Goal Recognition via Variational Causality

Jiaqi Wen
University of Houston
Houston, United States
jwen4@uh.edu

Leonardo Amado
University of Aberdeen
Aberdeen, United Kingdom
leonardo.amado@abdn.ac.uk

ABSTRACT

Recent advances in *Goal Recognition* have yielded a new class of approaches capable of solving goal recognition problems without relying on predefined domain theories, defined as *Model-Free Goal Recognition*. Most existing approaches rely on neural networks, probabilistic theories or approximated domain theories to recognize goals without relying on explicitly defined domain knowledge. However, these approaches often neglect the causal relationships contained in the data used for their process. This oversight overlooks an opportunity to make their goal recognition process more accurate, explainable and robust. We propose a novel *Model-Free Goal Recognition* approach that integrates causality through *Variational Inference*, which, to the best of our knowledge, is an entirely novel class of techniques for goal recognition. The method encompasses three key stages: *Causal Discovery*, *Counterfactual Inference*, and decision-making grounded in *Trajectory Likelihood*. Our approach outperforms the existing state-of-the-art methods in all tested domains. Moreover, its strong noise resilience ensures that its performance in noisy environments is nearly indistinguishable from standard conditions, fully showcasing its robustness.

KEYWORDS

Goal Recognition, *Causality*, *Variational Inference*, *Reinforcement Learning*

ACM Reference Format:
Jiaqi Wen and Leonardo Amado. 2025. Goal Recognition via Variational Causality. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

In general, the interaction between an intelligent agent and the world is goal-oriented, making *Goal Recognition* (*GR*) a crucial task to infer the agent's true intentions and goals based on a series of observations.[9] However, usual *GR* approaches require handcrafted domain theories, explaining the inner workings of the observed domain. This is not feasible for many domains, especially when working in real-world scenarios. Recently, much research has been focusing on alleviating the need for domain knowledge for *GR* approaches.This yield a new class of *GR* approaches, called *Model-Free GR* [4].

Although current *Model-Free GR* methods offer considerable flexibility, they also face significant limitations. Firstly, the black-box nature of these approaches face substantial challenges in domains where transparent decision-making is required, as they lack explainability for their decisions, as highlighted by Maynard et al. [15]. These methods rely on complex machine learning models, making it difficult for users to understand or trust the decision-making process, especially in fields like healthcare or autonomous systems, where interpretability is crucial [2, 11, 14, 26]. Secondly, there is a high demand for data. *Model-Free* methods typically require large volumes of observational data to infer goals accurately, but in real-world scenarios, data can be scarce, costly to obtain, or noisy, which limits their broader practical applicability[25].

In recent years, there has been some exploration of explainable *GR* methods, with a primary focus on maximising the interpretability of agent behaviour[12, 13]. The method based on the Weight of Evidence (*WOE*) proposed by Alshehri et al. [1] effectively measures the strength of evidence for a particular observation in supporting one goal hypothesis while opposing another.

Applying causal inference to the *GR* problem is a highly promising endeavour due to its numerous advantages. Firstly, causal inference clarifies relationships between variables, enhancing transparency and interpretability, which is crucial for improving the model's trustworthiness. Secondly, unlike neural network-based *GR* methods, causal inference establishes deeper causal structures, enhancing generalisation and ensuring robust performance across diverse contexts. In contrast, traditional neural network-based methods primarily rely on pattern recognition. Lastly, in dynamic or uncertain complex environments, causal inference can effectively identify the causal relationships between goals and environmental changes, filtering out irrelevant noise data. This allows systems to adapt flexibly to changing environments while focusing on key factors, highlighting the great potential of causal inference in *GR*.

To address these limitations, we propose a new class of *Model-Free GR* algorithms that integrate causal theory. Building on the reinforcement learning framework by Amado et al.[3], this approach uses variational inference (*VI*) to estimate counterfactual effects and construct an approximate causal graph. The causal model and *Q-table* together inform the trajectory likelihood algorithm, enabling explainable decision-making. Our main contributions are fourfold:

- **Causal Discovery**: We have redefined the concept and conditions for measuring causality within time series. Based on this novel definition, we have designed the *Causal Factor Algorithm* and the *Causal Entropy Algorithm* to construct two types of causal models: *Causal Graph* and *Causal Cube*.
- **Counterfactual Inference**: We introduced a *do-Calculus* method to implement interventions on the causal graph. By employing a *parameterised Sigmoid* function as a bridge, we utilised *VI* to approximate counterfactual effects, thereby constructing an approximate causal graph.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

- **Trajectory Likelihood**: We developed a trajectory likelihood algorithm rooted in interpretable reasoning, which integrates a causal model with a *Q-table* to facilitate estimate the most likely goal.
- **Observation Recovery**: In addressing observation that include *State-Only* and *Action-Only*, we designed two completely inverse, fair, and highly reliable observation recovery algorithms to restore the complete *state-action* observation.

2 BACKGROUND

2.1 *GR* Problem and *Model-Free GR*

The essence of *GR* lies in inferring an agent's underlying goals through observing their behaviour in an environment. Traditional *Model-Based GR* methods typically rely on explicitly predefined domain models, which details the inner workings of the environment. Accordingly, the existing literature categorises such *GR* problems under Definition 2.1[4]. However, constructing an accurate domain model often requires a domain expert, capable of buidiling correct domain knowledge, which can be both costly and time-consuming.

*Definition 2.1 (**Model-Based GR**).* A Model-Based *GR* problem $\mathcal{P}^{\mathcal{M}}_{GR} = \langle \mathcal{M}, s_I, \mathcal{G}, O \rangle$ is a tuple that comprises a model $\mathcal{M}$, representing the properties and actions of an environment, an initial state s_I, a set of possible goals $\mathcal{G}$, including the intended goal $G^* \in \mathcal{G}$ unknown to the recognizer, and a sequence of observations O that projects a sequence of interactions in an environment to achieve G^*.

To address the need for pre-defined domain knowledge, *Model-Free GR* (as defined by Amado et. al. in [4]) methods have increasingly become a focal point of research in recent years. They define two different types of *Model-Free* approaches: Model Agnostic, where these approaches recognize goals without any sort of domain knowledge, and Model Approximate ,where these approaches approximate domain knowledge from data. The method we propose belongs to the *Model-Approximate* class of *Model-Free GR* algorithms, as defined in Definition 2.2. The fundamental idea behind this class of algorithms is to approximate domain models through data (or sampling), thereby computing a substitute for the pre-defined domain knowledge. Compared to *Model-Agnostic* approaches, *Model-Approximate* algorithms are often better at explaining their decisions, as Model Agnostic approaches are often machine learning models where the decisions are concealed in the inference process.

*Definition 2.2 (**Model-Approximate GR**).* Let $\mathcal{P}^{\mathcal{M}}_{GR}$ be a *Model-Free GR* problem. A *GR* process is *Model-Approximate* if it first approximates the underlying model $\mathcal{M}$ as $\widetilde{\mathcal{M}}$. Denoted as $\mathcal{P}^{\widetilde{\mathcal{M}}}_{GR} = \langle \widetilde{\mathcal{M}}, s_I, \mathcal{G}, O \rangle$

2.2 Causality

In recent years, much research has increasingly emphasized the study of causality as a means to address the limitations inherent in correlation-based machine learning[16, 17], with research efforts primarily categorized into *Causal Discovery* and *Causal Inference*.

2.2.1 ***Causal Discovery***. Causal discovery seeks to uncover causal relationships between variables using observed data, with minimal or no interventions, and to infer how changes in cause variables affect outcome variables under unobserved conditions. These relationships are typically represented by a Directed Acyclic Graph (*DAG*), as shown in Figure 1, encompassing three fundamental causal structures[27] : (1) **Fork** - $X \leftarrow U_{XY} \rightarrow Z$, (2) **Chain** - $X \rightarrow Y \rightarrow Z$ and (3) **Collider** - $U_Z \rightarrow Z \leftarrow Y$. Here, node U represents the latent variables, while nodes X, Y, and Z are observed variables. The directed edges indicate that the causal variable influences the outcome variable.

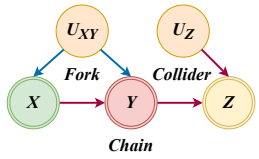

Figure 1: Example of a causal graph.

The interaction between the agent and the environment can be viewed as a time series of actions from a starter state to another state. For *GR* problems, uncovering the causal relationship between the agent's actions and the goals necessitates redefining a specific form of causality to assess the strength of causal links between events within the time series. As delineated in Definition 2.3, we derived this causality by extrapolating from the reflex arc model in biology.

*Definition 2.3 (**Conditions Of Strong Causality In A Time series**).* For a time series, a strong causal relationship between event X and event Y requires the following conditions to be met:

- **Necessary Condition 1:** Event X should specifically trigger event Y, with minimal occurrence of unrelated events Z.
- **Sufficient Condition 2:** Given Condition 1, causal event X should trigger a specific number of resultant events Y shortly after. The longer it takes for event Y to occur, the weaker the inferred causality.
- **Sufficient Condition 3:** If Condition 1 is satisfied, the frequency of causal event X should closely match that of resultant event Y.
- **Sufficient Condition 4:** If Conditions 1 and 2 are met, an increase in the occurrences of event X should strengthen the causal relationship with event Y.
- **Sufficient Condition 5:** If Conditions 1 and 2 are met, any unilateral change in the occurrences of A or B without a proportional change in the other will increase *Relative Noise* and weaken causality. The established causality is maintained only if both A and B change simultaneously.

2.2.2 ***Causal Inference***. Causal inference aims to address the question, "How will variable Y change if variable X changes?" [19] Pearl et al. argue that causal inference goes beyond analysing probabilities under static conditions, extending to the study of dynamic changes in events under varying conditions.[18] Key methods include intervention analysis and counterfactual inference. The intervention involves observing the effect on the outcome variable by externally manipulating the cause variable, a process known as the ***do-Calculus***, typically achieved by setting the cause variable to a fixed value rather than relying on its natural occurrence.

Counterfactual inference provides a deeper understanding of causal relationships by constructing and analysing hypothetical causal scenarios. In our approach, counterfactual inference relies on Variational Inference (*VI*). *VI* is an optimisation technique used to approximate complex probability distributions by substituting them with a simpler, more tractable distribution, thereby transforming the inference problem into an optimisation task.[10] The specific form of *VI* we use will be detailed in the next section.

3 PROPOSED METHOD

In this section, we present in detail our proposed approach - *Goal Recognition via Variational Causality* (*GRVC*). Figure 2 illustrates the algorithmic flow of *GRVC*.

3.1 Causal Discovery

We employed the same *Q-Learning* method as Amado et al. to sample for each goal G[3]. This algorithm explores the environment with the objective of achieving the candidate goal, terminating upon successful completion of the goal. As a reinforcement learning algorithm, it seeks to discover an optimized trajectory toward the goal, minimizing the number of steps required to reach the desired outcome. The aim is not on optimising the performance of reinforcement learning, but to gather enough sampling towards a goal to construct a causal model. We plan to construct two causal models from the sampled sequences: *Causal Cube* and the *Causal Graph*. Each sequence for a given G encompasses N paths leading to that G.

3.1.1 ***Casual Cube.*** The causal cube *(S, A, S')* is constructed for each G to evaluate the causal strength of the transition from state $s \in S$ to $s' \in S'$ after the agent's action $a \in A$, denoted as $(S, A) \rightarrow S'$. Before constructing the causal cube for a G, all relevant sampling sequences are concatenated into a chronological series. The causal intensity is quantified using a ***Causal Factor Algorithm***, as defined in Definition 2.3. The overall formula for the ***Causal Factor*** ω is derived from the three core components in Definition 2.3.

$$\omega = Distance\_Factor \times Relative\_Noise \times Consistency \tag{1}$$

- ***Distance Factor***: The distance can be seen as the response time of the resultant event $(s_k, a\_)$ to the causal event (s_i, a_j), where $a\_$ means the action after state s_k is irrelevant. As shown in Figure 3, we divide the sampled sequence into smaller intervals based on (s_i, a_j) to calculate distance values within each interval. There are three possible cases based on the sequential distance between (s_i, a_j) and $(s_k, a\_)$, temporarily referred to as X and Y, respectively.
  - **Situation 1:** After X occurs, Y is triggered between X and the next X.

$$Distance = index(nearest\_Y) - index(X) \tag{2}$$

  - **Situation 2:** If Y was triggered before the current X but not before the next X, we apply a penalty and select the furthest Y from the current X.

$$Distance = index(X) - index(furthest\_previous\, X\_Y) \tag{3}$$

  - **Situation 3:**After X, if Y is not triggered before the next or previous X, we impose a severe penalty by selecting the earliest Y, which is the furthest from the current X.

$$Distance = index(X) - index(furthest\_previous\_Y) \tag{4}$$

$$Distance\_Factor = \frac{1}{mean(Distances)} \tag{5}$$

Where *Distances* is the set of *Distance*.

- ***Relative Noise***: The *Relative Noise* is the ratio of the number of Y triggered by X. Here's how it's calculated:

$$Relative\_Noise = \frac{count\_Y\_between\_X}{count\_X} \tag{6}$$

Here, $count\_Y\_between\_X$ is the number of Y occurrences between adjacent Xs, and $count\_X$ is the total number of Xs.

- ***Consistency***: This measures the causal consistency between X and Y. Causality holds if X and Y change proportionally in the same direction; otherwise, it diminishes.

$$Consistency = \log\left(1 + \frac{count\_X \times count\_Y\_between\_X}{Total\_Count \times count\_Y}\right) \tag{7}$$

Where $total\_count$ is the total number of steps and $count\_Y$ is the number of Y occurrences in the sampling.

Given the substantial memory demands of constructing such a cube, we have devised an optimised data structure to represent it efficiently.

$$C_G = \{[S, A, count((S, A))] : [\omega_{S'}]\} \;\; G \in \mathcal{G} \tag{8}$$

where $\omega_{S'}$ is a certain causal factor of $(S, A) \rightarrow S'$. We use median-based pruning to remove (S, A) pairs with counts below the median, reducing computational complexity.

3.1.2 ***Causal Graph.*** The causal graph measures the causal intensity between A and $\mathcal{G}$. Figure 4 illustrates the process of transforming causal cubes into a causal graph. Initially, given a causal cube (S, A, S') for a particular G, it is compressed into a causal plane (A, S') via ***Average Method***, where this plane represents the average causal intensity between an action taken in any S and the resulting S'. Subsequently, following Definition 2.3 and Necessary Condition 1, a ***Causal Entropy*****(*CE*)** algorithm is employed to compress the plane further, producing the causal graph that links A and $\mathcal{G}$. This algorithm leverages entropy to reflect the specificity of an A triggering a particular S, while accounting for causal factors. In other words, if a particular A can specifically trigger a S, then within the distribution of the average causal intensity between A and all S, this distribution tends to increase in entropy; conversely, it decreases. The algorithm ensures that the compression process rationally conveys causal information, thereby making the causal graph a trustworthy representation.

$$CE(Graph) = mean(Graph) \times (1 + Entropy(Graph)) \tag{9}$$

$$Entropy(Graph) = -\sum(prob_{distribution} \times \log(prob_{distribution})) \tag{10}$$

$$prob_{distribution} = \frac{Graph}{row\_num(Graph)} \tag{11}$$

$$A\_G\_Causal\_Graph = \left[CE(A_i\_G_j)\right]_{A_i \in A}^{G_j \in \mathcal{G}} \tag{12}$$

After computing the causal graph for each G, the individual graphs are concatenated to form the overall $A\_G\_Causal\_Graph$ for the set of goals $\mathcal{G}$.

3.2 Counterfactual Inference

Counterfactual inference aims to approximate the causal graph given the set of observations, answering the question: "How would the $A\_G\_Causal\_Graph$ change given certain observed events?"

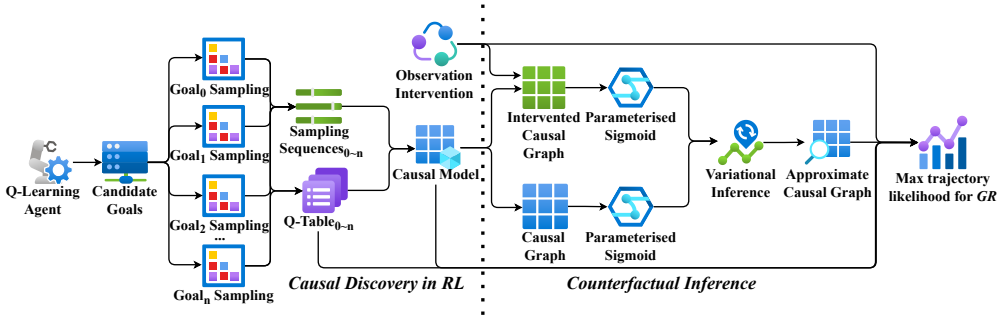

Figure 2: Algorithm workflow of *GRVC*.

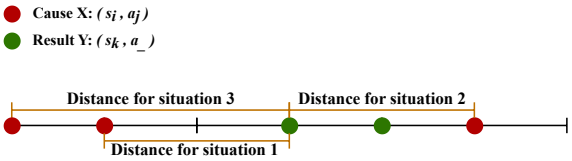

Figure 3: Three situations for calculating *Distance*.

3.2.1 ***Observational Interventions***. The primary step in counterfactual inference is intervention. We introduce a ***do-Calculus***, applying the A from the set of observations O as interventions on $A\_G\_Causal\_Graph$, resulting in an intervened causal graph $A\_G\_Causal\_Graph'$. The principle behind this intervention involves identifying the A in the O through $A\_G\_Causal\_Graph$ and, when the CE of these A exceeds a specified intervention threshold, we set them to the maximum value of their respective columns. We set intervention threshold to 0.5 because CE ranges between 0 and 1; a value exceeding 0.5 indicates that intervention is necessary. Lower thresholds leads to more aggressive intervention strategies.

$$A\_G\_Causal\_Graph' = \begin{bmatrix} CE(A_0\_G_0) & \cdots & CE(A_0\_G_n) \\ \cdots & \cdots & \cdots \\ CE(A_{O_i}\_G_0) & \cdots & CE(A_{O_i}\_G_n) \\ \cdots & \cdots & \cdots \\ CE(A_n\_G_0) & \cdots & CE(A_n\_G_n) \end{bmatrix} \tag{13}$$

do-Calculus:

$$CE(A_{O_i}\_G_j) = \begin{cases} max([CE(A_k\_G_j)]) & if\ CE(A_{O_i}\_G_j) > 0.5 \\ CE(A_{O_i}\_G_j) & else \end{cases} \tag{14}$$

3.2.2 ***Parameterised Sigmoid***. Causal theory typically requires obtaining counterfactual effects through specific methods after intervening in a causal graph. We employ Variational Inference (VI) to achieve this. Directly determining the true posterior post-intervention is a challenging task, however, we know that the $A\_G\_Causal\_Graph$ is derived from the sampling process, which is biased towards achieving a specific G^*. We assume that we are observing a rational agent, thus the set of observations O is part of an optimal plan to achieve G^*. Hence, $A\_G\_Causal\_Graph'$ reinforces the inclination towards G^*. while suppressing inclinations towards other G. Thus, $A\_G\_Causal\_Graph$ and $A\_G\_Causal\_Graph'$ are similar, allowing us to approximate the $A\_G\_Causal\_Graph$ by aligning it with $A\_G\_Causal\_Graph'$ via VI.

We devised a ***parameterised Sigmoid*** function with $\varphi(\alpha, \beta)$, which transforms causal graphs from the causal domain into the probabilistic domain, offering an inference mechanism for VI. The α modulates the gradient, while the β adjusts the horizontal displacement. Upon mapping both the original and intervened causal graph into the probabilistic space, VI approximates and optimises the optimal parameters, φ', to represent the approximate causal graph $A\_G\_Causal\_Graph_{\varphi'}$.

As illustrated in Figure 5, the $Sigmoid$ exhibits a rapid, monotonically increasing behaviour within the dashed region, referred to as the ***Mutant Interval***, while outside this region, the function approaches 0 or 1. The boundary points of the transition region are known as inflection points (represented by red or blue dots). We define the two inflection points of the ***Standard Sigmoid*** as the boundaries of the transition region: the ***Upper Inflection Point*** (UIP) is $(3, Sigmoid(3))$ which covers 95.2% of the distribution, corresponding to the statistical significance level of $p = 0.05$, while the ***Lower Inflection Point*** (LIP) is $(-3, Sigmoid(-3))$, which is symmetric to the ***Upper Inflection Point***.

- **Traverse** - β: $A\_G\_Causal\_Graph$ and $A\_G\_Causal\_Graph'$ select the average CE for each G as β because the data in the causal graph is biased. The objective is to ensure that the majority of the data is mapped to the $Mutant\ Interval$, rather than being concentrated near 0 or 1. Mapping all data close to 0 or 1 is problematic as it leads to polarization of causal relationships and the disappearance of the causal distribution curve.

$$\beta = \{mean(CE(A\_G)) \mid G \in \mathcal{G}\} \tag{15}$$

- **Slope** - α:
  - $\alpha\ LowerBound$: The Min α is the lower bound, where the $Mutation\ Interval$ of the $Sigmoid$ function is the smoothest. As shown in Figure 5, the $\alpha\ LowerBound$ (left red dot) is the value of α when the LIP is at the minimum CE (left yellow dot).
  - $\alpha\ UpperBound$: The Max α is the upper bound, where the $Mutation\ Interval$ of the $Sigmoid$ function is the steepest. The $\alpha\ UpperBound$ (left blue dot) is the value of α when the LIP is at the CE value closest to the mean(left green dot).

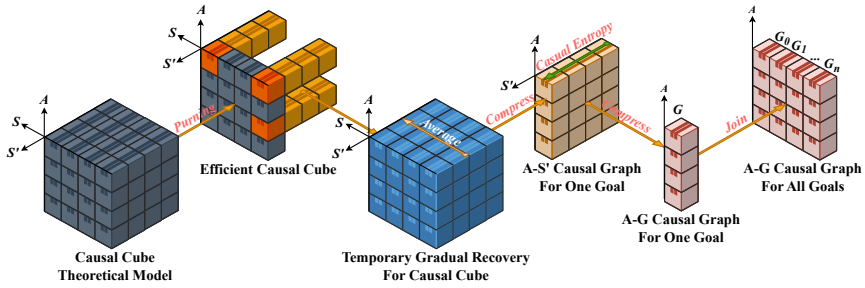

Figure 4: Workflow of transforming causal cubes into a causal graph.

Given a *LIP* and a β, the upper and lower bounds of α are determined by:

$$\alpha = -\frac{\log\left(\frac{1}{Standard\_Sigmoid(-3)} - 1\right)}{LIP - \beta} \tag{16}$$

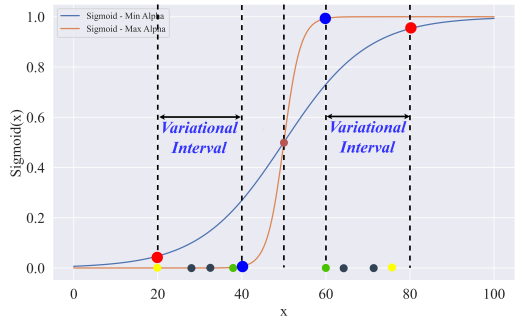

Figure 5: *Parameterised Sigmoid* and the viational internal of *VI*.

3.2.3 ***Variational Approximation***. By applying a *Sigmoid* mapping to both the original and intervention causal graphs, we use *VI* to find the optimal parameters φ' that maximise the Evidence Lower Bound (*ELBO*) thereby obtaining the $A\_G\_Causal\_Graph_{\varphi'}$. For the $A\_G\_Causal\_Graph'$, we divide the range of α into N intervals, where N is the number of *VI* iterations, typically $N = 1000$.

$$\begin{aligned}\max_{\varphi'} ELBO = \sum_{i=0}^{n} \max_{\varphi'} \mathbb{E}_{G_i \sim q(G_i \mid O, S_{\varphi'})} \left[\log\left(p(O \mid G_i, S_\varphi)\right)\right] \\ - KL\left(q(G_i \mid G, S_{\varphi'}) \| p(G_i, S_\varphi)\right)\end{aligned} \tag{17}$$

$$A\_G\_Causal\_Graph_{\varphi'} = Sigmoid(A\_G\_Causal\_Graph', \varphi') \tag{18}$$

Where:

$$S_\varphi = Sigmoid(A\_G\_Causal\_Graph, \varphi(mean(\alpha), \beta)) \tag{19}$$

$$S_{\varphi'} = Sigmoid(A\_G\_Causal\_Graph', \varphi(range(\alpha, N), \beta)) \tag{20}$$

3.3 Trajectory Likelihood

Finally, measuring Trajectory Likelihood is the final step in our *GR* process. As shown in Figure 2, this step infers the G^* from the O using the gathered information. The idea is to calculate two probabilities: the probability of each observed A at each step (called the ***Behaviour Policy*** π) and the likelihood of transitioning to the next observed S after the A (known as the ***Transition Probability*** p). Here, we introduce the basic formula for trajectory likelihood.

$$p(O \mid G_i, S_{\varphi'}) = \sum_{t=0}^{i} (\log p(s_{t+1} \mid s_t, a_t, S_{\varphi'}) \times \log \pi(a_t \mid s_t, Q_{G_i}, S_{\varphi'})) \tag{21}$$

$$Q_{G_j} \in \{Q_{G_0}, Q_{G_1}, \ldots, Q_{G_n} \mid G_j \in \mathcal{G}\} \tag{22}$$

Where Q denotes the *Q-table* learnt for each G sampling in *RL*.

- ***Behaviour Policy*** - π The purpose of the π is to calculate the likelihood of an occurrence of a time step (s_t, a_t) within O. Unlike the pseudo-policy based on *KL-Divergence* used by Amado et al.[3], we have integrated the concept of causal graph and Temperature-Softmax to propose a new policy function named the ***Causal-Temperature Softmax Policy***. For simplicity we $\pi_{Q_{G_i}, S_{\varphi'}}$ to refer to $\pi(a_t \mid s_t, Q_{G_i}, S_{\varphi'})$:

$$\pi_{Q_{G_i}, S_{\varphi'}} = \frac{\exp\left(\frac{Q_{G_i}(s_t, a_t) - \max(Q_{G_i}(s_t, A))}{\tau}\right) \cdot p(a_t \mid G_i, S_{\varphi'})}{\sum_{a' \in A} \exp\left(\frac{Q_{G_i}(s_t, a') - \max(Q_{G_i}(s_t, A))}{\tau}\right) \cdot p(a' \mid G_i, S_{\varphi'})} \tag{23}$$

$$\tau = len(Q_{G_i}(s_t, a')) \tag{24}$$

The Causal-Temperature Softmax Policy guides the next A using both the *Q-table* and causal graphs. In this policy, the numerator represents a specific (s_t, a_t) from the O, while the denominator is the sum of *Q-values* for all A in that S as per *Q-table*. Given the *Q-table* is often sparse, we introduce the *Softmax* strategy to accommodate this sparsity. We apply the natural exponential function to the *Softmax* outputs, leveraging its monotonicity and non-negativity over the real number range to handle *Q-values*. The temperature coefficient τ is used to adjust the strategy's dependence on the *Q-table*: a smaller τ increases dependence on the *Q-table*. Here, τ is set to the number of *Q-values* for a given state in the *Q-table* which helps to flexibly adapt the temperature values required for different problems.

In computing π, querying the *Q-value* for the A based on the S from the O is essential. However, noise in the O or unvisited states during sampling can lead to observed *Q-values* without any computed value in the *Q-table*. To address this, we introduced a penalty measure for π, calculated as follows:

$$penalty_\pi = \left(\frac{mean(Sigmoid(Q_{G_i}))}{\pi\_skip\_n + total\_step\_n}\right)^{\left(\frac{\pi\_skip\_n + total\_step\_n + p\_skip\_n}{exponential\_weighted\_mean(S_{\varphi'}\_G_i)}\right)} \tag{25}$$

For the design of *Penalty* π, our core idea is as follows:

– ***Severe Priority***: We set the penalty for π to be significantly harsher than that for p because π underpins the trajectory. If

Algorithm 1: Exponentially Weighted Mean

Function INPUT(*array, intervention_threshold*):
  $weights \leftarrow \exp(array - intervention\_threshold)$;
  $weighted\_sum \leftarrow \sum_{i=1}^{n} array_i \times weights_i$;
  $total\_weight \leftarrow \sum_{i=1}^{n} weights_i$;
  if $total\_weight \neq 0$ **then**
    $weighted\_average \leftarrow \frac{weighted\_sum}{total\_weight}$;
  else
    $weighted\_average \leftarrow 0$;
  return $\min(\max(weighted\_average, 0), 1)$;

π fails or is 0, it means the p is also unachieved. Therefore, we use an exponential function with a base between 0 and 1 to ensure a severe penalty.

- ***Convergence Priority***: We design the penalty to increase progressively with π's failures by incorporating $\pi\_skip\_n + p\_skip\_n + total\_step\_n$ in the exponent of the exponential function. The term $\pi\_skip\_n + p\_skip\_n$ ensures π diminishes more rapidly than p in a monotonically decreasing function. Including $total\_step\_n$ provides varying initial penalty values based on length of O, with longer sequences yielding smaller initial values.
- ***Numerical Magnitude***: To further diminish the inherently low magnitude of π, we impose a penalty by calculating the ratio of ***Exponential Weighted Mean***(as shown in Algorithm 1) to the exponential function. This division of an integer by a fraction less than one significantly accelerates the monotonic reduction of the exponential function.
- ***Causality***: G with higher $S_{\varphi'}$ receives lighter penalties compared to others with the same penalty step.

- ***Transition Probability*** - p The formula for calculating the p is as follows. Its meaning is quite evident, as it attempts to compute the causal transition probability from $(s_t, a_t) \rightarrow s_{t+1}$. For $p(a_t \mid G_i)$, we can obtain it by querying the $S_{\varphi'}$. However, for $p(s_{t+1} \mid s_t, a_t)$, the causal graph does not include state information, and the *Q-table* cannot directly provide information on transition probabilities. Therefore, we consider using the causal cube C_{G_i}.

$$p(s_{t+1} \mid s_t, a_t, S_{\varphi'}) = C_{G_i}(s_{t+1} \mid s_t, a_t) \cdot p(a_t \mid G_i) \tag{26}$$

Similarly, due to the necessity of locating states within the causal cube, the calculation of p might still be infeasible due to query failures. Consequently, we have designed a penalty measure akin to π to replace the standard calculation when it fails or equals 0. Additionally, a failure in the calculation of π will also trigger a penalty for p. The formula is as follows:

$$penalty_p = \left(\frac{mean(Sigmoid(Q_{G_i}))}{p\_skip\_n + total\_step\_n}\right)^{\left(\frac{total\_step\_n + p\_skip\_n}{exponential\_weighted\_mean(S_{\varphi'\_G_i})}\right)} \tag{27}$$

Ultimately, we obtain the optimal goal G^* by maximising the *Trajectory Likelihood*.

$$G^* = \arg\max_{G_i \in \mathcal{G}} p(O \mid G_i, S_{\varphi'}) \tag{28}$$

3.4 Observation Recovery

Currently, the approach we have described only works when having observations as a pair of states and actions (state-action pair). Given that traditional *GR* approaches typically rely solely on actions (or states), we aim to minimise structural adjustments to *GRVC* by focusing on reconstructing the observation set O. To this end, we propose two fully complementary ***Observation Recovery Algorithms***.

As illustrated in Figure 6, the first step involves obtaining a *Fused Q-table* and a *Fused Causal Graph*. The *Fused Q-table* is derived by consolidating the *Q-tables* from $\mathcal{G}$. The *Fused Causal Graph* is obtained by merging the sampling sequences from $\mathcal{G}$ into a single unified sequence and then applying a ***Causal Discovery Algorithm***. It is important to note that the resulting *Fused Q-table* and *Fused Causal Graph* do not favour any specific G but incorporate noise from other G. Nonetheless, this approach is sufficient for reconstructing a complete O that approximates the original. During the action recovery process, given a state s_n, we query the *Fused Q-table* to obtain the Q-values for all A at s_n and map these values to probabilities using a *Standard Sigmoid* function, reflecting the expected values of (S, A). Concurrently, we query the *Fused Causal Graph* to retrieve the causal factor ω for all A given state s_{n+1} and map these factors to probabilities using a *Sandard Sigmoid* function. In the case of a query failure, we proceed directly to a penalty process, which is independent of the O.

Then, We calculate the product of the probability matrices for (S, A) and (A, S), selecting the A with the highest product as the optimal choice. We consider action selection from both directions because the O forms a chain, and focusing solely on one direction may lead to an incomplete or unreachable observation sequence.

The process of recovering states is the inverse of the action recovery process. This symmetry is crucial in ensuring that both algorithms remain fair during the restoration process, preventing bias that might arise if one algorithm produces superior restoration outcomes compared to the other as far as possible.

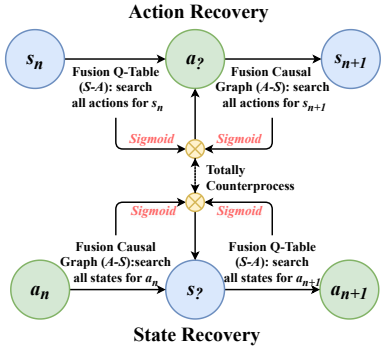

Figure 6: *Observation Recovery Algorithm.*

4 EXPERIMENTAL EVALUATION

4.1 Domains & Hyperparameters

In this section, we detail a thorough evaluation of the performance of *GRVC* across various domains, offering a comparison with other methods. Given the similar nature of our approach, we use the same three *GR* evaluation domains—Blocks, Hanoi, and SKGrid—used by Amado et al[3]. For each domain, we provide five noise-free variants

Obs.(%)	Domain	Accuracy			Precision			Recall			F-Score		
		VC	MU	KL	VC	MU	KL	VC	MU	KL	VC	MU	KL
10	Blocks	**0.85**	0.80	0.75	**0.90**	0.56	0.50	**0.90**	**0.90**	0.80	**0.90**	0.69	0.62
	Hanoi	0.90	**0.95**	**0.95**	0.82	**0.90**	**0.90**	**0.90**	**0.90**	**0.90**	0.86	**0.90**	**0.90**
	SKGrid	**0.97**	0.80	0.93	**1.00**	0.60	0.86	**1.00**	0.60	0.86	**1.00**	0.60	0.86
30	Blocks	**0.90**	0.85	0.85	**1.00**	0.62	0.62	**1.00**	**1.00**	**1.00**	**1.00**	0.77	0.77
	Hanoi	**1.00**	0.95	0.95	**1.00**	0.90	0.90	**1.00**	0.90	0.90	**1.00**	0.90	0.90
	SKGrid	**1.00**	0.85	0.90	**1.00**	0.70	0.80	**1.00**	0.70	0.80	**1.00**	0.70	0.80
50	Blocks	**0.95**	0.85	0.85	**1.00**	0.62	0.62	**1.00**	**1.00**	**1.00**	**1.00**	0.77	0.77
	Hanoi	0.95	**1.00**	0.90	**1.00**	**1.00**	0.80	**1.00**	**1.00**	0.80	**1.00**	**1.00**	0.80
	SKGrid	0.88	0.85	**0.95**	**0.90**	0.60	**0.90**	**0.90**	0.60	**0.90**	**0.90**	0.60	**0.90**
70	Blocks	**0.93**	0.85	0.85	**1.00**	0.62	0.62	**1.00**	**1.00**	**1.00**	**1.00**	0.77	0.77
	Hanoi	**0.97**	0.95	0.90	**1.00**	0.90	0.80	**1.00**	0.90	0.80	**1.00**	0.90	0.80
	SKGrid	**0.90**	0.85	**0.90**	**0.90**	0.70	0.80	**0.90**	0.70	0.80	**0.90**	0.70	0.80
100	Blocks	**0.93**	0.85	0.85	**1.00**	0.62	0.62	**1.00**	**1.00**	**1.00**	**1.00**	0.77	0.77
	Hanoi	0.97	**1.00**	0.95	**1.00**	**1.00**	0.90	**1.00**	**1.00**	0.90	**1.00**	**1.00**	0.90
	SKGrid	**0.95**	**0.95**	0.85	**1.00**	0.90	0.70	**1.00**	0.90	0.70	**1.00**	0.90	0.70
Avg	Blocks	**0.91**	0.84	0.83	**0.98**	0.61	0.60	**0.98**	**0.98**	0.96	**0.98**	0.75	0.74
	Hanoi	0.96	**0.97**	0.93	**0.96**	0.94	0.86	**0.98**	0.94	0.86	**0.97**	0.94	0.86
	SKGrid	**0.94**	0.85	0.89	**0.96**	0.70	0.78	**0.96**	0.70	0.78	**0.96**	0.70	0.78

Table 1: General performance comparison (Action-State)

Obs.(%)	Domain	Accuracy			Precision			Recall			F-Score		
		VC	MU	KL	VC	MU	KL	VC	MU	KL	VC	MU	KL
50-Noisy	Blocks	**0.88**	0.85	0.70	**0.90**	0.62	0.44	0.90	**1.00**	0.70	**0.90**	0.77	0.54
	Hanoi	0.93	**0.95**	0.80	**0.90**	**0.90**	0.60	**0.90**	**0.90**	0.60	**0.90**	**0.90**	0.60
	SKGrid	**0.97**	0.80	0.75	**1.00**	0.60	0.50	**1.00**	0.60	0.50	**1.00**	0.60	0.50
100-Noisy	Blocks	**0.95**	0.85	0.85	**1.00**	0.62	0.62	**1.00**	**1.00**	**1.00**	**1.00**	0.77	0.77
	Hanoi	0.97	**1.00**	0.95	**1.00**	**1.00**	0.90	**1.00**	**1.00**	0.90	**1.00**	**1.00**	0.90
	SKGrid	**0.97**	0.85	0.85	**1.00**	0.70	0.70	**1.00**	0.70	0.70	**1.00**	0.70	0.70
Avg	Blocks	**0.91**	0.85	0.78	**0.95**	0.62	0.53	0.95	**1.00**	0.85	**0.95**	0.77	0.65
	Hanoi	0.95	**0.97**	0.88	**0.95**	**0.95**	0.75	**0.95**	**0.95**	0.75	**0.95**	**0.95**	0.75
	SKGrid	**0.97**	0.82	0.80	**1.00**	0.65	0.60	**1.00**	0.65	0.60	**1.00**	0.65	0.60

Table 2: Anti-noise performance comparison (Action-State)

with different levels of observability, covering 10%, 30%, 50%, 70%, and 100%. Additionally, we introduce two noisy variants with 50% and 100% observability. Each variant in each domain includes 10 *GR* problems, each with 4 candidate goals, resulting in 210 problems.

We selected four state-of-art *GR* algorithms for comparison with *GRVC*. The first two algorithms are reinforcement learning-based methods proposed by Amado et al.[3], evaluated using MaxUtil and KL-divergence, referred to as ***MU*** and ***KL*** respectively. The second is a Landmark-based approach developed by Ramon et al.[20], known as ***LM***. The final method, ***GR***, is a heuristic estimator-based approach introduced by Ramírez and Geffner[23].

Next, we outline the hyperparameters used in this evaluation. To ensure fairness, both *GRVC* and *MU* utilised the same *Q-Learning* algorithm and hyperparameters (Sutton 1988). In this work, we improved the sampling algorithm compared to Amado et al.[3]'s work by prioritising sampling states and actions in the set of observations. So, to compare our approach with the their work, we re-run their approach with this new sampling technique, following the same rule that the training stops if the agent is able to achieve the goal 1,000 times. For *GRVC*, goals without sufficient samples within this limit are discarded. *GRVC* also has an additional hyperparameter: the number of *ELBO* iterations. We tested 100, 500, 800, and 1,000 iterations, finding performance fluctuations under 5%, with optimal results at 1,000 iterations. Further iterations were deemed unnecessary due to the narrow variational range.Consequently, results are based on 1,000 *ELBO* iterations.

We conducted performance evaluations under three observation input modes: having both action and state as observations (Action-State), only the states (State-Only), and only the actions (Action-Only). For each mode, we tested performance across varying levels of observability as well as in noisy environments, using standard machine learning metrics for evaluation. Notably, the State-Only and Action-Only modes, particularly in noisy conditions, present significant challenges for existing methods.

4.2 General Evaluation

As shown in Table 1, we evaluated the performance of each method under the ***Action-State*** observation mode. *GRVC* demonstrated near-perfect performance, achieving almost 100% accuracy across all observability conditions. Notably, in the SKGrid environment, *MU*'s performance was 4% to 8% lower than *KL*, whereas in the Blocks and Hanoi environments, MU outperformed *KL*. This discrepancy may be attributed to the characteristics of SKGrid's grid-based navigation, where multiple optimal paths can lead to the goal, making it difficult for *MU* to consistently maximise the utility of a single trajectory. *GRVC*, on the other hand, is less affected by this due to its integration of multiple optimal paths into its causal cube. When constructing the causal cube for each candidate *G*, the more paths leading to the goal, the stronger the corresponding ω, which in turn increases the *CE* of that goal. As long as the observation sequence contains steps from any optimal plan for the candidate goal, this chain effect is amplified during Variational Inference (*VI*).

4.3 Anti-Noise Evaluation

In the anti-noise evaluation, we compared the performance of each method in noisy observation environments. As shown in Table 2, *GRVC*'s average performance in both 50% and 100% observability was nearly equivalent to that in noise-free settings. While there was a 10% decline in performance at 50% observability, all metrics remained above 90% even in the worst-case scenario. Although *MU* also performed similarly to its noise-free results, *GRVC* consistently outperformed *MU*, particularly in the SKGrid domain, where *GRVC*'s accuracy surpassed *MU* by 15%, with other metrics leading by 35%. In contrast, *KL*'s Precision, recall, and f-score were approximately 10% lower than in noise-free conditions. Overall, the noise-resilience tests demonstrated *GRVC*'s exceptional robustness in noisy environments.

4.4 Unilateral Evaluation

In the unilateral evaluation, the algorithms are compared based on their performance under two types of observational inputs: State-Only and Action-Only. It is important to note that *RG* does not support State-Only inputs and *KL* does not support unilateral observations, thus the State-Only evaluation includes only *MU* and *LM*, while the Action-Only evaluation involves *MU*, *LM*, and *RG*.

GRVC does not inherently support unilateral observations. To address this, we developed a fair observation recovery algorithm that reconstructs unilateral observations into complete ones, effectively converting the problem into one of noise robustness. Figure 7 shows the average accuracy of the approaches when dealing with only state or only action as observation, as well as the inclusion of noise. The results demonstrate that the recovery algorithm performs exceptionally well in both modes, with recovery accuracy often exceeding 60% and reaching up to 97% on maximum.

As demonstrated in Figure 7, *GRVC* based on State-Only observations maintained over 95% performance across most domains

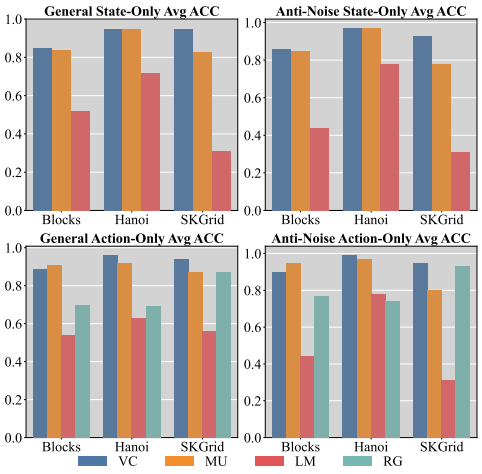

Figure 7: *Average accuracy of unilateral observations.*

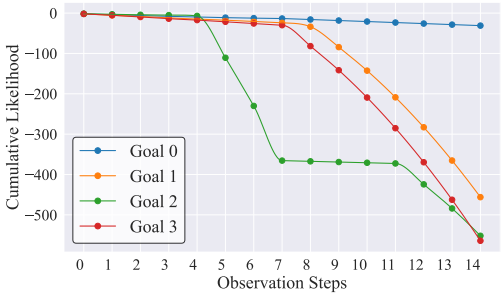

Figure 8: Stepwise cumulative likelihood plot for Problem 1 with 100% observation of Blocks.

in noise-free environments, with the exception of Blocks, where the accuracy was slightly lower at 85%, though still superior to other methods. In the Action-Only evaluation, *LM* showed lower precision and F-Score compared to other methods, with minimum values of 33% (Blocks) and 45% (SKGrid). *RG* also performed poorly, with Precision reaching only 40% in both Blocks and Hanoi. In contrast, *GRVC* consistently maintained over 94% in all metrics across different domains. *GRVC* successfully maintained performance in the State-Only anti-noise tests at levels comparable to the general evaluation. In contrast, *MU* and *LM* exhibited considerable performance degradation, with *MU* achieving 55% in precision, recall, and F-Score on SKGrid, and *LM* experiencing even more severe declines, with an average accuracy of just 31% and pecision of 17%. In the Action-Only tests, *GRVC* achieved 100% across most metrics, further validating the effectiveness of the recovery algorithm and the robust noise resilience of *GRVC*.

4.5 Decision Explainability

Finally, we showcase the explainability capabilities of our approach. The decision-making process of *GRVC* for each goal is determined by Equation 21, which allows for effective decision interpretation. Equation 21 describes the cumulative likelihood at each observation step when pursuing the current goal. Using Problem 1 in the Blocks domain with 100% observability as an example, Figure 8 illustrates the cumulative likelihood of observations. The likelihood at each observation step t is the sum of the likelihoods from step 0 to step t. From this, while achieving goal 1, there is a sharp drop in cumulative likelihood at step 8. Similarly, for goals 2 and 3, significant drops occur at steps 4, 7, and 11. Since the observation sequence is part of the optimal plan for achieving the correct goal, we infer that steps 4, 7, 8, and 11 are likely critical for the correct goal (*Goal 0*).

5 RELATED WORK

Regarding the origins of *Model-Approximate GR*, Bauer's seminal paper laid the foundational groundwork, detailing how to create plan libraries from logged actions.[7] Complementing this, Bisson et al. innovatively integrated neural networks with plan libraries, employing RNNs to delve into the decision-making processes of agents, with the aim of predicting actions based on learned behavioural patterns.[8] Amado et al.[5] extended Asai and Fukunaga's architecture[6], allowing for planning and plan recognition tasks over latent vectors. In this work, *PDDL* domain models are generated from latent vectors and used for *GR*. Recent studies have utilised Process Mining for *Model-Approximate GR*. These techniques extract information from event logs and traces to uncover models for *GR*. Process Mining facilitates the understanding of how real processes are executed, identifying bottlenecks, deviations, and opportunities for enhancing efficiency. Polyvyanyy et al. developed a probabilistic recognition method that depends on Process Mining techniques to discover models from plan traces or event logs.[22] Su et al. expanded and enhanced Polyvyanyy's approach, performing comprehensive and detailed empirical evaluations against the latest *GR* methods using various benchmarks.[24]

6 DISCUSSION & CONCLUSION

This work introduces a novel *Model-Approximate GR* method, *GRVC*, which combines causal reasoning with Variational Inference (*VI*) to achieve goal recognition. *GRVC* performs causal modelling on sampled sequences and applies *VI* for counterfactual reasoning, integrating trajectory likelihood to infer the correct goal. Evaluations demonstrate that *GRVC* excels in terms of both accuracy and stability, outperforming the existing state-of-the-art. Furthermore, *GRVC* retains great precision when tackling problems with noise, showing to be more robust than the other approaches.

As a short-coming, the current implementation of *GRVC* takes a lot of time to construct causal models, particularly in unilateral observation evaluations, where it requires more time and memory to compute the missing actions (or states), posing certain limitations for real-time applications. We believe incorporating causal reasoning into *GR* methods shows great potential, and our work represents a first step toward a new class of *GR* approaches. Future work will focus on two areas: testing our approach in complex stochastic domains and exploring the use of landmarks [21] in approximated domains (e.g., causal graphs) to enhance both performance and explainability.

REFERENCES

[1] Abeer Alshehri, Tim Miller, and Mor Vered. 2023. Explainable Goal Recognition: A Framework Based on Weight of Evidence. *Proceedings of the International Conference on Automated Planning and Scheduling* 33, 1 (Jul. 2023), 7–16. https://doi.org/10.1609/icaps.v33i1.27173

[2] Leonardo Amado, João Paulo Aires, Ramon Fraga Pereira, Mauricio Cecilio Magnaguagno, Roger Granada, and Felipe Meneguzzi. 2018. LSTM-Based Goal Recognition in Latent Space. *CoRR* abs/1808.05249 (2018). arXiv:1808.05249 http://arxiv.org/abs/1808.05249

[3] Leonardo Amado, Reuth Mirsky, and Felipe Meneguzzi. 2022. Goal Recognition as Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 36, 9 (Jun. 2022), 9644–9651. https://doi.org/10.1609/aaai.v36i9.21198

[4] Leonardo Amado, Sveta Paster Shainkopf, Ramon Fraga Pereira, Reuth Mirsky, and Felipe Meneguzzi. 2024. A Survey on Model-Free Goal Recognition. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, Kate Larson (Ed.). International Joint Conferences on Artificial Intelligence Organization, 7923–7931. https://doi.org/10.24963/ijcai.2024/877 Survey Track.

[5] Leonardo Amado, Ramon Fraga Pereira, João Aires, Mauricio Magnaguagno, Roger Granada, and Felipe Meneguzzi. 2018. Goal Recognition in Latent Space. In *2018 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN.2018.8489653

[6] Masataro Asai and Alex Fukunaga. 2018. Classical Planning in Deep Latent Space: Bridging the Subsymbolic-Symbolic Boundary. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018). https://doi.org/10.1609/aaai.v32i1.12077

[7] Mathias Bauer. 1998. Acquisition of abstract plan descriptions for plan recognition. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence* (Madison, Wisconsin, USA) *(AAAI '98/IAAI '98)*. American Association for Artificial Intelligence, USA, 936–941.

[8] Francis Bisson, Hugo Larochelle, and Froduald Kabanza. 2015. Using a recursive neural network to learn an agent's decision model for plan recognition. In *Proceedings of the 24th International Conference on Artificial Intelligence* (Buenos Aires, Argentina) *(IJCAI'15)*. AAAI Press, 918–924.

[9] Daniel Borrajo, Sriram Gopalakrishnan, and Vamsi K. Potluru. 2020. Goal recognition via model-based and model-free techniques. *CoRR* abs/2011.01832 (2020). arXiv:2011.01832 https://arxiv.org/abs/2011.01832

[10] Ankush Ganguly and Samuel W. F. Earp. 2021. An Introduction to Variational Inference. *CoRR* abs/2108.13083 (2021). arXiv:2108.13083 https://arxiv.org/abs/2108.13083

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[12] Josiah P. Hanna, Arrasy Rahman, Elliot Fosong, Francisco Eiras, Mihai Dobre, John Redford, Subramanian Ramamoorthy, and Stefano V. Albrecht. 2021. Interpretable Goal Recognition in the Presence of Occluded Factors for Autonomous Vehicles. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 7044–7051. https://doi.org/10.1109/IROS51168.2021.9635903

[13] Yue Hu, Kai Xu, Budhitama Subagdja, Ah-Hwee Tan, and Quanjun Yin. 2021. Interpretable Goal Recognition for Path Planning with ART Networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN52387.2021.9534409

[14] Sarah Keren, Avigdor Gal, and Erez Karpas. 2015. Goal Recognition Design for Non-Optimal Agents. *Proceedings of the AAAI Conference on Artificial Intelligence* 29, 1 (Mar. 2015). https://doi.org/10.1609/aaai.v29i1.9645

[15] Mariane Maynard, Thibault Duhamel, and Froduald Kabanza. 2019. Cost-Based Goal Recognition Meets Deep Learning. *CoRR* abs/1911.10074 (2019). arXiv:1911.10074 http://arxiv.org/abs/1911.10074

[16] Leland Gerson Neuberg. 2003. Causality: models, reasoning, and inference, by judea pearl, cambridge university press, 2000. *Econometric Theory* 19, 4 (2003), 675–685.

[17] Ana Rita Nogueira, Andrea Pugnana, Salvatore Ruggieri, Dino Pedreschi, and João Gama. 2022. Methods and tools for causal discovery and causal inference. *WIREs Data Mining and Knowledge Discovery* 12, 2 (2022), e1449. https://doi.org/10.1002/widm.1449 arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1449

[18] Judea Pearl. 2010. Causal Inference. In *Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008 (Proceedings of Machine Learning Research, Vol. 6)*, Isabelle Guyon, Dominik Janzing, and Bernhard Schölkopf (Eds.). PMLR, Whistler, Canada, 39–58. https://proceedings.mlr.press/v6/pearl10a.html

[19] J. Pearl, M. Glymour, and N.P. Jewell. 2016. *Causal Inference in Statistics: A Primer*. Wiley. https://books.google.com/books?id=I0V2CwAAQBAJ

[20] Ramon Fraga Pereira, Nir Oren, and Felipe Meneguzzi. 2020. Landmark-based approaches for goal recognition as planning. *Artificial Intelligence* 279 (2020), 103217. https://doi.org/10.1016/j.artint.2019.103217

[21] Ramon Fraga Pereira, Andre Grahl Pereira, and Felipe Meneguzzi. 2019. Landmark-Enhanced Heuristics for Goal Recognition in Incomplete Domain Models. *Proceedings of the International Conference on Automated Planning and Scheduling* 29, 1 (Jul. 2019), 329–337. https://doi.org/10.1609/icaps.v29i1.3495

[22] Artem Polyvyanyy, Zihang Su, Nir Lipovetzky, and Sebastian Sardina. 2020. Goal Recognition Using Off-The-Shelf Process Mining Techniques. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Auckland, New Zealand) *(AAMAS '20)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1072–1080.

[23] Miquel Ramırez and Hector Geffner. 2009. Plan recognition as planning. In *Proceedings of the 21st international joint conference on Artifical intelligence. Morgan Kaufmann Publishers Inc.* Citeseer, 1778–1783.

[24] Zihang Su, Artem Polyvyanyy, Nir Lipovetzky, Sebastian Sardiña, and Nick van Beest. 2023. Fast and accurate data-driven goal recognition using process mining techniques. *Artificial Intelligence* 323 (2023), 103973. https://doi.org/10.1016/j.artint.2023.103973

[25] Franz A. Van-Horenbeke and Angelika Peer. 2021. Activity, Plan, and Goal Recognition: A Review. *Frontiers in Robotics and AI* 8 (2021). https://doi.org/10.3389/frobt.2021.643010

[26] Leiyu Xie, Yuxing Yang, Zeyu Fu, and Syed Mohsen Naqvi. 2024. Position and Orientation Aware One-Shot Learning for Medical Action Recognition from Signal Data. *IEEE Transactions on Multimedia* (2024), 1–14. https://doi.org/10.1109/TMM.2024.3521703

[27] Alessio Zanga, Elif Ozkirimli, and Fabio Stella. 2022. A Survey on Causal Discovery: Theory and Practice. *International Journal of Approximate Reasoning* 151 (2022), 101–129. https://doi.org/10.1016/j.ijar.2022.09.004