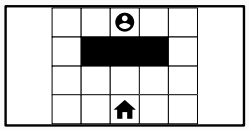


Goal/Plan Recognition problems have **four** key ingredients

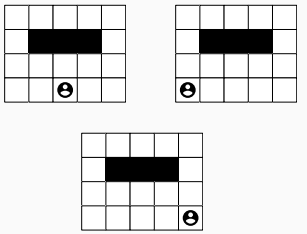
Domain Description



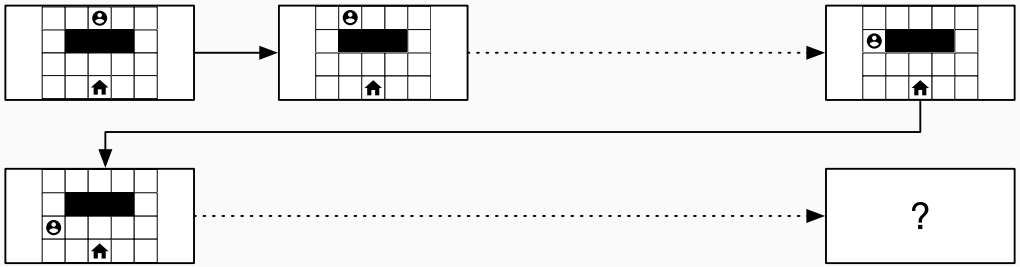
Initial State



Goal **Hypotheses**

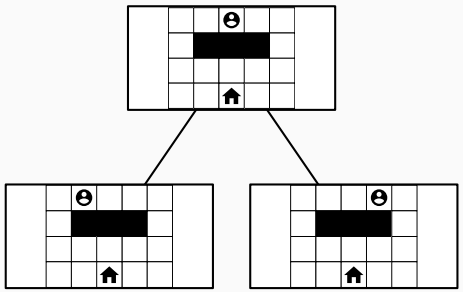


Observations

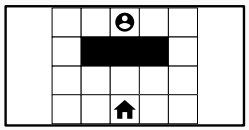


Goal/Plan Recognition problems have **four** key ingredients

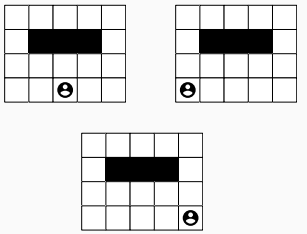
Domain Description



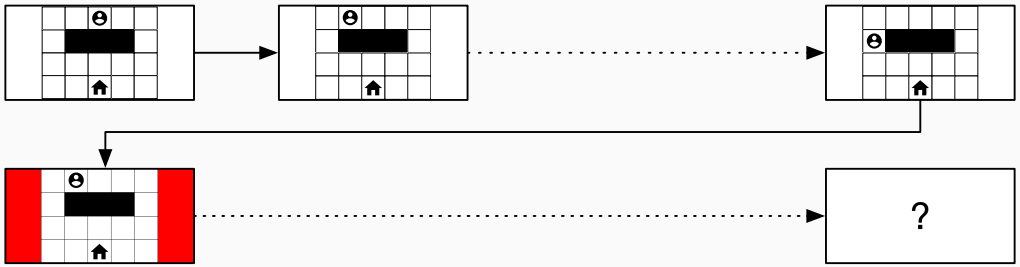
Initial State



Goal **Hypotheses**

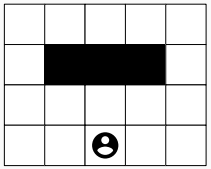


Observations



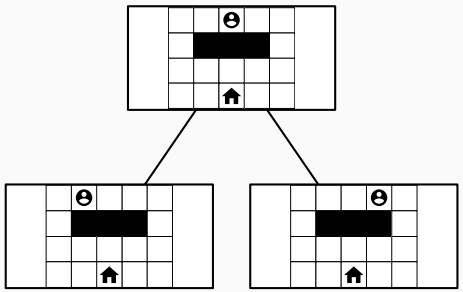
Solution

Correct Goal

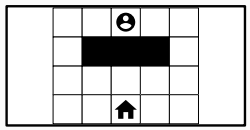


Goal/Plan Recognition problems have **four** key ingredients

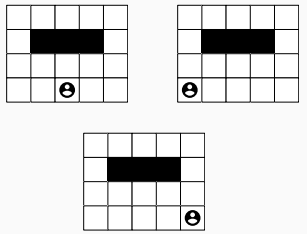
Domain Description



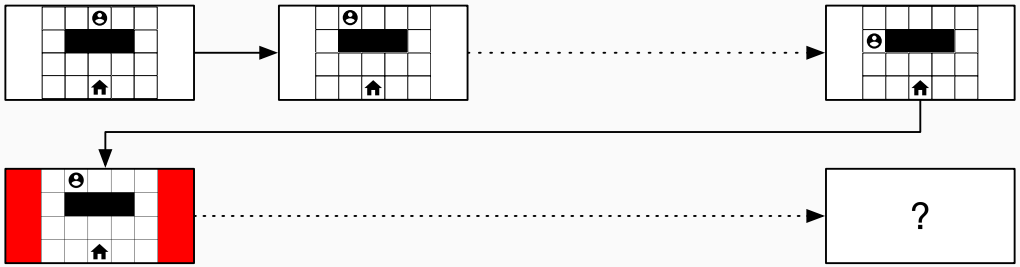
Initial State



Goal **Hypotheses**

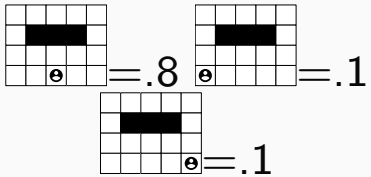


Observations



Solution

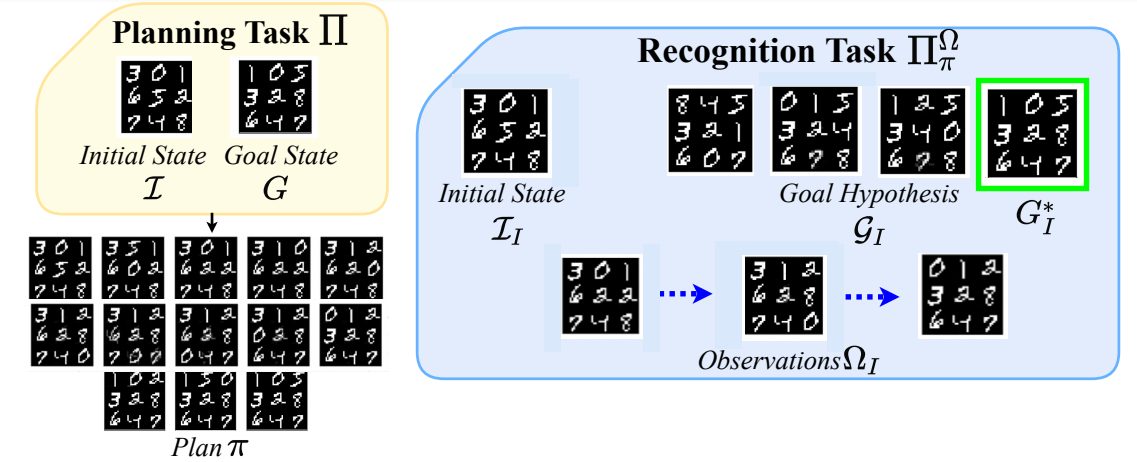
Probability Distribution



Goal and plan recognition example

Definition (Goal and plan recognition problem)

We formally define a plan (alternatively goal) *recognition problem* Π^{Ω}_{π} (alternatively $\Pi^{\Omega}_{\mathcal{G}}$) as tuple $\langle \Xi, \mathcal{I}, \mathcal{G}, \Omega \rangle$, where Ξ is a planning domain, $\mathcal{I}$ is an initial state, $\mathcal{G}$ is a set of *goal hypotheses*, which includes a correct goal G^* (unknown to the observer), and Ω is a sequence of observations[1]



[1] Ramírez and Geffner, "Plan recognition as planning".

Motivation

Recent approaches to goal and plan recognition improve performance under partial and noisy observability:

- However, dealing with these problems remains a challenge.[2]

Regardless of technique, the quality of the available observations directly affects performance.

This work develops *neuro-symbolic* recognition approaches

- Can combine learning and planning techniques
- Thus compensating for noise and missing observations using prior data

[2] Meneguzzi and Pereira, "A Survey on Goal Recognition as Planning".

Contributions

We evaluate our approaches in standard human-designed planning domains as well as domain models automatically learned from real-world data[3]

Our approach outperforms the baseline in both types of domains

- Empirical experimentation shows that our approaches reliably infer goals and compute correct plans in the experimental datasets.

Our overall framework is easily adaptable and extendable.

[3] Amado et al., "Latrec: Recognizing goals in latent space".

PPR

PPR Overview

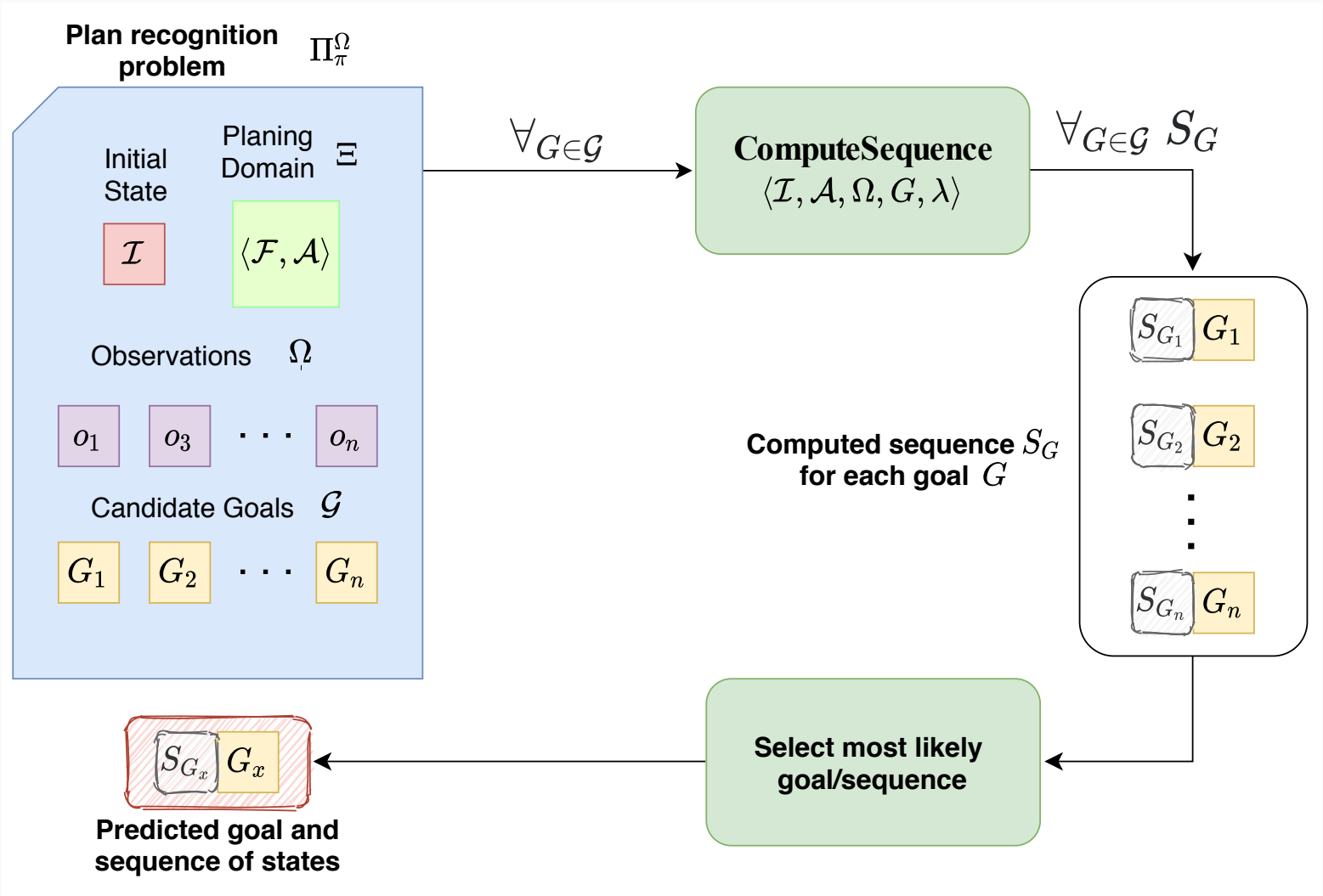
We solve plan recognition by computing a sequence of intermediary states achieved by a plan π.

Our algorithm tries to rebuild the sequence of states induced by a plan:

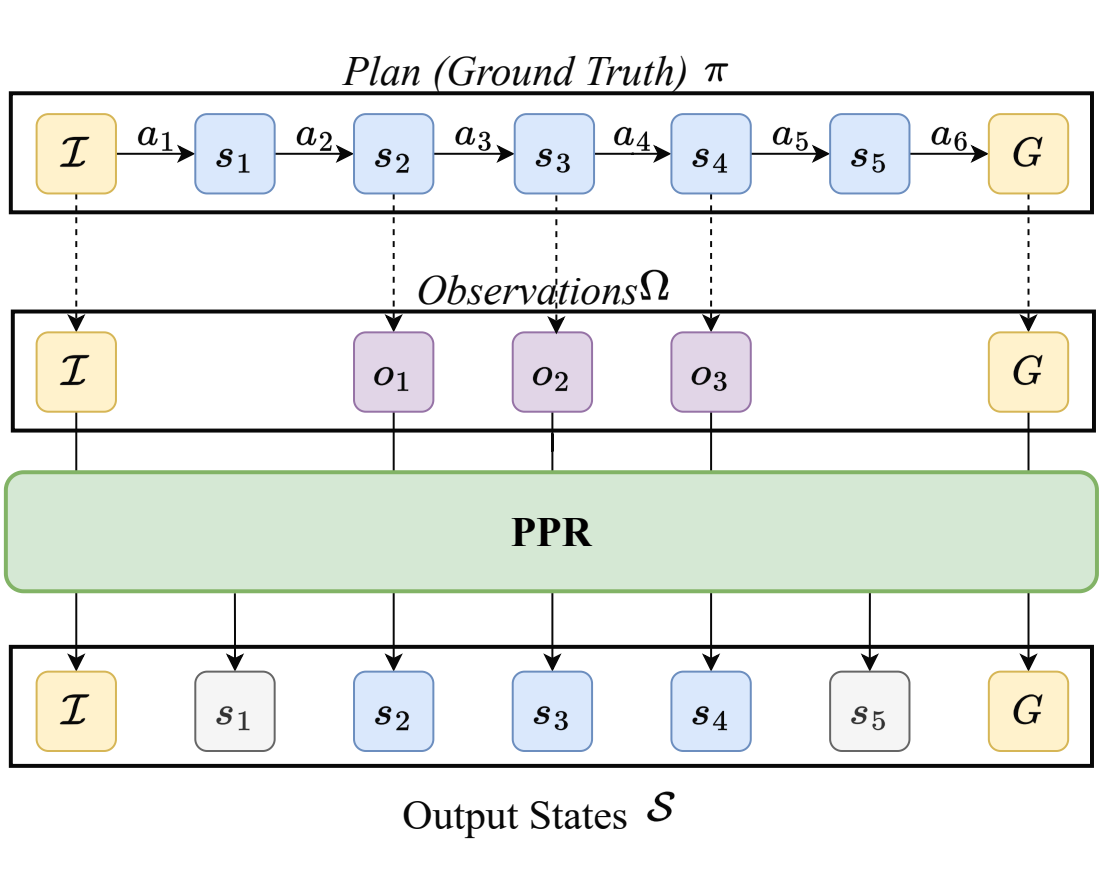
1. iterating through the sequence of observations; and
2. filling in any gaps due to partial observability.

To compute the intermediary states, we use *predictor functions*.

PPR Overview



Compute Sequence (Missing Observations)

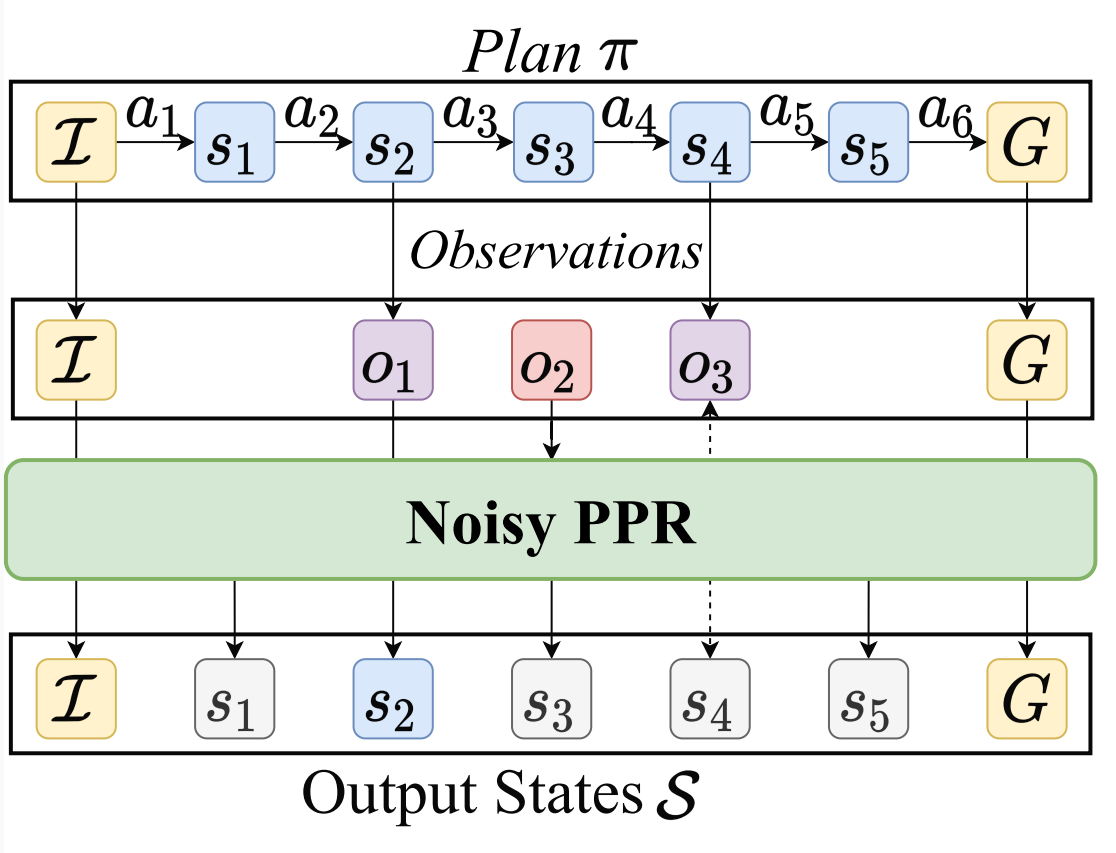


Compute Sequence

Algorithm 1: $\textsc{computeSequence}(\mathcal{I}, \mathcal{A}, \Omega, G, \lambda)$

```
1  S ← ⟨I⟩
2  if Ω_|Ω| ⊨ G then Ω̂ ← Ω else Ω̂ ← Ω · G
3  predicted ← 0
4  for o in Ω̂ do
5      while ¬∃_{∈A}(o = γ(S_|S|, a)) do
6          s' ← predictNextS(A, S, G, o)
7          S ← S · s'
8          predicted += 1
9          if G ∈ S or predicted > λ then return S
10     S ← S · o
11     predicted ← 0
12 return S
```

Compute Sequence (Noisy Observations)



Predictor functions

Given a sequence of states $\mathcal{S}$, a *predictor function* should ideally compute the probability $\mathbb{P}(s' \mid \mathcal{S})$ of all states in the state space $2^{\mathcal{F}}$ being the next successor state s', thus selecting the most likely state $\arg\max_{s' \in 2^{\mathcal{F}}} |p(s' \mid \mathcal{S})|$.

We develop and evaluate 3 predictor functions:

1. A machine learning approach using LSTMs to predict s' ($\mathtt{PPR}^{\sigma}$).
2. A purely symbolic function leveraging planning heuristics ($\mathtt{PPR}_{\mathsf{h}}$).
3. Finally, we combine both previous predictors functions to create a *neuro-symbolic approach* ($\mathtt{PPR}_{\mathsf{h}}^{\sigma}$)

Estimates the most likely next state using an LSTM, which receives a sequence of states as input.

Network output is a set of facts (true or false) comprising an approximately reconstructed state $\hat{s}$.

Since we have no guarantees that the reconstructed state is valid, we use $\hat{s}$ to discriminate the successor state.

Logically, the successor state must be achievable from the last state $\mathcal{S}_{|\mathcal{S}|}$ of the prior state sequence $\mathcal{S}$.

PPR$^{\sigma}$ chooses the most likely goal candidate by:

computing the achievable states C from $\mathcal{S}_{|\mathcal{S}|}$ using the planning domain; and

comparing the cosine distance of each achievable state $s \in C$, selecting as s' the state that is closer to the model's prediction.

$$s' = \arg\min_{c \in C} |\cos(bin(c), \hat{s})| \tag{1}$$

where $bin(c)$ is a binary representation of a candidate state c.

PPR_h predicts the most likely next state s' using exclusively a heuristic function h.

Again, we compute all achievable successor states C from the last state in the sequence $\mathcal{S}$.

Given all the achievable states C (candidate states), we apply a domain-independent heuristic to evaluate which state in C is the most likely next state.

To choose the most likely next state in C, we compute the mean h_s of the heuristic value h from each state $s \in C$ to the next observation o, and from each state to the goal hypothesis G, as follows:

$$h_s(s, o, G) = (h(s, o) + h(s, G))/2$$

Finally, we select the state s' with the lowest h_s value as the most likely successor:

$$s' = \arg\min_{c \in C} |h_s(c, o, G)|$$

$\text{PPR}_{\text{h}}^{\sigma}$, combines the two previous predictor functions, using a machine learning model and a planning heuristic.

This approach aims to combine the reconstruction of PPR^{σ} with the heuristic distance between goal hypotheses and the next observation of PPR_{h}.

Compute Sequence

Algorithm 2: predictNextS$(\mathcal{A}, \mathcal{S}, G, o)$

$\mathcal{M} \leftarrow$ any model for $p(\mathrm{f} \mid \mathcal{S})$ /* e.g. an ANN*/
$\vartheta \leftarrow$ confidence of $\mathcal{M}$
$\hat{s} \leftarrow$ a state computed using $\mathcal{M}(\mathcal{S})$
$s \leftarrow \mathcal{S}_{|\mathcal{S}|}$
$C \leftarrow \{s' \mid s' = \gamma(s, a), \text{for } a \in A, \text{s.t. } s \models \text{pre}(a)\}$ /* Candidate next states.*/
for $s' \in C$ **do**
    if $distance(bin(s'), \hat{s}) > \vartheta$ **then**
        $C \leftarrow C - s'$
return $\arg\min_{c \in C} |h_s(c, o, G)|$ /* s' with max L_G.*/

Discriminate the correct goal (plan)

First, we discard goal hypotheses that are not in the last state of their predicted sequence.

Then, we rank the remaining goals based on their compliance with the set of observations Ω, i.e. $|\mathcal{S}_G \cap \Omega|$.

If there is a tie, we select the shortest sequence $\mathcal{S}_G$, following the notion that agents are at least approximately rational and prefer shorter plans.

Finally, if we still have a tie, we compare the cosine distance of the last state of each computed sequence with its respective goal hypothesis.

EXPERIMENTS AND RESULTS

Experiments

We use 4 domains to evaluate our approach, two standard planning domains and two image based domains.

1. Blocks,
2. Hanoi,
3. MNIST 8-Puzzle
4. Lights out

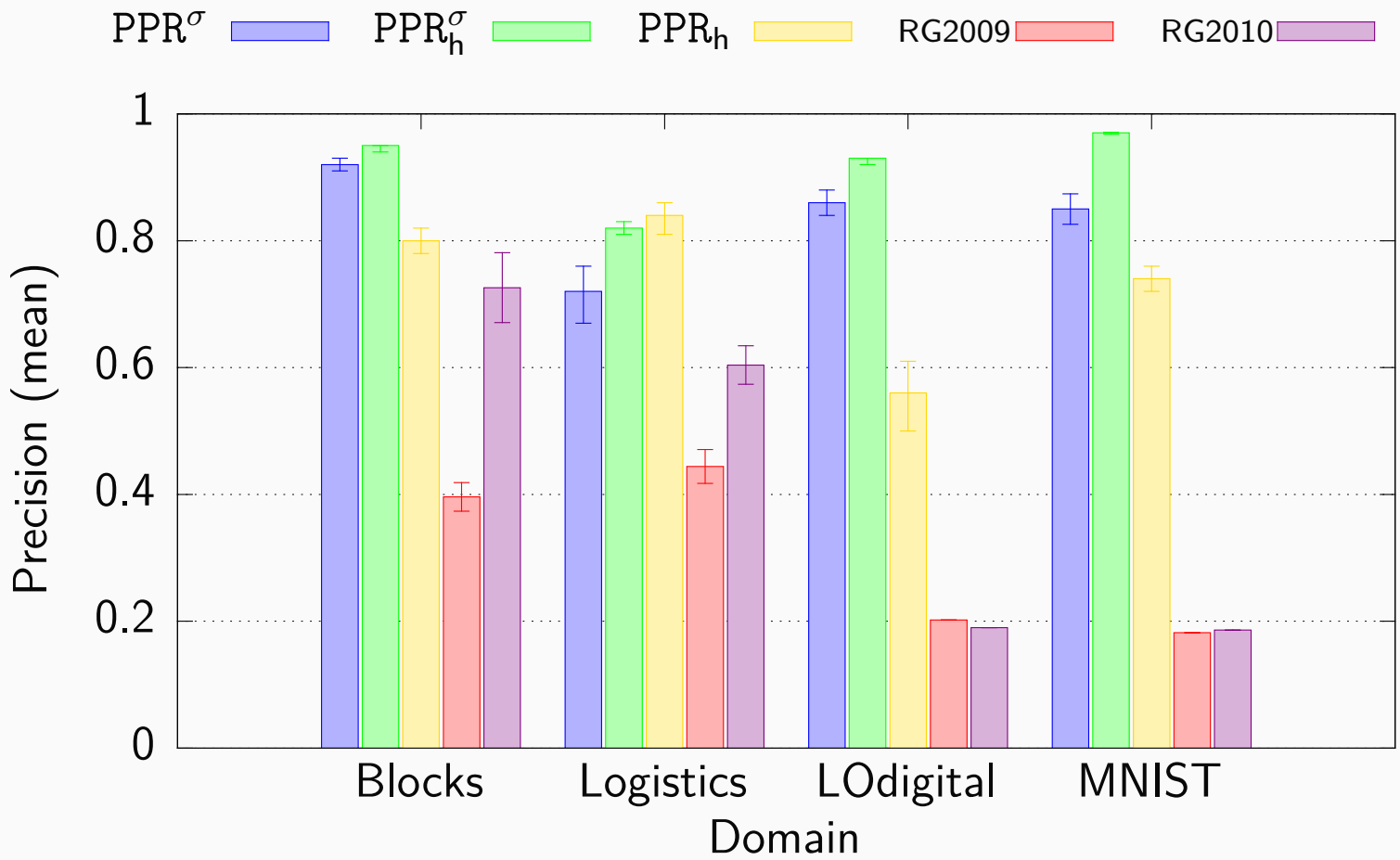
Experiments

We use the 20 planning problems from each domain test dataset to generate observation traces and create plan recognition problems.

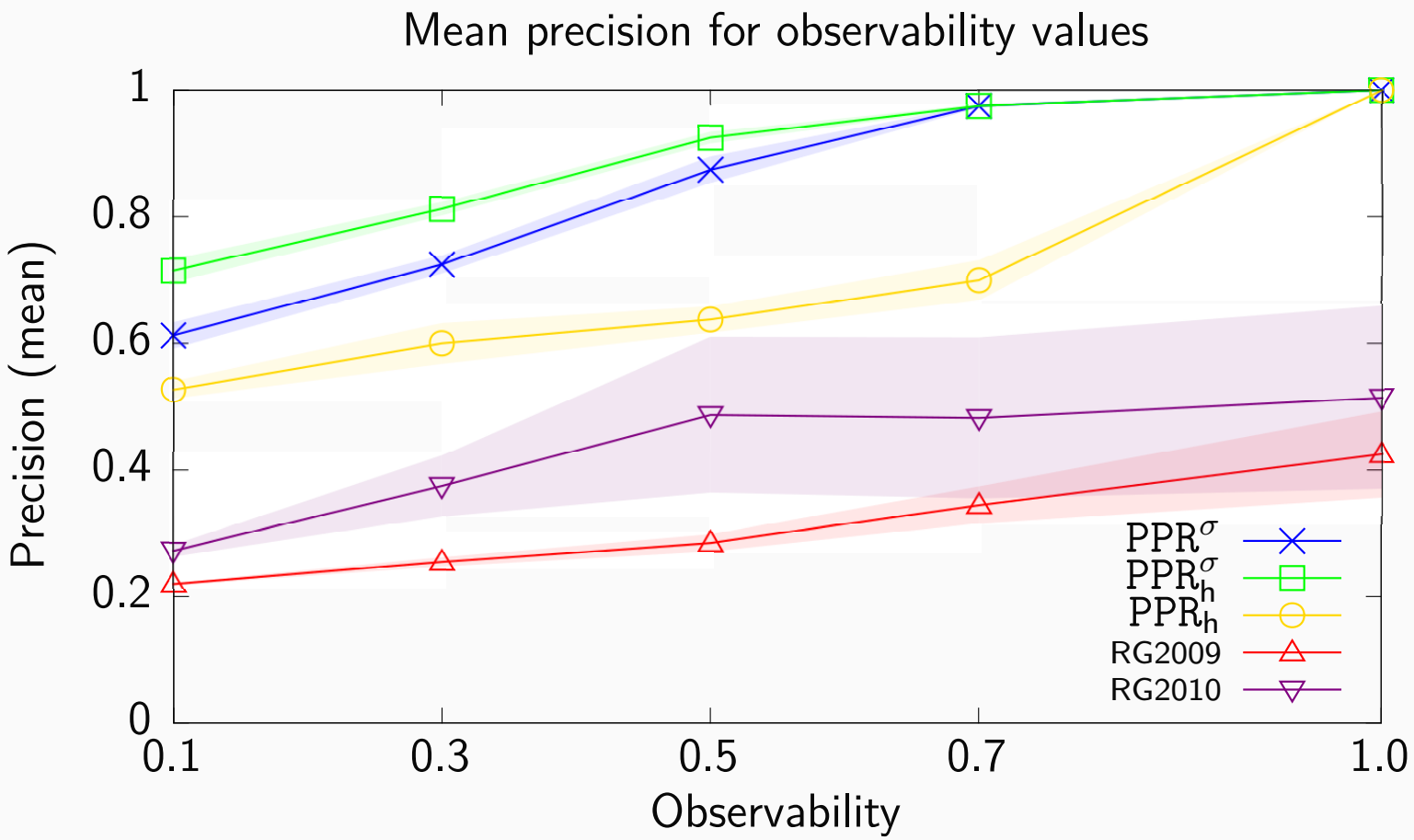
We randomly remove observations from the test plans to obtain levels of observability of 10%, 30%, 50%, and 70%, as well as full observability (100%).

We compare the variations of PPR against two seminal plan recognition approaches, RG2009 and RG2010.

Results under full observability



Results with different types of observations



CONCLUSION AND FUTURE WORK

Conclusion

The main contributions of this paper are:

A novel approach for plan recognition with very high precision both in handcrafted and automatically generated domains.

Our approach can recognize plans even when dealing with noisy observations, achieving high precision in noisy scenarios.

The predictor function can be replaced, working as a black-box. Any predictor function can be applied, such as more complex machine learning models or more sophisticated symbolic approaches.

Questions?