

# Beyond Static Assumptions: the Predictive Justified Perspective Model for Epistemic Planning

Guang Hu<sup>1</sup>[0000–0003–3629–8040], Weijia Li<sup>1</sup>[0009–0008–9342–1619], and  
Yangmengfei Xu<sup>1</sup>[0000–0003–4320–0064]

The University of Melbourne, Parkville VIC 3010, AUS  
{ghu1,weijia3,yangmengfeix}@student.unimelb.edu.au

## 1 Introduction

*Epistemic Planning* (EP), as an advancement of classical planning, has been developed to enable sound planning based on agents’ knowledge and beliefs. Currently, all EP approaches follow the assumption (the “static” assumption from classical planning) that the environment remains static unless explicitly altered by an agent’s actions. However, in practical applications, this assumption sometimes fails to hold. For example, it is unrealistic to treat all pedestrians as static for autonomous driving vehicles operating in busy urban areas. Accurately modeling pedestrian motion—often represented as a first-order polynomial—is therefore critical, particularly in scenarios involving jaywalking. Moreover, this process can be significantly impaired by occlusions such as trucks, increasing risks to both autonomous vehicles and human drivers. In such situations, experienced truck drivers may proactively signal surrounding vehicles whose view is obstructed, based on their nested belief that those drivers lack an accurate belief about the jaywalker’s motion model. Addressing this gap necessitates an epistemic reasoning model capable of accounting for environmental changes, thereby aligning EP frameworks more closely with real-world applications.

Currently, EP is primarily addressed through three main approaches. The *Dynamic Epistemic Logic* (DEL)-based approach [1] was first proposed, and it maintains a Kripke structure [2] using an event-based model, which requires explicit action effects to specify modal logic changes. *Knowledge Bases* strategy is another approach that maintains and updates agents’ knowledge/belief databases by converting the epistemic planning problems into classical planning problems [8, 7]. When the scale of the problem increases, both DEL and pre-compilation methods become computationally challenging. To address this challenge, a *state-based* approach, *Planning with Perspectives* (PWP) [5] leverages external functions and lazy evaluation, which enable offloading the epistemic formula reasoning from the planner, hence improving both efficiency and expressiveness. Moreover, they define a semantics that reasons based on solely agents’ observations, which proved can be done in polynomial time with regard to nesting depth. However, the PWP approach only handles knowledge (not belief). A recent continuation study introduced the *Justified Perspectives* (JP) model

to handle the belief [6]. The JP model generates the belief in a way drawing inspiration from two intuitions of human reasoning: humans believe what they see; and, for the parts they could not see, humans believe what they have seen in the past unless they saw evidence to suggest otherwise.

Compared to other approaches, the state-based approach is more suitable as: 1) it is computationally efficient; 2) it is expressive; and 3) it does not require modeling epistemic logic updates explicitly (as action effects). Therefore, in this work, we propose an extension of the JP model to accommodate dynamic environments.

## 2 Predictive Justified Perspective Model

In this section, we formally propose the *Predictive Justified Perspective* (PJP) Model to model a continuously changing environment.

The definition of the signature  $\Sigma = (Agt, V, D, \mathcal{R})$ , language  $L_{GKB}(\Sigma)$ , states (sets of variable assignments,  $\mathcal{S}$  and  $\mathcal{S}_c$  as state spaces and complete-state space), and model instance  $M = (Agt, V, D, \pi, O_1, \dots, O_k)$  are adopted from the JP model [6], as well as three functions: observation function ( $O_i : \mathcal{S} \rightarrow \mathcal{S}$ , which should be contractive, idempotent, and monotonic), retrieval function ( $R : \vec{\mathcal{S}} \times \mathbb{Z} \times V \rightarrow \mathbb{D}$ ) and Justified Perspective (JP) function ( $f_i : \vec{\mathcal{S}}_c \rightarrow \vec{\mathcal{S}}_c$ ). Their intuition is that agents reason about beliefs by constructing a justified perspective (state sequence) according to their own observation ( $O_i$ ), and for the parts they are not observing, they retrieve the most recent observation ( $R$ ).

To differentiate from the concept of the “static” variable in AI planning, the variables in our model are named as *processual*<sup>1</sup> variable which are state variables whose values can evolve as a result of external environmental processes, independently of the agent’s actions. To describe a set of processual variables, the processual variable model is denoted as  $\Omega$ , and defined below.

**Definition 1 (Processual Variable Model)** *Given  $\mathcal{T}$  is a set that includes all processual variable types,  $\Omega$  is defined as:*

$$\Omega = (V, \mathcal{T}, type, \eta), \text{ where } : type : V \rightarrow \mathcal{T}, \eta : V \rightarrow \mathbb{R}^*, * \in \mathbb{N}. \quad \blacksquare$$

To indicate the changing rules, for each processual variable  $v \in V$ , the type and coefficients (or parameters) are defined as  $type(v)$  and  $\eta(v)$ . A “static” variable is modeled by the new framework as a processual variable with a special type *static*. It is considered as a base case ( $=0$ ) in which  $\eta(v) = \{()\}$ . Unlike classical planning variables, which are modified exclusively by the agent’s action effects, processual variables are updated according to not only the agent’s action effects but also exogenous transition rules or continuous dynamics reflecting changes in the environment. In the remainder of this paper, the term “variable” refers to a processual variable unless stated otherwise.

Then, the PJP model inherits the JP model, except for the processual variable model  $\Omega$  and the set of all *Predictive Retrieval* (PR) functions. The PR functions

<sup>1</sup> The idea of the process is from PDDL+ [4].

are defined abstractly as follows, providing flexibility and expressiveness for the modeller to model any variable with a known type.

**Definition 2 (Predictive Retrieval Function)** *A predictive retrieval function  $pr_{type(v)} : \vec{S} \times \mathbb{N} \times V \rightarrow \mathbb{D}$  takes the input of a state sequence  $\vec{s}$ , a timestamp  $t$  and a variable  $v$ , and outputs the predicted value of  $v$  at  $t$  based on  $\vec{s}$ , where  $type(v) \in \mathcal{T}$  is the processual variable  $v$ 's type. The predictive retrieval function  $pr_{type(v)}$  should satisfy the following properties. Given the input state sequence  $\vec{s} = [s_0, \dots, s_n]$ , its prediction  $\vec{p}$  is defined as  $\vec{p} = [p_0, \dots, p_n]$ , where for  $t' \in [0, n]$ ,  $p_{t'} = \{v = pr_{type(v)}(\vec{s}, t', v) | v \in V\}$ .*

– **Preserving Consistency** [Compulsory]:

$$\forall v \in V, \forall t < |\vec{s}|, v \in \vec{s}[t] \Rightarrow \vec{p}[t](v) = \vec{s}[t](v)$$

– **Recursive Consistency** [Compulsory]: *Let  $\vec{W} \subseteq \vec{S}$  ( $\vec{S}$  is sequence space), such that:  $\forall \vec{w} \in \vec{W} \Rightarrow |\vec{w}| = |\vec{s}|$  and  $\forall t < |\vec{s}| \Rightarrow \vec{s}[t] \subseteq \vec{w}[t] \subseteq \vec{p}[t]$ .*

$$\forall v \in V, \forall t < |\vec{s}|, \forall \vec{w} \in \vec{W} \Rightarrow pr_{type(v)}(\vec{w}, t, v) = \vec{p}[t](v)$$

– **Reconstructive Consistency** [Optional]:

$$\forall i \in \text{Agt}, \exists \vec{w} \in \vec{S}, O_i(\vec{w}) = \vec{s} \Rightarrow \vec{s} = O_i(\vec{p}) \quad \blacksquare$$

The prediction function  $pr_{type(v)}$  estimates the value of variable  $v$  at timestamp  $t$  by deriving  $v$ 's changing pattern based on the given state sequence  $\vec{s}$ . A valid predictive retrieval function must be preserving consistent and recursively consistent. **Preserving Consistency** ensures the predicted value of the variable is consistent with its input. This gives  $\forall t < |\vec{s}| \Rightarrow \vec{s}[t] \subseteq \vec{p}[t]$ , where  $\vec{p}$  is constructed in the above definition. **Recursive Consistency** ensures that the predictive retrieval function yields stable results when the input state sequence is the original input state sequence ( $\vec{s}$ ) with extra values from its prediction ( $\vec{p}$ ). This consistency condition requires that for all sequences in the set of sequences  $\vec{W}$  (which is the same as the original input sequence with extra assignments from its prediction), applying the predictive retrieval function to  $\vec{w}$  yields the same predicted value as in  $\vec{p}$ . A simple example involves a sequence that contains only one variable  $v'$  with values of  $[1, \perp, 3, \perp]$ . By applying  $pr_{type(v')}([1, \perp, 3, \perp], t', v')$  for all timestamps  $t' < 3$ , we have predicted values of  $v$  as  $[1, 2, 3, 4]$ . Then, for an input sequence that is the same as the original sequence but with extra values from its prediction, such as  $[1, 2, 3, \perp]$ , the predicted value  $pr_{type(v')}([1, 2, 3, \perp], t', v')$  should remain the same for all timestamps  $t' < 3$  ( $[1, 2, 3, 4]$ ). The optional **Reconstructive Consistency** ensures that, for all agents, the predictive function is consistent with their own observation functions. That is, when using the agent's observation as the input, the predicted values should not alter the agent's observation. This property only ensures agents' beliefs are justifiable ( $B_i K_i \varphi \Rightarrow K_i \varphi$ ) following the JP model, while the others ensure beliefs logic following **KD45** axioms.

As a base case,  $pr_{static}$  works the same as in the JP model ( $pr_{static}(\vec{s}, t, v) = R(\vec{s}, t, v)$ ). Then, it is up to the modeller to design  $PR$  functions for other processual variable types. With all  $PR$  functions that follow Definition 2, we can now provide our definition to generate justified perspectives with prediction.

Then, we can define a *Predictive Justified Perspective* (PJP) function for agent  $i$ ,  $f_i : \vec{S} \rightarrow \vec{S}_c$  similar to the JP function, except that the values of unobserved variables are not returned by  $PR$  functions.

The semantics for the PJP model and the JP model are the same, except that the perspectives now contain predictions. It is more challenging to show that the ternary semantics for the PJP model (with all  $PR$  functions being preserving consistent and recursively consistent) satisfies **the KD45** axioms, particularly because “ $f_i(\vec{s}) = f_i(f_i(\vec{s}))$ ” does not hold. In addition, we claim that the PJP model follows the same complexity class as the JP model, except with additional computations on  $PR$  functions. The theorems and proofs, as well as example predictive retrieval functions, can be found in a complete version of this paper through <https://arxiv.org/abs/2412.07941>.

### 3 Experiment

To show the effectiveness and efficiency of the PJP model, experiments are conducted on the most challenging benchmark domain, Grapevine, with the adoption of dynamically changing variables. The vanilla version of the *Breadth-First Search (BFS)* is used in all tests to avoid the influence of the search algorithm.

Grapevine domain [8] is a well-known benchmark in EP. It involves three agents ( $a$ ,  $b$ , and  $c$ ) who can **share**, **lie**, or **move** in two connected rooms ( $rm_1$  and  $rm_2$ ), and initially, all three agents are located in  $rm_1$ . Each agent ( $a$  as an example) has a secret ( $sa$ ), and they can share its value truthfully ( $ssa = tsa$ ) or lie about it ( $ssa = lsa$ ). In this experiment, to make it more challenging, the agents are allowed to **share** others’ secrets. That is, they can share others’ secrets based on the values they believe in. It is noted that the other agents don’t know the truth or falsity of the shared value (e.g.  $ssa$ ), and thus, when they share others’ secrets, they can only share the  $ssa$  according to their beliefs, which can be equal to  $tsa$ ,  $lsa$ , or other numbers (e.g. from a false prediction). To force examining agents’ beliefs instead of knowledge (derived from current observation), we carefully choose the belief value of  $ssa$  and enforce that all agents need to take a **wait** action right after any share or lie action.

The following outcome metrics are included to demonstrate the performance of the PJP model: the number of generated nodes ( $|gen|$ ), the total execution time ( $\tau_t$ ), the rule of processual variables (Rule of  $x$ ), corresponding coefficients for  $tsa$  and  $lsa$  are provided in column  $\eta(tsa)$  and  $\eta(lsa)$ , the predictive retrieval function of the processual variables ( $pr_{type(x)}$ ), definitions of all  $pr_{type(x)}$  functions, the plan length ( $|plan|$ ), and the goal conditions (Goal).

All source code, including domain encoding, predictive retrieval function implementations, and plan validator, has been released as an open-source project. The predictive retrieval functions used in this experiment are indicated by their

names, and the formal mathematical definitions are documented (omitted due to page limit).

### 3.1 Result and Discussion

$ gen $	$\tau_t(s)$	Rule of $x$	$\eta(tsa)$	$\eta(lsa)$	$pr_{type(x)}$	$ plan $	Goal
G0 81	0.11				$pr_{1st\_poly}$	2	$tsa = 5 \wedge B_b ssa = 4$
G1 1211	1.78				$pr_{1st\_poly}$	4	$tsa = 7 \wedge B_b ssa = 7$
G2 1211	2.12	$x = at + b$	$[1, 3]$	$[1, 1]$	$pr_{linear\_reg}$	4	$tsa = 7 \wedge B_b ssa = 7$
G3 421	0.55				$pr_{1st\_poly}$	4	$tsa = 7 \wedge B_b ssa = 8$
G4 58662	197.07				$pr_{1st\_poly}$	7	$B_c ssa = 10 \wedge B_a B_c ssa = \perp$
G5 63916	221.56				$pr_{1st\_poly}$	7	$B_c ssa = 10 \wedge B_a B_c (ssa \neq 10 \wedge ssa \neq \perp)$
G6 17277	26.84	$x = at^2 + bt + c$	$[1, 0, 2]$	$[1, 0, 0]$	$pr_{2st\_poly}$	6	$tsa = 38 \wedge B_b ssa = 38$
G7 81	0.15	$x = a^t$	$[3]$	$[1]$	$pr_{power}$	2	$tsa = 9 \wedge B_b ssa = 9$
G8 17277	35.93	$x = a \sin(bt + c)$	$[8, 5, 4]$	$[1, 1, 1]$	$pr_{sin}$	6	$tsa = 4.23 \wedge B_b ssa = 4.23$

**Table 1.** Experimental results for Grapevine instances.

The result is shown in Table 1. Variable  $tsa$  and  $lsa$  are first-order polynomials in G0-G5, while we also show other example types in G6, G7, and G8. G0 shows the base case where  $a$  only shared  $tsa$  once, which means  $b$  does not have enough value to predict  $ssa$ . G3 shows that  $b$  believes  $ssa$  is a value that never occurred during the currently visited or reached search space (as both  $tsa$  and  $lsa$  are smaller than 8). G4 and G5 are challenging instances such as agent  $c$  believes the correct  $ssa$  value, while agent  $a$  is either not aware of that or has false beliefs on  $c$  about  $ssa$ .

The experiment results show the PJP model is able to solve problem instances in EP domains involving dynamically changing variables. By forming agents' predictive justified perspectives using  $pr_{type(\cdot)}$  and  $f_i$ , the agents are able to reason about unseen changing variables with reasonable predictions.

The PJP model adopted the strength of the JP model, including arbitrary nesting and action-model free (without the need to explicitly model action's epistemic effects), which makes it suitable for further applications. An interesting point that can be noticed is that the choice of the predictive retrieval function may influence the efficiency of the planner. For example, the  $pr_{linear\_reg}$  used in G2 is less efficient compared to the analytical method in G1.

In addition, when considering a more advanced grapevine domain, where agents can lie about someone's secrets to substage other agents' beliefs about this someone. This potentially can be handled by a predictive retrieval function that iterates throughout all the observations and generates all possible coefficient combinations, then choosing the one with the most occurrences. With the flexibility of the PJP model, other commonly used outlier rejection algorithms such as *three-sigma rule* and *Random Sample Consensus* (RANSAC) [3] can also be implemented as a valid  $PR$  function. This indicates the possibilities to utilize the PJP model incorporating external functions to solve broader problems.

## References

1. Bolander, T., Andersen, M.B.: Epistemic planning for single- and multi-agent systems. *Journal of Applied Non-Classical Logics* **21**(1), 9–34 (2011). <https://doi.org/10.3166/jancl.21.9-34>, <https://doi.org/10.3166/jancl.21.9-34>
2. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. MIT Press (1995). <https://doi.org/10.7551/MITPRESS/5803.001.0001>, <https://doi.org/10.7551/mitpress/5803.001.0001>
3. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In: Fischler, M.A., Firschein, O. (eds.) *Readings in Computer Vision*, pp. 726–740. Morgan Kaufmann, San Francisco (CA) (1987). <https://doi.org/https://doi.org/10.1016/B978-0-08-051581-6.50070-2>, <https://www.sciencedirect.com/science/article/pii/B9780080515816500702>
4. Fox, M., Long, D.: Modelling mixed discrete-continuous domains for planning. *J. Artif. Intell. Res.* **27**, 235–297 (2006). <https://doi.org/10.1613/JAIR.2044>, <https://doi.org/10.1613/jair.2044>
5. Hu, G., Miller, T., Lipovetzky, N.: Planning with perspectives – decomposing epistemic planning using functional strips. *J. Artif. Int. Res.* **75** (dec 2022). <https://doi.org/10.1613/jair.1.13446>, <https://doi.org/10.1613/jair.1.13446>
6. Hu, G., Miller, T., Lipovetzky, N.: Planning with multi-agent belief using justified perspectives. In: *Proceedings of the Thirty-Third International Conference on Automated Planning and Scheduling. ICAPS ’23, AAAI Press* (2023). <https://doi.org/10.1609/icaps.v33i1.27193>, <https://doi.org/10.1609/icaps.v33i1.27193>
7. Muise, C., Belle, V., Felli, P., McIlraith, S., Miller, T., Pearce, A.R., Sonenberg, L.: Efficient multi-agent epistemic planning: Teaching planners about nested belief. *Artificial Intelligence* **302**, 103605 (2022). <https://doi.org/https://doi.org/10.1016/j.artint.2021.103605>, <https://www.sciencedirect.com/science/article/pii/S0004370221001569>
8. Muise, C., Belle, V., Felli, P., McIlraith, S.A., Miller, T., Pearce, A.R., Sonenberg, L.: Planning over multi-agent epistemic states: A classical planning approach. In: *AAAI Conference on Artificial Intelligence* (2015), <https://api.semanticscholar.org/CorpusID:14923173>