Planning as Goal Recognition: Deriving Heuristics from Intention Models

Giacomo Rosa[*,1], **Jean Honorio**[*], **Sebastian Sardina**[**], **Nir Lipovetzky**[*]

[*]The University of Melbourne, [**]RMIT University
Melbourne, Australia
[1]rosag@student.unimelb.edu.au

Abstract

Classical Planning seeks to find a sequence of actions, a plan, that maps a starting state into one of the goal states. If a trajectory appears to be leading to the goal, should we prioritise exploring it? Seminal work in Goal Recognition (GR) has defined GR in terms of a Classical Planning problem, adopting classical solvers and heuristics to recognise plans. We come full circle, and study the adoption and properties of GR-derived heuristics for seeking solutions to Classical Planning problems. We propose a new class of efficiently-computable heuristics and show that they improve the performance of top-scoring planners. Our work provides foundational knowledge for understanding and deriving probabilistic heuristics for Planning.

Introduction

We study the connection between goal recognition (GR) and classical planning, and how GR can be used to interpret (and develop) heuristic search in AI Planning, thus doing a full loop after the seminal work on using planning to perform GR (Ramirez and Geffner 2009, 2010). Classical planning is the field of AI that seeks to find a sequence of actions, a *plan*, that maps an initial state in a problem into a state that satisfies a specific goal condition. In classical planning, actions are deterministic, states are fully observable and represented through binary variables (*facts*), and no other actions occur outside the plan (i.e., static environment). A common strategy adopted by solvers is to perform a search over the state space, using heuristics which estimate the distance to the goal, such as the FF (Bonet and Geffner 2001; Hoffmann and Nebel 2001) and Landmark (Richter, Helmert, and Westphal 2008) heuristics, to guide the search. Other techniques have also proven to be effective in enhancing search efficiency. These include identifying *helpful actions* (Hoffmann and Nebel 2001), which prioritise operators likely to contribute to goal achievement, and leveraging *novelty* measures (Lipovetzky and Geffner 2012, 2017), which favor exploration of states exhibiting previously unseen combinations of features. While these techniques do not estimate goal distance directly, they complement heuristic search by prioritizing states that aid the search through alternative mechanisms.

The goal recognition (GR) task involves inferring an agent's goals or plans based on partial observations of its behaviour (Sukthankar et al. 2014).[1] Traditional approaches rely on a predefined *plan library*, which encodes known plans for implicit goals, allowing recognition through matching observed actions to entries in the library (Kautz and Allen 1986). More recent formulations, such as *goal recognition as planning* (Ramirez and Geffner 2009, 2010), cast recognition as a planning problem itself: a (declarative) goal is considered more likely if the observed actions align with an optimal or near-optimal plan to achieve it. Recent contributions have extended this paradigm to account for irrational behaviour of agents (Masters and Sardiña 2021), as well as adopting estimated measures and using information contained in the *effects* of observed actions to recognise goals and plans (Pereira, Oren, and Meneguzzi 2017; Wilken et al. 2024).

In light of these recent developments, we revisit search for resource-bounded agents that seek to perform an intelligent search. Due to its resource limitations, such agents bias the search towards some fragment of all possible traces (Pollack 1992; Bratman, Israel, and Pollack 1988). We analyze the intentionality of these traces with the lens of work in GR, under the intuition that some traces are seen as observations that are "more intended" towards the goal than others, and push our search algorithms to explore "more intended paths". This process is framed as a binary GR problem, where rather than estimating the intentionality of one observation towards multiple goals, we aim to assign and compare the intentionality of different observations towards a distinguished goal.

Contributions. We first present a plan-library model of GR for a resource-bounded agent in a planning domain. This allows us to define the "goal-intentionality" of observed traces, and study the properties of solvers that bias their search through such definitions. We then connect our account to previous contributions in probabilistic goal recognition through a re-definition of the $P(O \mid G)$ quantity, and derive a new formulation for the cases where observations are described through a set of independent facts. Our theory informs the definition of a new class of *intention-based heuristics* for classical planning, which we show are helpful

[1] Other common terms are Plan Recognition (PR) or Intention Recognition (IR). While suble differences exists among them, in this paper, we shall use these terms interchangeably.

in improving the state of the art in classical planning benchmarks. We tie our results to our theory, providing experimental evidence of theorised properties of our heuristics.

Preliminaries

The classical planning model is defined as $\Phi = \langle S, s_0, S_G, A, f\rangle$, where S is the discrete finite state space, $s_0 \in S$ is the initial state, $S_G \subseteq S$ is the set of goal states, A is the set of (deterministic) actions, and $f : A \times S \mapsto S$ denotes the model partial transition function, with $f(a, s)$ denoting the next state $s' \in S$ after applying action $a \in A$ in state $s \in S$. When f is undefined, the action is not applicable in the state. We write $A(s)$ to denote the set of actions applicable in state s, i.e., $A(s) = \{a \in A \mid f(a, s) \text{ is defined}\}$. A solution to a classical planning model is given by a ***plan***, a sequence of actions $\langle a_0, \ldots, a_m\rangle$ that induces a state sequence $\langle s_0, \ldots, s_{m+1}\rangle$ such that $a_i \in A(s_i)$, $s_{i+1} = f(a_i, s_i)$, and $s_{m+1} \in S_G$ for $i \in \{0, \ldots, m\}$.

A STRIPS[2] (Fikes and Nilsson 1971) problem is defined through tuple $\mathcal{P} = \langle F, A, I, G\rangle$, where F denotes the set of boolean variables, or fluents, A is the set of actions a, $I \subseteq F$ is the set of atoms that fully describe the initial state, and $G \subseteq F$ is the partial assignment that describes goal states. We assume unit cost actions in this work.

Planning Model Notation Given a Classical Planning problem $\mathcal{P}$, a ***trajectory*** denotes a sequence of alternating states and actions $\langle s_k, a_k, s_{k+1}, a_{k+1}, \ldots, s_m, a_m, s_{m+1}\rangle$, where both the first and last elements are states, such that $s_i \in S$, $a_i \in A(s_i)$, and $s_{i+1} = f(s_i, a_i)$; where $k \leq i < m + 1$. Every trajectory induces two projections: an ***a-trajectory***, which is the sequence of actions $\langle a_k, a_{k+1}, \ldots, a_m\rangle$, and an ***s-trajectory***, which is the sequence of states $\langle s_k, s_{k+1}, \ldots, s_{m+1}\rangle$. We use $\mathcal{T}(s_i, \pi)$ to represent the s-trajectory induced by an a-trajectory π applied from state s_i. We use $\mathcal{L}(s, \pi)$ to denote the last state $s_{m+1} \in S$ in the s-trajectory $\mathcal{T}(s, \pi)$. For simplicity, when s is s_0 (the initial state of $\mathcal{P}$ as per I), we just write $\mathcal{T}(\pi)$ and $\mathcal{L}(\pi)$, resp.

We place two constraints on considered trajectories: *1)* ***acyclic***: no state may appear more than once in a trajectory; and *2)* ***non-goal-extending***: goal states can only appear as the last state of the trajectory. These are reasonable assumptions, as any cycle is redundant and extending a-trajectories beyond a goal state is superfluous for the problem considered. Given $S_i, S_j \subseteq S$, we use $\Pi(S_i, S_j)$ to denote the set of *acyclic* and *non-goal-extending* a-trajectories that can be applied to a state $s_i \in S_i$ to yield a valid s-trajectory that begins at state s_i and ends at state $s_j \in S_j$. A *plan* is then an a-trajectory $\pi \in \Pi(s_0, S_G)$; in other words, an a-trajectory that, when applied to s_0, reaches a valid goal state $s_g \in S_G$. Note that more than one sequence of actions (a-trajectories) may yield the same history of states (s-trajectories). Similarly, we use ***I-reachable a-trajectory*** to refer to all a-trajectories $\pi \in \Pi(s_0, S)$. We adopt the definition of an ***observation sequence*** from previous Plan and Goal Recognition literature (Ramirez and Geffner 2009; Masters and Sardiña 2021) as *any* sequence of actions $\langle o_1, ..., o_m\rangle$, with $o_i \in A$. An action sequence ***satisfies*** an observation sequence iff it ***embeds*** it, meaning that there is a monotonic function f that maps each observation $o_i \in A$ to the index of an identical action in the action sequence such that $f(o_i) < f(o_j)$ for all $j > i$. It follows from the above definitions that, given the set $\mathcal{O}$ of all possible observation sequences for problem P, $\Pi(s_0, S_G) \subseteq \Pi(s_0, S) \subseteq \Pi(S, S) \subseteq \mathcal{O}$.

An a-trajectory π' ***contains*** a-trajectory π, written $\pi \sqsubseteq \pi'$, iff there exist, possibly empty, sequences of actions α and β such that $\pi' = \alpha \cdot \pi \cdot \beta$. This relation is reflexive, i.e., an a-trajectory contains itself. An a-trajectory π' ***extends*** an a-trajectory π, written $\pi \sqsubseteq_{\text{pfx}} \pi'$, iff there exists a, possibly empty, suffix β such that $\pi' = \pi \cdot \beta$. We define the set of ***maximal a-trajectories*** $\mathcal{M}$ as the set of I-reachable a-trajectories that are not extended by any other I-reachable a-trajectory in $\Pi(s_0, S)$:

$$\mathcal{M} = \{\pi \in \Pi(s_0, S) \mid \forall \pi' \in (\Pi(s_0, S)\backslash\{\pi\}),\ \pi \not\sqsubseteq_{\text{pfx}} \pi'\}.$$

Thus, $\Pi(s_0, S_G) \subseteq \mathcal{M} \subseteq \Pi(s_0, S)$. We also define the operator $\propto_{\text{rank}}$, which indicates that two quantities induce the same ranking (the ordering is preserved): $k \propto_{\text{rank}} l := k(x) < k(y) \iff l(x) < l(y)$.

Intention-Based Search

In this section we introduce the characteristics of our Classical Planning framework through a model that makes use of an explicit library of *sampled a-trajectories* to describe derived probabilities. Our framework imagines a Plan Recognition problem where an agent that is at the initial state s_0 can sample from the set of all maximal a-trajectories $\mathcal{M}$ starting at s_0, and performs two operations: *1)* it identifies a non-empty subset $\hat{\mathcal{M}} \subseteq \mathcal{M}$ of candidate maximal a-trajectories it may take; and *2)* it assigns a weight to every maximal a-trajectory, a measure of "preference" for that a-trajectory. Given the subset of maximal a-trajectories $\hat{\mathcal{M}}$ that we assume the agent may consider taking, the relative weight of a maximal a-trajectory $\pi^{\mathcal{M}} \in \hat{\mathcal{M}}$ indicates the probability the agent will select that a-trajectory, as opposed to other a-trajectories in $\hat{\mathcal{M}}$. Sampling a subset of all maximal a-trajectories can be viewed as the agent being resource constrained, and thus accounting for the capacity to perform operations on only a subset of all available maximal a-trajectories. This framework is a construct that explains how we derive the probabilities we use as heuristics for planning in terms of an underlying goal recognition problem.

IRPL Model

We provide a ***I-Reachable Plan-Library*** (IRPL) model that only considers I-reachable a-trajectories in a planning problem as valid observation sequences, and derives probabilities relative to an implicit library of sampled maximal a-trajectories, and the subset of those a-trajectories that constitute valid plans. This allows us to illustrate the usefulness of adopting such probabilities as "heuristic signals" in a Planning problem, under the simplified ideal scenario where such probabilities are exact estimates. Given problem description $\langle F, A, I, G\rangle$ and the set of all I-reachable

[2] Stanford Research Institute Problem Solver

a-trajectories $\Pi(s_0, S)$, an agent at initial state s_0 samples the set $\hat{\mathcal{M}}$ of maximal a-trajectories it may follow, and assigns a weight to each maximal a-trajectory according to a ***weight function*** $w : \mathcal{M} \rightarrow \mathbb{R}^+$. Let the set of sampled plans be $\hat{\mathcal{M}}_G = \Pi(s_0, S_G) \cap \hat{\mathcal{M}}$. For I-reachable a-trajectories $O \in \Pi(s_0, S)$, let $C(O) = \{\pi' \in \hat{\mathcal{M}} \mid O \sqsubseteq_{\text{pfx}} \pi'\}$ be the set of all sampled maximal a-trajectories that extend O. Let $C_G(O) = \{\pi' \in \hat{\mathcal{M}}_G \mid O \sqsubseteq_{\text{pfx}} \pi'\}$ be the set of sampled plans that extend O, and $C_{\neg G}(O) = \{\pi' \in (\hat{\mathcal{M}} \setminus \hat{\mathcal{M}}_G) \mid O \sqsubseteq_{\text{pfx}} \pi'\}$ be the maximal non-plans extending O, such that $C_G(O) \bigcup C_{\neg G}(O) = C(O)$ and $C_G(O) \bigcap C_{\neg G}(O) = \emptyset$.

We frame these sets of maximal a-trajectories π as ***events*** $\mathcal{E}$, with probability:

$$P(\mathcal{E}) = \sum_{\pi' \in \mathcal{E}} w(\pi') / \sum_{\pi'' \in \hat{\mathcal{M}}} w(\pi'') \tag{1}$$

Thus, $P(G)$ is the event that π is a plan:

$$P(G) := P(\hat{\mathcal{M}}_G) = \sum_{\pi' \in \hat{\mathcal{M}}_G} w(\pi') / \sum_{\pi'' \in \hat{\mathcal{M}}} w(\pi'') \tag{2}$$

Similarly, $P(\neg G) := P(\hat{\mathcal{M}} \setminus \hat{\mathcal{M}}_G)$.
$P(O)$ is the event that π extends O:

$$P(O) := P(C(O)) = \sum_{\pi' \in C(O)} w(\pi') / \sum_{\pi'' \in \hat{\mathcal{M}}} w(\pi'') \tag{3}$$

We can then obtain conditional probability $P(O \mid G) := P(C(O) \mid \hat{\mathcal{M}}_G)$:

$$P(O \mid G) = \sum_{\pi' \in C_G(O)} w(\pi') / \sum_{\pi'' \in \hat{\mathcal{M}}_G} w(\pi'') \tag{4}$$

where $C_G(O) = C(O) \bigcap \hat{\mathcal{M}}_G$.[3]

Finally, Bayesian posterior $P(G \mid O)$ becomes the weight of all sampled plans to the goal extending O, over the weight of all sampled maximal a-trajectories extending O:

$$P(G \mid O) = \frac{p(O \mid G)p(G)}{p(O)} = \frac{\sum_{\pi'' \in C_G(O)} w(\pi'')}{\sum_{\pi' \in C(O)} w(\pi')} \tag{5}$$

As a result, for any set of a-trajectories for which probabilities $P(O \mid G)$ and $P(G \mid O)$ are defined for all elements in the set, we can obtain a ranking through

$$\arg\max_O P(O \mid G) = \arg\max_O \sum_{\pi' \in C_G(O)} w(\pi') \tag{6}$$

$$\arg\max_O P(G \mid O) = \arg\max_O \frac{P(O \mid G)}{P(O \mid \neg G)} \tag{7}$$

where equation 7 is obtained by taking $\arg\max_O P(G \mid O)/P(\neg G \mid O)$ and noting that $P(G)$ and $P(\neg G)$ are constant when considering a single goal in planning problems.

Framework Properties

We first state the results[4], followed by analysis. When extending the domain of conditional probabilities to the set of all possible observations in a planning problem, we adopt the convention of setting undefined probabilities to 0. This reflects the perspective of a resource-bounded agent, which cannot account for trajectories it has never observed and therefore would not consider following them.

Claim 1 *Given non-empty $\hat{\mathcal{M}}$ and $\hat{\mathcal{M}}_G$, and any w, $P(O_e \mid G) \leq P(O_p \mid G)$ if O_e extends O_p.*

Claim 2 $P(O \mid G) = 0$ *and* $P(G \mid O) = 0$ *for all a-trajectories O that are not extended by any plan $\pi' \in \hat{\mathcal{M}}_G$.*

Lemma 1 *Given non-empty $\hat{\mathcal{M}}$ and $\hat{\mathcal{M}}_G$, and any w, for all a-trajectories O_e extending an a-trajectory O_p by one action,* $\max_{O_e} P(G \mid O_e) \geq P(G \mid O_p)$.

Theorem 1 *Given non-empty $\hat{\mathcal{M}}$ and $\hat{\mathcal{M}}_G$, and any w, a planner that expands* $\max P(G \mid O)$ *will find a plan in number of expansions $m \leq \max_{\pi' \in \hat{\mathcal{M}}_G} |\pi'|$.*

A maximal a-trajectory set $\hat{\mathcal{M}}$ is ***Goal-Adjacent Single-Plan*** (GASP) if every a-trajectory that is adjacent to a goal (can be extended by a single action to reach a goal state) is extended by only one plan in $\hat{\mathcal{M}}_G$.

Lemma 2 *Let $\hat{\mathcal{M}}$ be a non-empty GASP set of sampled a-trajectories, and let $|\hat{\mathcal{M}}_G| > 0$. Suppose w is a weight function such that $w(\pi) > w(\pi') \iff cost(\pi) < cost(\pi')$. A planner that expands a-trajectories in order of* $\max P(O \mid G)$ *is guaranteed to find a minimal-cost plan among all valid plans in the sample $\hat{\mathcal{M}}$. The result also holds under the weaker condition $cost(\pi) \leq cost(\pi') \Rightarrow w(\pi) \geq w(\pi')$, provided ties in $P(O \mid G)$ are broken by preferring shorter trajectories.*

Lemma 3 *Under the same conditions as Lemma 2, except without requiring $\hat{\mathcal{M}}$ to be GASP, the first expanded goal node is guaranteed to be optimal w.r.t. all plans in $\hat{\mathcal{M}}$.*

Lemma 4 *A planner that expands a-trajectories in order of* $\max P(O \mid G)$ *will expand at most* $\sum_{\pi \in \hat{\mathcal{M}}_G}(|\pi|) - |\hat{\mathcal{M}}_G|$ *nodes.*

Theorem 2 *Given non-empty $\hat{\mathcal{M}}$ and $\hat{\mathcal{M}}_G$, let $\hat{\mathcal{M}}$ be GASP. For a planner that expands a-trajectories in order of* $\max P(O \mid G)$*; as samples are added to $\hat{\mathcal{M}}$, the change in plan length found is non-increasing, and the change in worst-case number of expansions is non-decreasing.*

Assume $\hat{\mathcal{M}}_G$ may contain "mistakes": trajectories that do not in fact reach the goal. We model this by assigning each $\pi \in \hat{\mathcal{M}}_G$ a Bernoulli(γ) indicator for being a plan. Thus, in expectation, only a fraction γ of $\hat{\mathcal{M}}_G$ are plans. To capture this effect, goal–restricted weights (i.e., weights in sums over $\hat{\mathcal{M}}_G$ or $C_G(O)$) are deterministically rescaled by γ, $w^\gamma(\pi) := \gamma\, w(\pi)$. We then define $P^\gamma(O \mid G) := \sum_{\pi' \in C_G(O)} w^\gamma(\pi') \,/\, \sum_{\pi'' \in \hat{\mathcal{M}}_G} w^\gamma(\pi'')$ and $P^\gamma(G \mid O) := \sum_{\pi'' \in C_G(O)} w^\gamma(\pi'') \,/\, \sum_{\pi' \in C(O)} w(\pi')$.

[3] These probabilities are well-defined: $P(G) + P(\neg G) = 1$, and $P(O) + P(\neg O) = 1$, where $P(\neg O) := P(\hat{\mathcal{M}} \setminus C(O))$; calculating $P(O, G) = P(O \mid G) \cdot P(G)$ and using $P(\neg G)$ to obtain $P(O, \neg G)$, then $P(O, G) + P(O, \neg G) = P(O)$.

[4] Supplementary proofs are provided in the appendix.

Lemma 5 $P^{\gamma}(O \mid G) = P(O \mid G)$ *and* $P^{\gamma}(G \mid O) = \gamma \cdot P(G \mid O)$.

Theorem 3 *The* $\arg\max$ *over* O *in Equations 6 and 7 does not change when using* $P^{\gamma}(O \mid G)$ *and* $P^{\gamma}(G \mid O)$.

Remarks. We briefly summarise general properties derived from the presented theorems. Claim 2 implies that following any a-trajectory with both $P(O \mid G) > 0$ and $P(G \mid O) > 0$ is a valid strategy for reaching a goal. Theorem 1 shows that expanding nodes according to Equation 7 follows a hill climbing strategy and is strongly goal directed. In contrast, Equation 6 induces an exploratory strategy, akin to breadth first search as shorter a-trajectories tend to have higher $P(O \mid G)$ probability, as noted in Claim 1. This approach expands sampled solution trajectories until it selects a sample optimal plan. Theorem 2 reflects the sampling exploration-exploitation tradeoff: increasing the number of samples in $\hat{\mathcal{M}}$ can improve solution quality, but also increases the worst case number of expansions. Lastly, Theorem 3 shows that this expansion order remains valid even when calculating estimates incorporating a constant error rate across all sampled trajectories, hinting at further applicability to realistic imperfect sampling scenarios.

Uniform Regimes

In what follows, we introduce and analyze the properties of two "basic" weight functions. We consider these as the two general uniform weighting processes, where we assign equal probability to, respectively (1) every sampled maximal a-trajectory, (2) every action choice in state transitions.

We define a ***Uniform Maximal a-trajectory Probability*** (UMP) weight function as a weight function $w(\pi) = c$ where c is a non-zero constant, implying a uniform probability of the agent selecting any sampled maximal a-trajectory in $\hat{\mathcal{M}}$. Let us define quantities $N_T = |\hat{\mathcal{M}}|$, $N_G = |\hat{\mathcal{M}}_G|$, $N_C(O) = |C(O)|$, and $N_{CG}(O) = |C_G(O)|$.

Corollary 1 *Given non-empty* $\hat{\mathcal{M}}$ *and* $\hat{\mathcal{M}}_G$*, and a UMP weight function* w*, the probabilities obtained become* $P(O) = \frac{N_C(O)}{N_T}$, $P(G) = \frac{N_G}{N_T}$, $P(O \mid G) = \frac{N_{CG}(O)}{N_G} \propto N_{CG}(O)$, $P(G \mid O) = \frac{p(O|G)p(G)}{p(O)} = \frac{N_{CG}(O)}{N_C(O)}$.

Corollary 2 *Given any* $\hat{\mathcal{M}}$ *and a UMP weight function* w*, let each trajectory in* $\hat{\mathcal{M}}_G$ *be subject to a Bernoulli mistake probability* $1 - \gamma$*. A planner selects* $\arg\max P^{\gamma}(O \mid G)$ *expands, in expectation, the a-trajectory that is extended by the greatest number of plans (i.e., trajectories in* $\hat{\mathcal{M}}_G$ *that actually reach the goal).*

Corollaries 1 and 2 show that under a UMP weight function, $P(O \mid G)$ is proportional to the number of plans extending O, and ordering the open list by $\arg\max P(O \mid G)$ favours such prefixes. Ordering by $\arg\max P(G \mid O)$ favours prefixes with a higher ratio of plan completions to non-plan continuations. Both improve guidance toward solutions and reduce node expansions. Corollary 2 also shows that, even when assuming a uniform chance of mistakes, $P(O \mid G)$ remains unchanged and still favours traces that are expected to lead to more plans.

A ***Uniform Transition Probability*** (UTP) weight function assigns to a maximal I-reachable a-trajectory $\pi = \langle a_0, \ldots, a_{k-1} \rangle$ the product of uniform action probabilities at each step, $w(\pi) = \prod_{i=0}^{k-1} \left[|\mathrm{A}(s_i)|\right]^{-1}$, where s_0 is the initial state, $s_{i+1} = f(a_i, s_i)$, and $\mathrm{A}(s_i)$ is the set of applicable actions at state s_i. That is, at each step the agent selects an applicable action with uniform probability.

Lemma 6 *Given non-empty* $\hat{\mathcal{M}}$ *and* $\hat{\mathcal{M}}_G$*, a UTP weight function* w*, and an a-trajectory* O *that is extended by single solution plan* π_s*, the number of nodes generated to find* π_s *by a planner that expands according to* $\max P(O \mid G)$ *is lower bounded by* $-\ln\left[P(O \mid G) \cdot P(G)\right] \cdot e \propto -\ln[P(O \mid G)]$.

Theorem 4 *Given non-empty* $\hat{\mathcal{M}}$ *and* $\hat{\mathcal{M}}_G$*, a UTP weight function* w*, and a planner that expands according to* $\max P(O \mid G)$*, a lower-bound number of node generations required to achieve any plan that extends* O *is* $-\ln[P(O \mid G) \cdot P(G)] \cdot e$.

Theorem 4 shows that expanding nodes according to Equation 6 with a UTP weight function follows the a-trajectory that minimises the best case number of node generations needed to find a plan in $\hat{\mathcal{M}}$. This strategy can be seen as *optimistic in the face of uncertainty*, where uncertainty refers to unexplored regions of the state space. It assumes the subgraph extending the selected trajectory has an ideal shape; as new information is revealed, this estimate may worsen, leading the search to prefer other subgraphs.

From Probabilistic GR to Classical Planning

GR-as-Planning. Ramirez and Geffner (2010) provide a framework for probabilistic goal recognition as planning for a rational agent. They define likelihoods through a Boltzmann distribution $P(O \mid G) := \alpha \cdot exp\{-\beta c(O, G)\}$, where α is the normalizing constant and $c(O, G)$ is the optimal cost of a plan to achieve G that embeds O. Their definition is derived from a likelihood

$$P(O \mid G) = \sum_{\pi \in \mathcal{O}} P(O \mid \pi) \cdot P(\pi \mid G) \tag{8}$$

where $P(O \mid \pi)$ is 1 or 0 depending on whether π embeds O and observations O are independent of the goal given π, under the assumptions that 1) $P(\pi \mid G)$ is proportional to $e^{-\beta c(\pi)}$, where $c(\pi)$ is the cost of a plan π, and 2) the summation is dominated by the largest $P(\pi \mid G)$ term.

Probabilistic Framework for Classical Planning. We adapt Equation 8 to provide a more general definition of $P(O \mid G)$ that ties our framework to existing theory in goal-recognition-as-planning.

Let $z : \mathcal{O} \to \{0, 1\}$ represent the ***sampling function*** over all observations to the goal. We define conditional probability

$$P(\pi \mid G) := \frac{z(\pi) \cdot w(\pi)}{\sum_{\pi' \in \mathcal{O}} z(\pi') \cdot w(\pi')} \tag{9}$$

to obtain a formulation that parallels the IRPL sample-reweight process for obtaining $P(O \mid G)$.

This formulation is more general than the IRPL model described earlier, as both O and π can be any action sequence in $\mathcal{O}$, not limited to I-reachable or sequential a-trajectories. In some cases, such as the IRPL model, we may desire to restrict the set of observations considered. We thus also define functions $P_{\text{seg}}(O \mid \pi)$ and $P_{\text{pfx}}(O \mid \pi)$, with value 1 if π contains O, and π extends O, respectively. The intended domain of O and the sampling function $z(\pi)$ then determine the scope of the equation. For instance, under the IRPL assumptions with UMP weights, where $z(\pi) = 1$ if $\pi \in \Pi(s_0, S_G)$ and 0 otherwise, $P(O \mid \pi) := P_{\text{pfx}}(O \mid \pi)$ and $w(O) = 1$, we get that $\sum_{\pi' \in \mathcal{O}} z(\pi') \cdot w(\pi') = N_G$ and $\sum_{\pi \in \mathcal{O}} P(O \mid \pi) \cdot z(\pi) \cdot w(\pi) = N_{CG}(O)$.

In prior GR work, $P(\pi \mid G)$ typically assumes π is a full plan in $\Pi(s_0, S_G)$. By making the sampling function $z(\pi)$ explicit, we can extend the definition to include sequences in $\mathcal{O}$ that are not necessarily plans or goal reaching, but may still be goal directed or useful for guiding the search. This flexibility is important in planning, as it allows Equation 8 to support approximations of $P(O \mid G)$ based on partial plans or heuristic estimates.

Divergence-based Conditional Probability. Following recent work in GR (Pereira, Oren, and Meneguzzi 2017; Wilken et al. 2024), we represent observations through the facts implied by underlying action sequences or trajectories. This allows us to shift focus from the specific actions to the information conveyed by them, and alternative sequences are not penalised if they yield the same achieved facts. Let O^F represent a set of observed facts. In a GR setting, this would represent facts added by actions in action sequence O. In a planning context, it may also include facts that are true in the initial state. We define $P(O^F \mid G)$ in terms of facts $q \in O^F$ with conditional independence assumption:

$$P(O^F \mid G) = \prod_{q \in O^F} P(q \mid G) \tag{10}$$

where, adapting Eq. 8 and 9, we get the probability of observing fact $q \in O^F$ in a sampled path to the goal:

$$P(q \mid G) = \sum_{\pi \in \mathcal{O}} P(q \mid \pi) \cdot P(\pi \mid G) \tag{11}$$

We define for each fact $q_i \in O^F$ an associated Bernoulli distribution $P^i_G(q_i) := P(q_i \mid G)$, and a hard assignment $P^i_O(q_i) = 1$. Let $P_G(O^F) = \prod_{i:q_i \in O^F} P^i_G(q_i)$ and $P_O(O^F) = \prod_{i:q_i \in O^F} P^i_O(q_i) = 1$. Under these conditions, the KL divergence can be simplified to

$$D_{\text{KL}}(P_O(O^F) \parallel P_G(O^F)) = -\sum_{i:q_i \in O^F} \log P^i_G(q_i) \tag{12}$$

From Eq. 10 and 12 we thus derive

$$P(O^F \mid G) = e^{-D_{\text{KL}}(P_O(O^F) \| P_G(O^F))} \tag{13}$$

The cost based formulation of Goal Recognition (GR) (Ramirez and Geffner 2010; Masters and Sardiña 2021) is grounded in the assumption of agent rationality: observation sequences that align with near optimal plans are considered evidence of goal intended behaviour. In a planning context, this motivates prioritising sequences that appear to belong to near optimal solution plans. Similarly, Equation 13 interprets rationality in terms of the divergence of observations O^F from the distribution $P_G(O^F)$, which captures statistical evidence of goal intendedness from estimated solution paths. In the GR setting, it is intuitive that observations consistent with goal intended behaviour, when normalised by observation likelihood, are more likely to reflect intent toward that goal. In our experiments, we show that prioritizing trajectories with high $P(O^F \mid G)$, those that reflect unnormalised known goal intended patterns, provides a signal for efficient state space exploration and traversal.

Unlike cost based formulations, our divergence metric $D_{\text{KL}}(P_O(O^F) \parallel P_G(O^F))$ abstracts away action order as a consequence of the conditional independence assumption. This enables a concise theoretical model, where goal intention is described in terms of Bernoulli distributions over facts, and supports efficient computation. This divergence based formulation can be seen as a generalization of $P(O \mid G)$ in Plan Library GR (and by extension the IRPL model) over facts: rather than checking the sample plan membership of observations, it measures how closely the information in observations aligns with that in sampled plans.

Planning Heuristics

We devise two heuristics inspired by our definitions of $P(O \mid G)$ and $P(O^F \mid G)$, which share the characteristic of an initial, more costly information extraction step, to estimate goal-intended traces before beginning the search, followed by efficient online probability evaluations.

Cost-based Conditional Probability

Starting from Equations 8 and 9, we implicitly redefine the sampling and weighting function in terms of a single estimate y; $z(\pi) \cdot w(\pi) := y(\pi)$. Following the assumptions on the behaviour of a rational agent in (Ramirez and Geffner 2010) described in the previous section, we define $y(\pi) := exp\{-\hat{c}(O, G)\}$. We estimate cost $\hat{c}(O, G) = |O| + h^{ff}_t(\mathcal{T}(O))$, where $h^{ff}_t(\mathcal{T}(O))$ is a *trajectory-based variant* of the h^{ff} heuristic (Hoffmann and Nebel 2001), then derive a ranking heuristic from probability:

$$P(O \mid G) \propto_{\text{rank}} e^{-\hat{c}(O,G)} \propto_{\text{rank}} -\hat{c}(O, G) \tag{14}$$

The heuristic $h^{ff}_t(\mathcal{T}(O))$ is computed in two stages. First, during an initial information extraction phase, we construct the relaxed planning graph (RPG) from the initial state and compute h^{ff} by performing a backward search from the goals over the RPG to extract a greedy relaxed plan. From this process, we store the subgraph that includes only the facts selected in the relaxed plan and their *best supporter* actions (Keyder and Geffner 2008). We refer to this stored subgraph as the ***ff-graph***. During search, rather than recomputing h^{ff} at each state, we derive a relaxed state by marking as achieved all facts made true along the current trajectory from the initial state. Then, starting from the goal, we perform a backward traversal over the stored ff-graph to estimate the length of a relaxed plan. The traversal stops at facts already achieved, thereby skipping redundant segments of the relaxed plan. This avoids counting actions whose effects have already been achieved or made redundant by the relaxed state, and results in a value of h_t^{ff} that is less than or equal to the root h^{ff} computation value.

While h_t^{ff} may be less informed than h^{ff}, as it does not reflect deletes or changes in the RPG structure, it is significantly faster and preserves the non-increasing $P(O \mid G)$ property from Claim 1. The estimate $\hat{c}(O, G)$ may still break this property due to adding $|O|$, however this increase is at most one from parent to child node. We refer to this heuristic as ***plan-cost probabilities*** (*pcp*).

Fact Observation based Conditional Probability

Our second set of heuristics are derived from Equation 13. At the initial state, we calculate ***fact observation probabilities*** as described in (Wilken et al. 2024) to estimate the set of Bernoulli distributions $P_G^i(q_i)$ for all $q_i \in F$. Fact observation probability estimation first samples a set of *delete-relaxed* plans to the goal, then evaluates the probability of observing a fluent q in a sampled relaxed plan to the goal. Such process follows closely the sample-weighting process described in the IRPL model, albeit in the delete-relaxed state space and observing facts instead of actions to obtain $P(q \mid G)$ rather than $P(O \mid G)$, as in Eq. 11. We thus describe this estimation process in terms of $P_{\text{rel}}(q \mid G)$, the conditional probability *in the relaxed problem*, and set $P_G^i(q_i) := P_{\text{rel}}(q_i \mid G)$. We then can define heuristic

$$P(O^F \mid G) \propto_{\text{rank}} \sum_{q \in O^F} \log P_{\text{rel}}(q \mid G) \tag{15}$$

where we set O^F to be all facts observed in a given trajectory. The rank of trajectories evaluated by the heuristic is thus proportional to the KL divergence between the observed trajectory and the estimated distributions of goal-intended information $P_G^i(O^F)$, from Eq. 13. Given all $P_G^i(q_i)$, $P(O^F \mid G)$ is computed in time linear in $|O^F|$.

We refer to this heuristic as ***relaxed plan observation probabilities*** (*rpop*). In our experiments, we set the number of sampled relaxed plans to 100. This value was selected through analysis of results in the context of GR in (Wilken et al. 2024), and empirical testing.

Experimental Results

Experimental Setup

We run our experiments on a VM with an AMD EPYC 7763 processor, with each test running on a single core. We adopt Downward Lab's experiment module (Seipp et al. 2017), whereas our proposed solvers and heuristics are implemented in C++ using the LAPKT planning module (Ramirez et al. 2015). Our adopted branch of LAPKT uses the FD grounder (Helmert 2009), with the exception of problems that produce axioms, which are not currently supported in LAPKT. In such problems, our planners automatically fall-back to the Tarski grounder (Francés, Ramirez, and Collaborators 2018; Singh et al. 2021b). All experiments are limited to 1800 seconds and 8 *GB* time and memory constraints, following the satisficing track of the *International Planning Competition* (IPC) (Taitler et al. 2024). The problem set is composed of all IPC satisficing track benchmarks, selecting the latest problem sets for recurrent domains.

Improving BFWS Solvers

Given the exploratory behaviour induced by $P(O \mid G)$ outlined in our theoretical analysis, we integrate our proposed heuristics with a BFWS solver (Lipovetzky and Geffner 2017), which balances exploration of the search space and exploitation of heuristics, to evaluate improvements in its exploratory behaviour. Table 1 compares the performance of BFWS($f5$) (Lipovetzky and Geffner 2017) and BFWS$_t$($f5$) (Rosa and Lipovetzky 2024) with that of new variants that adopt *pcp* and *rpop$_r$* as third tie-breaking heuristics.

RPOP-Restart. BFWS($f5$) uses *Partitioned Novelty* (Lipovetzky and Geffner 2017) to partition each planning problem into multiple sub-problems, and the *goal-count heuristic* h^{GC}, that counts the number of unachieved goal facts, is used both to inform such partitioning, and as tie-breaking heuristic. We adapt *rpop* to this planner through ***rpop-restart*** (rpop$_r$). The difference between *rpop* and *rpop$_r$* is that the latter only adds log probabilities from facts that have been observed in the trajectory from the last state that improved h^{GC}, as opposed to all facts achieved from the start. By "restarting" at the most recent h^{GC} improvement, it regains informedness in the cases where it was lost. Otherwise, if a fact is observed that did not appear in any sampled relaxed plan to the goal, it would strongly penalise the probability of all descendant nodes. This helps account for inaccuracies in fact occurrence estimates introduced by adopting distributions derived from relaxed plans.

Basic BFWS Variants. Table 1 highlights performance gains from our proposed techniques. Both *pcp* and *rpop$_r$* increase coverage and reduce the average number of node expansions, indicating enhanced informedness. Overall, *rpop$_r$* achieves stronger results, partly due to its faster computation, which allows BFWS to find solutions more quickly. Although *pcp* also guides search more effectively, its slower evaluation time diminishes its impact on solution time and coverage. Nonetheless, its preference for shorter estimated plans leads to lower average plan costs. The median time across all benchmark problems for the information extraction phase is 56ms for the *pcp* heuristic, and 9ms for the *rpop$_r$* heuristic. The overhead from this phase thus has little impact on planning times in most problems.

SOTA Variants. Our best performing variants integrate *rpop$_r$* as the third tie-breaker in the BFWS$_t$($f5$)-Landmarks solver. BFWS$_t$($f5$)-Landmarks substitutes the h^{GC} heuristic in BFWS$_t$($f5$) with the ***Landmarks heuristic*** (h^{LM}) (Richter, Helmert, and Westphal 2008). *Rpop$_r$-UTP* further modifies trajectory weighting, to highlight the practical impact of the weighting scheme in our theoretical model.

RPOP$_r$-UTP. When calculating the proportion of sampled relaxed plans in which a given fact occurs for *rpop*, each sampled relaxed plan is given an equal weight. *rpop$_r$-UTP* (Uniform Transition Probability) re-weights sampled relaxed plans according to a UTP weight function, giving higher importance to plans with greater UTP values.

	BFWS			BFWS$_t$		
Domain	$f5$	$f5$-pcp	$f5$-rpop$_r$	$f5$	$f5$-pcp	$f5$-rpop$_r$
Coverage (1831)	1510	1526	**1560 (5.03)**	1557 (3.67)	1558 (2.55)	**1599 (3.81)**
% Score	76.77%	77.63%	**80.20% (0.35)**	79.90% (0.24)	79.99% (0.19)	**82.94% (0.32)**
N Fewest Expansions	267	739	**764**	285	750	**781**
N Min. Time	348	261	**985**	475	239	**917**
N Min. Plan Cost	867	**1009**	649	899	**1020**	651
Avg. EpS	40667	23019	36182	42421	22461	36569

Table 1: Performance comparison of BFWS($f5$) and BFWS$_t$($f5$) vs. variants with *pcp* and *rpop$_r$* heuristics. ***% score*** is the average of the % of instances solved in each problem domain. Results for solvers with a randomised component represent the mean, and include the standard deviation across 5 measurements. *N Fewest Expansions, N Min. Time* and *N Min. Plan Cost* represent the number of solved problems where a variant scores best in the respective metrics, including ties.[5] Results on BFWS($f5$) and BFWS$_t$($f5$) are compared separately. Avg. EpS represents the average number of expansions per second across problems solved by all six planners. Results indicate that *pcp* and *rpop$_r$* reduce the number of expansions across a significant portion of problems, and *rpop$_r$* improves both coverage and solution time across a sizeable portion of problems. *pcp* negatively impacts solution time but benefits plan costs compared to the baseline.

Planner	**Coverage (1831)**	**% score**	**Agile score**
Dual-BFWS	1607	83.6%	1200.8
ApxNoveltyT	1611 (3.5)	83.8% (0.2)	1233.7 (0.2)
LAMA	1535	79.1%	1192.3
Scorpion-M	1591	82.9%	1206.4
RPOP$_r$	1621 (3.2)	84.6% (0.3)	1229.4 (3.4)
RPOP$_r$-UTP	1616 (2.1)	84.2% (0.1)	**1236.0 (1.9)**
BFNoS-Dual	1641 (0.6)	86.2% (0.1)	1173.3 (3.5)
RPOP$_r$-Dual	**1655 (1.5)**	**87.0% (0.1)**	1232.6 (2.9)

Table 2: Mean coverage and ***Agile score***[6] of our proposed planners. RPOP$_r$ and RPOP$_r$-UTP refer to BFWS$_t$-$f5$-Landmarks adopting the respective heuristics. RPOP$_r$-Dual is a modification of BFNoS-Dual that replaces the BFNoS frontend with RPOP$_r$. Our enhancements of BFWS($f5$) outperform SoTA planners, on average solving more problems, more quickly, and without resorting to multiple open lists or multiple runs. RPOP$_r$-Dual builds on these results to serve as an improved frontend solver for dual-strategy planners.

Results. Table 2 compares our best-performing variants with multiple SoTA Novelty planners *BFNoS-Dual* (Rosa and Lipovetzky 2024), *Dual-BFWS* (Lipovetzky and Geffner 2017), and *Approximate Novelty Tarski* (Singh et al. 2021a), as well as past IPC satisficing track winners *LAMA* (Richter and Westphal 2010) and *Scorpion-Maidu* (Corrêa et al. 2023). Results indicate improved coverage of our proposed modifications of BFWS($f5$). *rpop$_r$-UTP* also improves Agile score over the base *rpop$_r$*, at the cost of some problem coverage. Agile score is a performance metric that jointly evaluates coverage and runtime.

Such improvements can be achieved through the use of a single combination of heuristics, without the need to resort to multiple open lists or multiple runs, unlike benchmark planners that all adopt a combination of complementary open lists or solvers. This characteristic is useful both for studying solver behaviour, as well as for serving as a component of improved satisficing and agile planners. We demonstrate this by replacing the frontend in a high-coverage dual strategy solver — BFNoS-Dual, which combines a BFNoS frontend with the backend of Dual-BFWS — with BFWS$_t$($f5$)-Landmarks-RPOP$_r$ (*RPOP$_r$-Dual* in Table 2), resulting in improved coverage and Agile score.

We further note correlation between our experimental results for *rpop$_r$-UTP* and Theorem 4, suggesting that prioritising sampled relaxed plans with better estimated bounds on the number of node generations to find the goal can accelerate the search, albeit at the expense of problem coverage. UTP weights promote a more "committed" search, whereby a greater focus on following optimistic relaxed plan samples can solve problems earlier when these estimates are accurate, but also mislead the search when they are not. The latter case may occur when sampled relaxed plans diverge from correct plans, potentially missing important actions or facts.

Concluding Remarks

Our proposed planning-as-goal-recognition theoretical framework offers a new perspective on heuristic search, interpreting evaluation functions as processes that infer the intention of trajectories to the current state with respect to the goal. Such trajectories, therefore, do not only reveal the cost so far, but also their goal intendedness. Our proposed heuristics improve the base performance of BFWS, adopting a computationally cheap information extraction phase that allows the solvers to match complex IPC planners in Agile scores, while exceeding their coverage. The one-off time cost of the information extraction phase opens the door to potentially more informed and expensive estimation methods to further improve problem solving capability. The probabilistic nature of our technique can also lead to new solutions in related problems, such as learned heuristics for Planning, and facilitate connections between traditional Planning and non-symbolic yet planning-inspired solutions, such as Tree-of-Thoughts reasoning techniques (Yao et al. 2023) adopted by large language models.

[5] Using seed 0 for variants with a random component.

[6] ***Agile score*** is 1 for problems solved in $T \leq 1s$, and $1 - \frac{\log(T)}{\log(300)}$ for $1 < T \leq 300$.

Acknowledgments

We thank the reviewers for their valuable feedback. This research was supported by use of the Nectar Research Cloud and by the Melbourne Research Cloud. The Nectar Research Cloud is a collaborative Australian research platform supported by the NCRIS-funded Australian Research Data Commons (ARDC).

References

Bonet, B.; and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence*, 129(1-2): 5–33.

Bratman, M. E.; Israel, D. J.; and Pollack, M. E. 1988. Plans and Resource-Bounded Practical Reasoning. *COMPINT*, 4(3): 349–355.

Corrêa, A. B.; Francès, G.; Hecher, M.; Longo, D. M.; and Seipp, J. 2023. Scorpion Maidu: Width Search in the Scorpion Planning System. In *Tenth International Planning Competition (IPC-10): Planner Abstracts*.

Fikes, R. E.; and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4): 189–208.

Francés, G.; Ramirez, M.; and Collaborators. 2018. Tarski: An AI Planning Modeling Framework. https://github.com/aig-upf/tarski.

Helmert, M. 2009. Concise finite-domain representations for PDDL planning tasks. *Artificial Intelligence*, 173(5-6): 503–535.

Hoffmann, J.; and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, 14: 253–302.

Kautz, H. A.; and Allen, J. F. 1986. Generalized plan recognition. In *AAAI*, 32–37.

Keyder, E.; and Geffner, H. 2008. Heuristics for planning with action costs revisited. In *ECAI 2008*, 588–592. IOS Press.

Lipovetzky, N.; and Geffner, H. 2012. Width and serialization of classical planning problems. In *ECAI 2012*, 540–545. IOS Press.

Lipovetzky, N.; and Geffner, H. 2017. Best-first width search: Exploration and exploitation in classical planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Masters, P.; and Sardiña, S. 2021. Expecting the unexpected: Goal recognition for rational and irrational agents. *AIJ*, 297: 103490.

Pereira, R.; Oren, N.; and Meneguzzi, F. 2017. Landmark-based heuristics for goal recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Pollack, M. E. 1992. The Uses of Plans. *AIJ*, 57(1): 43–68.

Ramirez, M.; and Geffner, H. 2009. Plan recognition as planning. In *IJCAI*, 1778–1783.

Ramirez, M.; and Geffner, H. 2010. Probabilistic plan recognition using off-the-shelf classical planners. In *AAAI*, 1121–1126.

Ramirez, M.; Lipovetzky, N.; Singh, A.; and Muise, C. 2015. Lightweight Automated Planning ToolKiT. http://lapkt.org/. Accessed: 2025.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks Revisited. In *AAAI*, volume 8, 975–982.

Richter, S.; and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39: 127–177.

Rosa, G.; and Lipovetzky, N. 2024. Count-based novelty exploration in classical planning. In *Proceedings of the European Conference on Artificial Intelligence*, volume 392, 4181–4189.

Seipp, J.; Pommerening, F.; Sievers, S.; and Helmert, M. 2017. Downward Lab.

Singh, A.; Lipovetzky, N.; Ramirez, M.; and Segovia-Aguas, J. 2021a. Approximate novelty search. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, 349–357.

Singh, A.; Lipovetzky, N.; Ramirez, M.; Segovia-Aguas, J.; and Frances, G. 2021b. Grounding Schematic Representation with GRINGO for Width-based Search.

Sukthankar, G.; Geib, C.; Bui, H. H.; Pynadath, D.; and Goldman, R. P. 2014. *Plan, activity, and intent recognition: Theory and practice*. Newnes.

Taitler, A.; Alford, R.; Espasa, J.; Behnke, G.; Fišer, D.; Gimelfarb, M.; Pommerening, F.; Sanner, S.; Scala, E.; Schreiber, D.; et al. 2024. The 2023 International Planning Competition.

Wilken, N.; Cohausz, L.; Bartelt, C.; and Stuckenschmidt, H. 2024. Fact Probability Vector Based Goal Recognition. In *ECAI 2024*, 4254–4261. IOS Press.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36: 11809–11822.

Appendix A: Supplementary Proofs

Proof of Claim 1. Follows from Equation 6, since $C_G(O_e) \subseteq C_G(O_p)$.

Proof of Claim 2. Follows from Equations 6 and 7.

Proof of Lemma 1. Every maximal a-trajectory that extends O_p also extends one and only one of its children a-trajectories. Thus, $P(G \mid O_p)$ is a weighted average of the probability of its children. $P(G \mid O_p) = \frac{p(O_p|G)p(G)}{p(O_p)} = \frac{\sum_{\pi'' \in C_G(O_p)} w(\pi'')}{\sum_{\pi' \in C(O_p)} w(\pi')}$. Each child trajectory O_e extends O_p with a single action, and each O_e extends O_p with a *different* action to all other O'_e (otherwise they would not be distinct child a-trajectories). Then, each maximal a-trajectory $\pi' \in C(O_p)$ must also extend one and only one child trajectory O_e, such that the union of sets of maximal a-trajectories $\bigcup_{O_e} \left[C(O_e)\right] = C(O_p)$, and $\sum_{O_e} |C(O_e)| = |C(O_p)|$; and $\bigcup_{O_e} \left[C_G(O_e)\right] = C_G(O_p)$, and $\sum_{O_e} |C_G(O_e)| = |C_G(O_p)|$. Let $X(O) = \sum_{\pi' \in C(O)} w(\pi')$, and $X_G(O) =$

$\sum_{\pi' \in C_G(O)} w(\pi')$. Then $X(O_p) = \sum_{O_e} X(O_e)$, and $X_G(O_p) = \sum_{O_e} X_G(O_e)$. The weighted average is $P(G \mid O_p) = \frac{X_G(O_p)}{X(O_p)} = \sum_{O_e} \left[\frac{X(O_e)}{X(O_p)} \cdot \frac{X_G(O_e)}{X(O_e)}\right]$. Proof follows from the fact that a weighted average cannot be less than all its components.

Proof of Theorem 1. The first expanded node will have $P(G \mid O)$ greater than all other nodes. Since $P(G \mid O) > 0$, from Claim 2 we know that at least one plan to the goal exists, and since from Lemma 1 its best child will have probability $\geq$ to that of its parent, it will itself be expanded next. By induction we obtain the statement.

Proof of Lemma 2. Assume for contradiction that the first goal-adjacent a-trajectory ρ expanded corresponds to a plan π that is not of minimal cost in $\hat{\mathcal{M}}_G$. Let π^* be a lower-cost plan, and let ρ^* be the unique goal-adjacent a-trajectory extended by π^* (guaranteed by GASP). Since $w(\pi^*) > w(\pi)$, and $P(\rho \mid G)$ is proportional to $w(\pi)$ for goal-adjacent a-trajectories in a GASP sample, we have that $P(\rho^* \mid G) > P(\rho \mid G)$, contradicting the assumption that ρ was expanded first.

Suppose instead that ρ^* has not yet been expanded to its goal-adjacent suffix. In this case, the current a-trajectory prefix of π^* has $P(\cdot \mid G)$ equal to the sum of weights of all plans in $\hat{\mathcal{M}}_G$ that extend it, including π^*. Since $P(\cdot \mid G)$ values decrease (or stay constant) as a-trajectory length increases, because each step partitions the supporting plans further, then the current P value of the prefix is greater than or equal to $P(\rho^* \mid G)$. Hence, if π^* has not yet reached its goal-adjacent suffix, its current prefix must still have $P(\cdot \mid G) > P(\rho \mid G)$, again contradicting the assumption that ρ was expanded first. Therefore, the first expanded goal-adjacent a-trajectory must correspond to a minimal-cost plan in $\hat{\mathcal{M}}_G$.

If $w(\pi) \geq w(\pi')$ when $\text{cost}(\pi) \leq \text{cost}(\pi')$, plans of different cost may have equal weight. Thus, their goal-adjacent a-trajectories may share the same $P(O \mid G)$ value. In such cases, we break ties by preferring shorter a-trajectories, ensuring that the first generated plan corresponds to a lowest-cost plan.

Proof of Lemma 3. Follows from Claim 1 that the first expanded plan has highest weight of all plans. The rest of the proof follows same logic as Lemma 2.

Proof of Lemma 4. Follows from Claim 2 that at most all nodes in s-trajectories that are extended by plans in $\hat{\mathcal{M}}_G$ need to be expanded.

Proof of Theorem 2. Follows from previous Lemmas 2 and 4, and considering that as samples are added, the minimum plan cost in the set can only decrease.

Proof of Lemma 5. For this proof, we derive the results in terms of formulas that directly adopt a Bernoulli indicator b_π for being a plan, showing that the deterministic rescaling and related results are equivalent. The expectation $E[b_\pi]$ governs the probability of a trajectory $\pi \in \hat{\mathcal{M}}_G$ not being a mistake, i.e., π is a plan. Following Bernoulli distribution with error $1 - \gamma$, thus $E[b_\pi] = \gamma$, we obtain expectation $E_b[w(\pi)] := w(\pi) \cdot E[b_\pi] = \gamma \cdot w(\pi)$ of non-mistake plan weight. Defining conditional probabilities that directly use non-mistake expected values, we get that

$$\begin{aligned} P^b(O \mid G) &:= \sum_{\pi' \in C_G(O)} E_b[w(\pi')] \,/ \sum_{\pi'' \in \hat{\mathcal{M}}_G} E_b[w(\pi'')] \\ &= \left[\gamma \cdot \sum_{\pi' \in C_G(O)} w(\pi')\right] \Big/ \left[\gamma \cdot \sum_{\pi'' \in \hat{\mathcal{M}}_G} w(\pi'')\right] \\ &= P(O \mid G) \end{aligned}$$

and

$$\begin{aligned} P^b(G \mid O) &:= \sum_{\pi'' \in C_G(O)} E_b[w(\pi'')] \,/ \sum_{\pi' \in C(O)} w(\pi') \\ &= \left[\gamma \cdot \sum_{\pi' \in C_G(O)} w(\pi')\right] \Big/ \sum_{\pi' \in C(O)} w(\pi') \\ &= \gamma \cdot P(G \mid O) \end{aligned}$$

It is clear that these equations are equivalent to the deterministically rescaled definitions for $P^\gamma(O \mid G)$ and $P^\gamma(G \mid O)$. Note that the denominator in the equation $P^b(G \mid O)$ does not make use of the expected value in the summation. This is because the Bernoulli random variable accounts for mistakes in *plans*. The summation over $C(O)$ in the denominator does not discount trajectories that do not reach the goal, and sums them independently of whether they are plans.

Proof of Theorem 3. Follows from Lemma 5 and Equations 6 and 7.

Proof of Corollary 1. Follows from Equations 1,3,4, and 5; setting weight function $w(\pi) = 1$, then the value of each summation is equivalent to the number of elements in the relevant sets.

Proof of Corollary 2. Follows from Theorem 3 and Corollary 1.

Proof of Lemma 6. The lower bound on nodes generated is given by the minimum possible number of nodes generated while following π_s that achieves UTP weight $w(\pi_s) = P(O \mid G) \cdot P(G)$. For each expanded state $s_i \in \mathcal{T}(\pi_s)$, the number of generated nodes increases by $|\mathrm{A}(s_i)|$, and the weight of the a-trajectory to s_i is multiplied by $\frac{1}{|\mathrm{A}(s_i)|}$. For $w(\pi_s) = \frac{1}{X}$ the minimum number of generated nodes is thus given by solving $\min \sum_{s_i \in \mathcal{T}(\pi_s)} (|\mathrm{A}(s_i)|)$ s.t. $\prod_{s_i \in \mathcal{T}(\pi_s)} |\mathrm{A}(s_i)| = X$. A lower bound to the integer solution is achieved by solving the real version of the problem, which can be solved analytically through the AM-GM Inequality to yield $e \cdot \ln(X)$.

Proof of Theorem 4. $P(O \mid G)$ is equivalent to the sum of the weight of all plans extending O, over a common denominator. From Lemma 6, it follows that the number of node generations required to solve any plan increases inversely to the plan's weight. Thus, the minimum number of node generations occurs when a single solution plan extends O.