Learning to Reason about Contextual Knowledge for Planning under Uncertainty

Cheng Cui[1,2] **Saeid Amiri**[1] **Yan Ding**[1] **Xingyue Zhan**[1] **Shiqi Zhang**[1]

[1]Department of Computer Science, SUNY Binghamton, Binghamton, New York, USA
[2]Cognex Corporation, Natick, Massachusetts, USA
{ccui7, samiri1, yding25, xzhan215,zhangs}@binghamton.edu

Abstract

Sequential decision-making (SDM) methods enable AI agents to compute an action policy toward achieving long-term goals under uncertainty. Existing research has shown that contextual knowledge in declarative forms can be used for improving the performance of SDM methods. However, the contextual knowledge from people tends to be incomplete and sometimes inaccurate, which greatly limits the applicability of knowledge-based SDM methods. In this paper, we develop a novel algorithm for knowledge-based SDM, called PERIL, that learns from interaction experience to reason about contextual knowledge, as applied to urban driving scenarios. Experiments have been conducted using CARLA, a widely used autonomous driving simulator. Results demonstrate PERIL's superiority in comparison to existing knowledge-based SDM baselines.

1 INTRODUCTION

Artificial intelligence agents need to estimate the current world state while determining what to do based on the current state estimation, resulting in the problem of sequential decision-making (SDM) under partial observability [Kaelbling et al., 1998, Hausknecht and Stone, 2015, Jaakkola et al., 1994]. Existing research has demonstrated that an agent's SDM capability can be improved by reasoning with contextual knowledge to estimate the current world state [Zhang and Stone, 2015, Chitnis et al., 2018]. However, the contextual knowledge provided by domain experts can hardly be comprehensive, and sometimes includes inaccurate information. Motivated by the observation that AIs need significant efforts to recover from inaccurate knowledge in SDM tasks [Amiri et al., 2020], we aim to develop an approach to help the SDM robots learn to reason about contextual knowledge.

Consider a lane changing scenario in urban driving. On the one hand, the vehicle needs to perceive the environment, e.g., using Lidar range sensors, to detect whether there is sufficient room in the desired lane. The perception output, together with other contextual information (say weather and traffic), is then processed in a reasoning system to estimate the world state, including the intentions of other drivers (humans or not). On the other hand, the vehicle can plan actions to actively facilitate lane changing, such as using turn signals to request space, and slowing down to find room for the lane change. Existing methods have enabled robots to logically reason about the world state, and use the reasoning results to facilitate decision making [Zhang and Sridharan, 2022]. However, how to learn from a robot's decision making experience to improve the reasoning capability for SDM tasks is still an open problem.

In this paper, we develop a learning algorithm for knowledge-based SDM, called *perceptual reasoning and interactive learning* (**PERIL**), as shown in Figure 1 1. We use a perceptual reasoner that consists of a deep supervised learning classifier and a knowledge base of logic rules for perceiving and reasoning about the current world state. The perceptual reasoner takes as input streaming data from onboard sensors and observable facts, such as current time and weather. The contextual information is processed together to compute a distribution representing the current world state estimation. The distribution is then provided to the interaction component as an informative prior to guide its action selections toward achieving long-term goals.

The **main contributions** of this paper includes:

- A formal statement of the knowledge-based SDM problem we are concerned with, where we specify the algorithm input and output, as well as the assumptions;
- The PERIL algorithm that enables AIs to learn from both contextual knowledge and data gathered at runtime to close the perceive-reason-act loop;

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

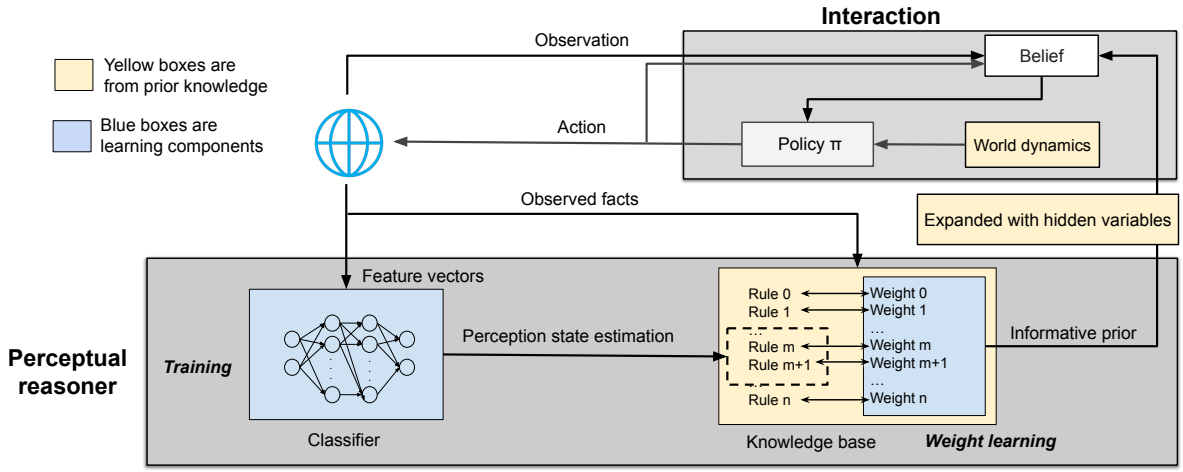

Figure 1: An overview of PERIL. The perceptual reasoner consists of a classifier for passive perception and a knowledge base of rules and weights for automated reasoning. It receives streaming data of feature vectors and facts from the environment. Based on the perception state estimation of the classifier and observed facts, the perceptual reasoner uses the knowledge base to infer an informative prior to compute the initial belief for the interaction. In the interaction, world dynamics refers to the way in which the environment and the state evolve over time (in POMDPs, world dynamics is represented by the transition function). Then a policy suggests the best action for information collection based on the belief at each time step. Finally, the loop is closed by providing feedback containing labels and ground truth to the perceptual reasoner, training the classifier, and learning the new weights of rules.

- Extensive experiments and illustrative trials in urban driving scenarios using CARLA-based [Dosovitskiy et al., 2017] simulation for demonstrating the superiority of our approach.

In comparison to competitive baselines [Amiri et al., 2020, Ulbrich and Maurer, 2013], we found that PERIL improves the autonomous vehicle's overall performance in sequential decision-making by increasing cumulative rewards and reducing interaction costs.

2 RELATED WORK

This paper is about incorporating perceptual reasoning and interactive learning into sequential decision-making under uncertainty. We discuss research topics that are relevant to this work.

Researchers have developed methods that incorporate human knowledge in declarative forms into planning under uncertainty frameworks [Göbelbecker et al., 2011, Zhang and Stone, 2015, Hanheide et al., 2017, Chitnis et al., 2018, Amiri et al., 2020, 2022]. There are other works that studied how human knowledge can be used to improve the performance of reinforcement learning (RL) agents [Zhang et al., 2022b, Leonetti et al., 2016, Yang et al., 2018, Icarte et al., 2022, Jiang et al., 2019, Hayamizu et al., 2021, Zhang et al., 2022b]. A survey paper summarized research on knowledge-based sequential decision making [Zhang and Sridharan, 2022]. Those methods use a knowledge base that cannot be updated as the agent becomes more experienced. Recently, LLM-based planning methods have been proposed, such as SayCan [Ahn et al., 2022], and Inner Monologue [Huang et al., 2022]. Nevertheless, these methods lack the ability to reason about human knowledge (COWP and LLM+P are exceptions [Liu et al., 2023, Ding et al., 2022b]), whereas our approach explicitly addresses quantitative uncertainty. In comparison, PERIL learns to reason about contextual knowledge, producing agent behaviors that are robust to imperfect knowledge.

AIs, including autonomous vehicles, that operate in the real world require the simultaneous capabilities of perception for estimating the current world state, and planning to achieve long-term goals [Nilsson, 1984, Thrun et al., 2005]. It is a common practice that the perception component outputs the current world state in a symbolic form to the planning component [Khandelwal et al., 2017, Veloso, 2018, Shuai and Chen, 2019]. There is recent research from the literature that tightly integrates the perception and planning components [Hausknecht and Stone, 2015, Lee et al., 2020, Wang et al., 2020, Srinivas et al., 2018, Ding et al., 2022a]. There is the survey paper on interactive perception that summarized relevant research [Bohg et al., 2017]. They used machine learning techniques, e.g., a deep neural network, to estimate the current world state. What is passed to the planning component includes not only the current state in symbolic forms, but also the (un)reliability information. Also, recent

research developed an approximate algorithm to help the agent choose a subset of exogenous state variables to reason about when planning and planning in such a reduced state space can often be significantly more efficient than planning in the full model [Chitnis and Lozano-Pérez, 2020]. There exists research that uses universal planning networks to learn underlying representations through visual perceptions so as to optimize planning. The learned representation can be leveraged to transfer task-related semantics to other agents for more challenging tasks. PERIL shares the same spirit with the above-mentioned methods by learning complex representations for estimating the current world state [Srinivas et al., 2018]. Beyond that, PERIL leverages contextual knowledge from domain experts to refine the output from neural networks (CNNs in our case) before passing it along to the planning component.

Autonomous vehicles, as a type of robots, need to plan their behaviors under partial observability [Bai et al., 2015]. More specifically, the on-board sensors cannot provide a global view of the environment, and the vehicles need to estimate the current world state based on the streaming data collected over time. POMDPs are well suitable for planning behaviors under partial observability [Kaelbling et al., 1998], and have been used in planning for autonomous vehicles [Ulbrich and Maurer, 2013, Wray et al., 2017, Suchan et al., 2019, Wray et al., 2021, Ha et al., 2020, Zhang et al., 2022a]. For instance, Wray et al. used POMDPs to reason at the times when the perception data is limited, but their approach does not leverage any contextual knowledge for reasoning.

Work closest to this research is an algorithm called LCORPP [Amiri et al., 2020] that learns from data and reasons about human knowledge to estimate the world state. They used LSTM [Hochreiter and Schmidhuber, 1997] for sequence classification, and P-log [Baral et al., 2009] for representing and reasoning about contextual knowledge. Other than a different application domain, the main difference from that work is that PERIL is able to learn from the interaction experience to improve its reasoning capability, using Markov logic networks [Richardson and Domingos, 2006, Domingos and Lowd, 2019]. To the best of our knowledge, PERIL is the first work that learns to use human knowledge for sequential decision-making under uncertainty.

3 BACKGROUND

In this section, we summarize three key techniques used in this paper: convolutional neural networks, Markov logic networks, and partially observable Markov decision processes.

3.1 CONVOLUTIONAL NEURAL NETWORKS

A convolutional neural network (CNN) is comprised of convolutional layers followed by fully connected layers as in a standard multilayer neural network [LeCun et al., 1998]. The basic building blocks of CNN consist of convolutional, pooling, activation, and fully-connected layers. In a convolutional layer, a filter is passed over the image, viewing a few pixels at a time. The convolution operation is a dot product of the original pixel values with weights defined in the filter. Pooling layers are used for downsampling, and fully-connected layers output a list of probabilities for different possible labels. The activation layers introduce non-linearity. The architecture of a CNN is designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal). We use CNNs for the perception of road conditions in this work.

3.2 MARKOV LOGIC NETWORKS

Markov networks are undirected cyclic probabilistic graphical models where each edge has a potential function [Richardson and Domingos, 2006, Domingos and Lowd, 2019]. Markov logic networks (MLNs) are a template for building Markov networks. They are first-order knowledge bases with a weight associated with each rule. A first-order logic knowledge base is a set of hard constraints on the set of possible worlds: if a world violates even one formula, it has zero probability. The basic idea in MLNs is to soften these constraints: *When a world violates one formula in the knowledge base it is less probable, but not impossible.* An MLN program is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and w_i is a real number that specifies the weight of the formula. Learning in MLNs can be done using the following equation:

$$\frac{\partial log P_w(X = x)}{\partial w_i} = n_i(x) - \sum_{x'} P_w(X = x') n_i(x')$$

where the sum is over all possible databases x', and $P_w(X = x')$ is $P(X = x')$ computed using current weight vector $w = (w_1, ..., w_i, ...)$ and $n_i(x)$ is the true groundings in data x. In this paper, we use MLNs to enable an SDM agent to not only reason with human knowledge, but also learn to improve its reasoning capability from experience.

3.3 PARTIALLY OBSERVABLE MDPS

Markov decision processes (MDPs) can be used for SDM. When the environment is not fully observable, we can use POMDPs that generalize MDPs by assuming partial observability of the current state [Kaelbling et al., 1998]. A partially observable MDP (POMDP) is a tuple $(S, A, T, R, Z, O, \gamma)$ where S is the state space, A is the action set, T is the state-transition function, R is the reward function, O is the observation function, Z is the observation set, and γ is a discount factor that determines the planning horizon.

A POMDP agent maintains a belief state distribution b with observations ($z \in Z$) using the Bayes update rule:

$$b'(s') = \frac{O(s', a, z) \sum_{s \in S} T(s, a, s') b(s)}{Pr(z|a, b)}$$

where s is the state, a is the action, $Pr(z|a, b)$ is a normalizer, and z is an observation. Solving a POMDP produces a policy that maps the current belief state distribution to an action toward maximizing long-term utilities.

4 PROBLEM STATEMENT

In this section, we formally present the knowledge-based sequential decision making problem to pave the way for the learning algorithm developed in this paper. We first define the problem domain by the tuple below:

$$\langle \Theta, \underbrace{E, F, H, \overbrace{Q, V}^{\mathbf{V}^I}}_{\mathbf{V}^R}, A, T, Z, O \rangle.$$

The agent is provided with contextual knowledge Θ, a finite set of first-order logical statements (rules). The logical rules of Θ are over a finite set of variables $\mathbf{V}^R = F \cup E \cup H \cup Q$, where F, E, H, and Q are sets of fact, evidence, hidden, and query variables respectively. V is the set of latent variables for interaction. Interaction variables $\mathbf{V}^I$ consists of query and latent variables ($\mathbf{V}^I = Q \cup V$). The agent is provided with a finite set of actions A that the agent can perform. T is a transition function: $T(s, a, s') = Pr(s'|s, a)$, where $s, s' \in S$ is the factored space specified by $\mathbf{V}^I$. Z is an observation set, and O is an observation function: $O(s, a, z) = Pr(z|s, a)$.

Figure 2 depicts the two sets of variables $\mathbf{V}^R$ and $\mathbf{V}^I$ for reasoning and interaction respectively, and their overlap on Q. Variable sets E, F, H, and Q are mutually exclusive. Logical reasoning with Θ produces the combinatorial possible settings of $\mathbf{V}^R$ that are consistent to the logical statements. The query variables are shared by both interaction and reasoning variables ($Q = \mathbf{V}^R \cap \mathbf{V}^I$, and $Q \neq \emptyset$). Some properties of the variables and their values:

- The agent cannot directly observe the variables of $H \cup Q \cup V$.
- Values of fact variables F can be directly collected from the world and no perception is needed.
- Variables E are estimated via streaming data λ from sensory readings (e.g., Lidar sensors and cameras).

In episode i and at execution time t, the agent receives $z_t \in Z$, and sensory readings λ_t, where λ_t^i is a perception of E_i. After each episode i (i.e., when a terminal state is reached), values of $\mathbf{V}^R \cup \mathbf{V}^I$ are provided by a human expert, and the collected data can be used for learning purposes.

The robot's task is specified by a reward function $R(s, a) \rightarrow \mathbb{R}$. The objective is to compute a policy π for the robot to choose actions at each time step toward maximizing its expected future discounted reward, $\mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, where γ is a discount factor, and r_t is the reward received at time t.

Remarks: The diagram in Figure 2 can be viewed as an integration of two subproblems. The "reasoning" box points to a logical-probabilistic reasoning subproblem [Richardson and Domingos, 2006, Baral et al., 2009, Wang et al., 2019], whose input includes logical facts (F), e.g., current time, and evidence (E), e.g., using computer vision techniques. The values of E are estimated using streaming data λ. One can use provided logical-probabilistic rules to infer the values of Q (**Subproblem I**). The "interaction" box and the two variables of A and Z together capture the dependencies of a POMDP *at one specific step*, which is the second subproblem of planning under uncertainty (**Subproblem II**). Variables Q and V form the state space of a POMDP, and values of Z are used for state estimation. One can compute a policy π for the POMDP for sequentially selecting $a \in A$. When the two subproblems overlap on some variables (Q), one can leverage the reasoning results to guide a robot's sequential decision-making. To the best of our knowledge, it is the first time that this integrated reasoning and planning problem is formulated using a pictorial diagram.

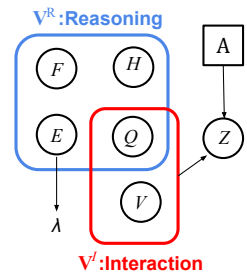

Figure 2: Domain variables and their dependencies.

While existing research has investigated reasoning for planning under uncertainty [Zhang and Stone, 2015, Chitnis et al., 2018, Amiri et al., 2020], those robots cannot improve their reasoning capabilities as the robots become more experienced. Next, we present a learning algorithm that helps a robot improve its skills of leveraging domain knowledge for decision making under uncertainty.

5 ALGORITHM

In this section, we present PERIL, short for "*perceptual reasoning and interactive learning*," a novel algorithm that addresses the knowledge-based sequential decision-making problem described in Section 4. A PERIL agent perceives the environments using supervised learning, reasons over domain variables using contextual knowledge, and generates interaction behaviors using a decision-theoretic planning approach. PERIL's reasoning capability is enhanced via relational learning as the agent is more experienced over time.

Algorithm 1 describes PERIL, the key contribution of this re-

Algorithm 1 PERIL

Ensure: Domain $\langle \Theta, E, F, H, Q, V, A, T, Z, O \rangle$, reward function R, and parameter N

Require: MLN system Sol^R, POMDP system Sol^P, relational learning system Lrn^R, and supervised learning system Lrn^S

1: Initialize dataset $\Phi \leftarrow \emptyset$; dataset $\Psi \leftarrow \emptyset$; $\pi \leftarrow$ random; classifier $C \leftarrow$ random
2: Initialize weights W, $w \leftarrow 1.0$, each w corresponds to $\theta \in \Theta$

3: Compute π using Sol^P for POMDP: $(Q \cup V, A, T, Z, O, R)$
4: **while** true **do** {No termination condition – lifelong learning}
5: **for** $i \in [0, N-1]$ **do**
6: Get λ, and f from the world, where f is a vector of F
7: $e \leftarrow C(\lambda)$; e is a vector and includes the values of E
8: $Pr(Q) \leftarrow Sol^R(\Theta, W, f, e)$
9: Compute distribution b over state set $S = Q \cup V$ using $Pr(Q)$ and uniform distributions over variables V
10: **while** s is not a terminal state **do**
11: Select action $a \leftarrow \pi(b)$ and execute a
12: Make an observation z
13: Update b based on a and z
14: **end while**
15: Collect ground truth values $\mathbf{v}^R = \{\hat{e}, \hat{f}, \hat{h}, \hat{q}\}$
16: Augment dataset: $\Phi \leftarrow \Phi \cup \{\lambda : \hat{e}\}$
17: Augment dataset: $\Psi \leftarrow \Psi \cup \{\mathbf{v}^R\}$
18: **end for**
19: $C \leftarrow Lrn^S(\Phi)$ {Supervised learning}
20: $W \leftarrow Lrn^R(\Theta, \Psi, W)$ {Relational learning}
21: **end while**

search. The input includes a domain description, a problem description specified by reward function R, and parameter N for batch-based learning. Implementing PERIL systems requires software tools for relational learning (Lrn^R) and supervised learning (Lrn^S), as well as MLN and POMDP systems (Sol^R and Sol^P). Lines 1-2 are for initialization, where Φ and Ψ are for storing data for supervised learning and relational learning respectively.

There are three loops in PERIL. Each iteration of the **outer while loop** (Lines 4-21) corresponds to one batch where supervised learning and relational learning are activated once (Lines 19-20). The nested **for-loop** (Lines 5-18) includes N iterations – each corresponding to a sequence of perceptual reasoning (Lines 5-9), interaction (Lines 10-14), and data augmentation (Lines 15-17). In perceptual reasoning, the agent infers the query variables Q, using the logical weighted rules (Θ, W), and the direct observations (f) or the estimated observations (e) from the world (Lines 6-8). Using the union of inferred Q and V, PERIL builds the initial prior belief b (Line 9), where the posterior is calculated in the inner interaction while-loop (Lines 10-14). Once the interaction loop is done, two datasets are augmented with the newly collected data instances. The **inner while loop** (Lines 10-14) corresponds to one episode, where the agent takes one action a, makes an observation z, and updates

(a) Cooperative (b) Not cooperative

Figure 3: Two lane merging situations in CARLA-based simulation. (a) The vehicle on the left is cooperative and yields the right of way. (b) The vehicle on the left is not cooperative

belief b. The actions in this loop are suggested by policy π calculated by Sol^P. PERIL is a lifelong learning algorithm for SDM, and does not have a termination condition.

PERIL agents learn from interaction experience to improve their capabilities of reasoning with contextual knowledge from people, and planning under uncertainty. To the best of our knowledge, no existing algorithm supports this "learning to reason and plan" capability. Next, we describe a full instantiation of PERIL, as applied to an urban driving domain.

6 INSTANTIATION

We use CARLA, an open-source autonomous driving simulation platform, to illustrate a realization of PERIL [Dosovitskiy et al., 2017]. A CARLA environment consists of 3D models of vehicles, traffic signs, buildings, and pedestrians. Figure 3 shows two example lane-merging situations. Next, we provide technical details of each component of our PERIL framework.

CNNs for Perception C is our classifier that takes as input raw sensory data (3D Lidar sensory readings in our case), and outputs the road condition. We use CNN to build classifier C, and to process streaming data λ from Lidar sensors. Figure 4 shows how C is constructed in our instantiation. The 3D sensory readings are first projected to 2D space. Then the road area is cropped out to generate a 2D image, which is fed into CNNs for classification. The output of classifier C is saved in variable $CarsDetected$ ($true$ or $false$). In our domain, E includes only one element: $E = \{CarsDetected\}$.

MLNs for Logical Probabilistic Reasoning We use MLN for logical probabilistic reasoning, and relational learning. Our MLN-based reasoner includes five variables: $Weather$, $Time$, $Crowded$, $CarsDetected$, and $Cooperative$. Among them, $Weather$ and $Time$ are fact variables: $F = \{Weather, Time\}$. The weather can be $Sunny$ or $Rainy$, and the time is either $Busy$ or $Normal$, which is used for reasoning about traffic condition. $Crowded$ and $Cooperative$ are query variables:

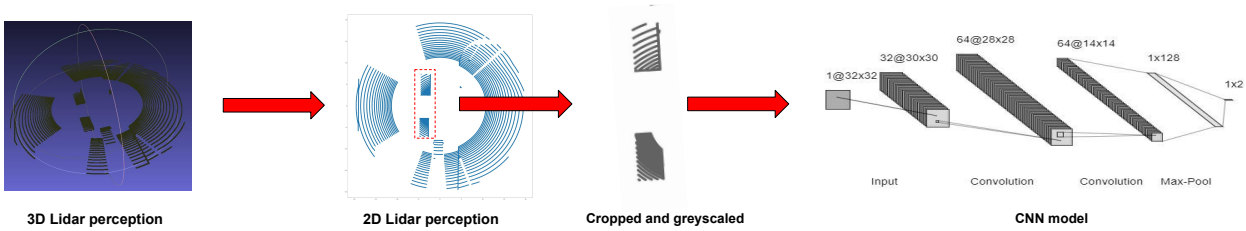

Figure 4: An overview of the perception component where the vehicle receives raw data from the Lidar sensor. The sensory readings are projected to 2D space, and converged into an image. Finally, a CNN outputs if the desired lane is sensed crowded.

$Q = \{Crowded, Cooperative\}$. $H = \emptyset$. There is one evidence variable $E = \{CarsDetected\}$. Other drivers' behaviors are simplified to a binary variable of $Cooperative$ with a domain of $true$ or $false$. An MLN program includes a set of first-order logical statements, where each is associated with a weight. We use MLN to build our logical probabilistic reasoner Sol^R and relational learning system Lrn^R. First-order logic rules Θ form the declarative domain knowledge base. For instance, the following rule

```
Time(+t,s)→ Crowded(+c,s)
```

indicates that the time implies the crowdedness of the road. If it is at busy time, it is likely the road is crowded. If it is at normal time, it is more likely that the road is not crowded. The second rule

```
Crowded(+c,s)→ CarsDetected(+d,s)
```

states that, when the road is crowded, it is more likely that the ego vehicle can detect surrounding cars. The third rule

```
Weather(+w,s) ∧ Crowded(+c,s)→ *Cooperative(s)
```

states that the weather condition and the road crowdedness affects the surrounding vehicles (drivers) being cooperative or not. For example, rainy weather (e.g., affecting drivers' visibility) and crowded roads might cause the drivers to be less cooperative. All rules Θ are associated with weights. During weight (relational) learning, each rule is converted to conjunctive normal form, and a weight is learned for each of its clauses. It should be noted that those are "commonsense" rules that are normally correct but not always. MLNs are well suited for learning to reason with those rules. We then use the input of H,E and F to infer the value of Query variables Q from MLN.

POMDPs for Planning under Uncertainty We use POMDPs to construct a probabilistic planner for active information gathering, and goal achievement. $S : Q \times V \cup \{term\}$ is the state space, where $term$ is a terminal state that identifies the end of an episode. $V = \{RoomAvailable\}$. $RoomAvailable = true$ means that there is room available in the desired lane for the ego vehicle's lane merging behavior. We consider three behaviors in our action space: $A = \{signal,\ move,\ merge\}$, where we assume the vehicle can only merge to one side of the road (say left). $signal$ means that the vehicle uses turn signal to indicate its intention to merge. $move$ means that the vehicle adjusts its position to get prepared for lane changing, which is also useful for communicating its intention to the other drivers. Intuitively, after the vehicle is confident that there is room in the desired lane, and the other drivers are cooperative, the vehicle should take the $merge$ action.

We use transition function $T(s, a, s') = Pr(s'|s, a)$ to model how action a leads the transition from s to s'. Actions except for $merge$ have different costs (a small negative value). Action $merge$ causes either a big reward or a big penalty (a big negative value), depending on the road condition (values of $Cooperative$, and $RoomAvailable$). For instance, if $Cooperative = false$ or $RoomAvailable = false$, action $merge$ will result in a big penalty. Action costs, success reward, and failure penalty are modeled in reward function $R(s, a)$.

The observation set is $Z : \{true, false, na\}$. We use the observation function $O(s, a, z) = Pr(z|s, a)$ to describe the perception model of the vehicle. For instance, when $Cooperative = true$, there is 0.7 probability that the vehicle observe $true$ (the other drivers are cooperative).

7 ILLUSTRATIVE EXAMPLE

Figure 5 shows an example trial. The vehicle first collected a "fact" that it was a rainy day at a busy time. The vehicle received streaming data, and the CNN classifier outputs that $CarsDetected = true$, meaning that the left lane is occupied by at least one vehicle. Reasoning with contextual knowledge about weather and time, our vehicle believed that it was likely the road was crowded and the other drivers were less cooperative. The ego vehicle then used our MLN-based reasoner to perform probabilistic inference, and found that $Pr(Crowded = true) = 0.995$, and $Pr(Cooperative = false) = 0.970$. Those probabilities were used to initialize

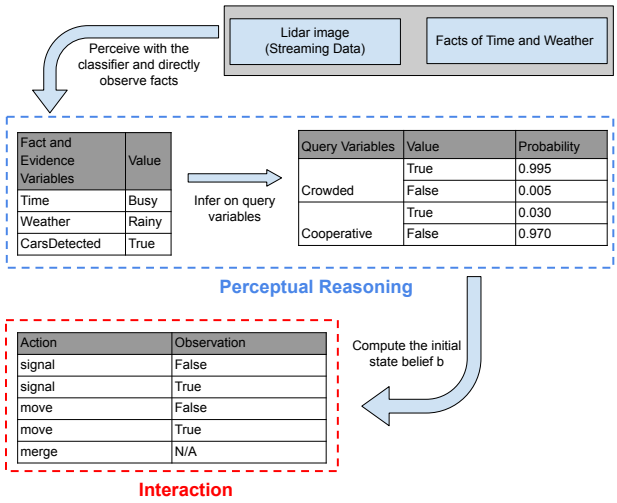

Figure 5: An illustrative example trial of PERIL.

the POMDP belief b. With the initial belief of the current state and sequential observations, the ego vehicle repeatedly selected actions as shown in Figure 6. After two $signal$ and two $move$ actions, the ego vehicle successfully completed a merging lane task.

8 EXPERIMENTS

We have conducted experiments using the CARLA simulator to evaluate the key hypothesis that learning to reason about domain knowledge improves the agent's performance within the sequential decision-making context. We have compared PERIL with the following baselines. **LCORPP** is a baseline method that uses supervised learning for perception, and automated reasoning to guide a probabilistic planner [Amiri et al., 2020]. LCORPP's knowledge base is hardcoded, so it cannot learn to reason about knowledge. **PERIL w/o POMDP** is the same as PERIL except that the action policy is manually crafted: the vehicle takes up to two $signal$ actions (depending on the confidence on state estimation), then a $move$ action, and $merge$. **POMDP-LC** is a classic POMDP-based approach for planning lane changing behaviors [Ulbrich and Maurer, 2013], which includes neither supervised learning nor relational learning.

Experiment Setup In each trial, we first spawn our ego vehicle, which is tasked to merge to the left lane. We set the range of Lidar sensor to $20m$. We sequentially spawn M vehicles on the left lane ($0 \leq M \leq 8$ in our case) within an area of $radius = 20m$ around the ego vehicle. If a vehicle has any contact with an existing one, then this vehicle is moved and re-spawned. We annotated the Lidar sensory data: if there exist two vehicles in the left lane that are at most $10m$ away from each other, then a Lidar instance is labeled $true$, i.e., $CarsDetected = true$. Otherwise, the label is $false$. Fact variables $Time$ and $Weather$ were sampled uniformly. $Crowded$ and $Cooperative$ were sampled using the Markov network of our MLN program. For instance, if $Time = normal$, then there is probability 0.7 that $Crowded = true$. We have added perception noise into the observation model. For instance, the vehicle's observation is correct in 0.7 probability. The costs of $signal$ and $move$ actions are $10s$ and $15s$ respectively. Successful and unsuccessful trials receive 100 and -100 reward respectively.

We used Alchemy for MLN-based relational learning and logical probabilistic reasoning.[1] POMDPs were solved using an off-the-shelf solver [Kurniawati et al., 2008]. We used PyTorch [Paszke et al., 2019] for training the CNNs.

Experimental Results Every data point in our figures is an average of 4,000 trials, evenly distributed into 5 runs. We evaluated the mean values of the 5 runs for each data point, and used the 5 mean values to generate the standard errors.

Figure 7 shows the results of comparing PERIL with three baseline methods. We see that PERIL achieved the highest cumulative reward on average, and required the lowest interaction cost on average. The LCORPP baseline produced the second best performance in both reward and cost, which indicates the usefulness of perceptual reasoning. In a stochastic world, LCORPP cannot learn how likely the handcrafted rules are correct while with perceptual reasoning, PERIL can learn weights associated with such rules for better reasoning. Specifically, PERIL uses MLN to learn to reason about contextual knowledge, which contributes to the best performance among the four methods. All methods produced an average success rate between 0.87 to 0.89, where we did not observe statistically significant differences among the methods. A successful merge is when the ego vehicle merges left in the presence of enough room and vehicle cooperation. An unsuccessful merge in our setup functions like a risky situation in practice, and does not indicate a collision, because autonomous vehicles (or human drivers) have collision-avoidance mechanisms, which are not considered in our experiments. Results here support our key hypothesis that PERIL outperforms baseline methods with higher rewards and lower costs.

Table 1 shows the performances of PERIL and baselines under low and high perception capabilities. Low (high) perception quality corresponds to a POMDP observation function, where the vehicle can correctly perceive "crowdedness" in 0.7 (0.9) probability. Our hypothesis is that PERIL's superiority over the other methods is not affected by the vehicle's perception system. The results suggest that PERIL significantly outperformed the baselines at **0.05 significance level**.

[1] https://alchemy.cs.washington.edu/

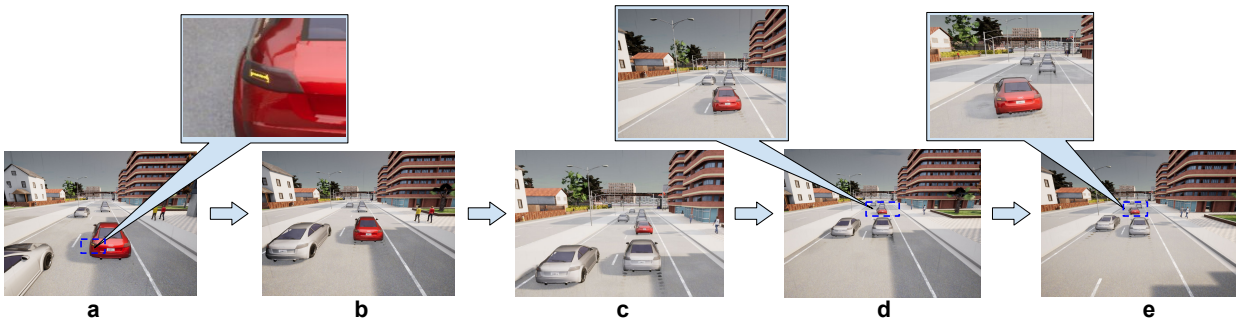

Figure 6: The ego vehicle took a sequence of actions in the interaction process to successfully merge left. (a) The ego vehicle intended to merge left. It turned on the left signal. (b) The surrounding vehicle on the left was not cooperative at first. The ego vehicle kept left blinking. (c) The surrounding vehicle on the left became cooperative, and the ego vehicle started to move left. (d) The ego vehicle kept moving left and found room in the left lane. (e) The ego vehicle successfully merged left.

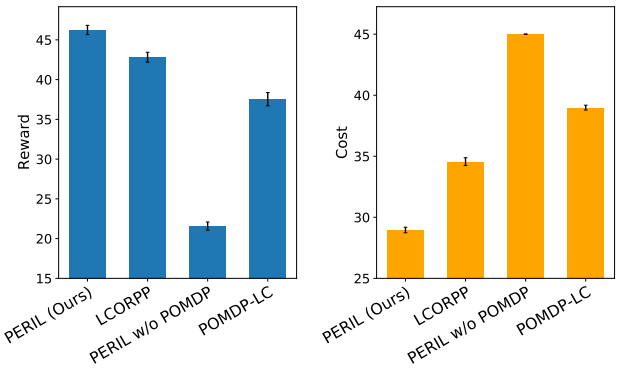

Figure 7: PERIL performed better than the baselines in both overall reward, and interaction cost.

Ablation Study We did an ablation study to evaluate the importance of the two learning components in PERIL (supervised learning and relational learning). The results are shown in Figure 8. Our first observation is that PERIL performed better than its two ablations in both overall reward, and interaction cost, except for the very early learning phase. Another observation is that relational learning plays an important role in the PERIL system. When relational learning was disabled, there was significant increase in interaction cost, in comparison to the ablation with supervised learning removed. This is potentially because the MLN-based reasoner can learn to "compensate" for the missing perception component.

9 CONCLUSION AND FUTURE WORK

In this work, we develop an algorithm called PERIL that learns to reason with contextual knowledge for sequential decision-making. PERIL uses convolutional neural networks for perception, Markov logic networks for reasoning, and partially observable Markov decision processes for planning under uncertainty. We have extensively evaluated PERIL in urban driving scenarios. Results suggest that PERIL outperformed competitive baselines, as well as its own ablations, in both overall reward and interaction cost.

Table 1: The performances of PERIL and baselines in reward and cost under different perception qualities. PERIL performed the best in both reward and cost with statistically significant improvement, as indicated using italic font.

Algorithm	Perception quality			
	Low		High	
	Reward	Cost	Reward	Cost
PERIL	***46.5*** (0.5)	***28.7*** (0.3)	***64.1*** (0.7)	***27.1*** (0.3)
LCORPP	43.5 (1.0)	34.1 (0.3)	62.4 (0.4)	31.0 (0.2)
PERIL w/o POMDP	20.9 (0.9)	45.0 (0.0)	20.2 (1.0)	45.0 (0.0)
POMDP-LC	40.2 (0.8)	39.7 (0.1)	62.4 (0.4)	32.3 (0.2)

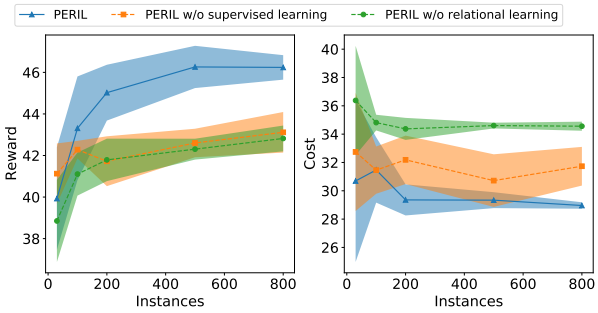

Figure 8: PERIL performed better than its two ablative versions as more data instances were provided for training.

Currently, the vehicle learns to perceive the environment (road condition) from data, and learns to improve its reasoning capability using MLN. One direction of future work is to replace the POMDP-based planner with a reinforcement learning component. By doing that, the vehicle will be able to learn to select actions from its task-completion experience. Another direction is to actively acquire knowledge from people [Amiri et al., 2019], commonsense knowledge

bases [Speer et al., 2017], or pre-trained models [Brown et al., 2020] to avoid hand-coding rules.

The experimental setting focuses only on a small subset problem of autonomous driving and certain set-ups are simplified, due to the limitation of time and resources. To further verify the scalability of PERIL in solving real-world complicated autonomous driving problems, the data collection in autonomous driving (or other multiagent, interactive) domains could be expensive, time-consuming, and sometimes risky, which is far beyond the scope of this paper. But this is definitely something PERIL (ours) practitioners should consider. We will consider applying PERIL to other non-driving domains and incorporating robot control into the loop in the future.

Acknowledgements

This work has taken place at the Autonomous Intelligent Robotics (AIR) Group, SUNY Binghamton. AIR research is supported in part by grants from the National Science Foundation (NRI-1925044), Ford Motor Company (URP Award 2019-2022), OPPO (Faculty Research Award 2020), and SUNY Research Foundation.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Saeid Amiri, Sujay Bajracharya, Cihangir Goktolgal, Jesse Thomason, and Shiqi Zhang. Augmenting knowledge through statistical, goal-oriented human-robot dialog. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

Saeid Amiri, Mohammad Shokrolah Shirazi, and Shiqi Zhang. Learning and reasoning for robot sequential decision making under uncertainty. In *AAAI*, volume 34, pages 2726–2733, 2020.

Saeid Amiri, Kishan Chandan, and Shiqi Zhang. Reasoning with scene graphs for robot planning under partial observability. *IEEE Robotics and Automation Letters*, 7 (2):5560–5567, 2022.

Haoyu Bai, Shaojun Cai, Nan Ye, David Hsu, and Wee Sun Lee. Intention-aware online pomdp planning for autonomous driving in a crowd. In *ICRA*, 2015.

Chitta Baral, Michael Gelfond, and Nelson Rushton. Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming*, 9(1):57–144, 2009.

Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Rohan Chitnis and Tomás Lozano-Pérez. Learning compact models for planning with exogenous processes. In *CoRL*, pages 813–822. PMLR, 2020.

Rohan Chitnis, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrating human-provided information into belief state representation using dynamic factorization. In *IROS*, 2018.

Yan Ding, Cheng Cui, Xiaohan Zhang, and Shiqi Zhang. Glad: Grounded layered autonomous driving for complex service tasks. *arXiv preprint arXiv:2210.02302*, 2022a.

Yan Ding, Xiaohan Zhang, Saeid Amiri, Nieqing Cao, Hao Yang, Chad Esselink, and Shiqi Zhang. Robot task planning and situation handling in open worlds. *arXiv preprint arXiv:2210.01287*, 2022b.

Pedro Domingos and Daniel Lowd. Unifying logical and statistical ai with markov logic. *Communications of the ACM*, 62(7):74–83, 2019.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *CoRL*, pages 1–16. PMLR, 2017.

Moritz Göbelbecker, Charles Gretton, and Richard Dearden. A switching planner for combined task and observation planning. In *AAAI*, 2011.

Jung-Su Ha, Danny Driess, and Marc Toussaint. A probabilistic framework for constrained manipulations and task and motion planning under uncertainty. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6745–6751. IEEE, 2020.

Marc Hanheide, Moritz Göbelbecker, Graham S Horn, Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, et al. Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, 247:119–150, 2017.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI fall symposium series*, 2015.

Yohei Hayamizu, Saeid Amiri, Kishan Chandan, Keiki Takadama, and Shiqi Zhang. Guiding robot exploration in reinforcement learning via automated planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 31, pages 625–633, 2021.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Rodrigo Toro Icarte, Toryn Q Klassen, Richard Valenzano, and Sheila A McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022.

Tommi Jaakkola, Satinder P Singh, and Michael I Jordan. Reinforcement learning algorithm for partially observable markov decision problems. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, pages 345–352, 1994.

Yuqian Jiang, Fangkai Yang, Shiqi Zhang, and Peter Stone. Task-motion planning with reinforcement learning for adaptable mobile service robots. In *IROS*, 2019.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 1998.

Piyush Khandelwal, Shiqi Zhang, Jivko Sinapov, Matteo Leonetti, Jesse Thomason, Fangkai Yang, Ilaria Gori, Maxwell Svetlik, Priyanka Khante, Vladimir Lifschitz, et al. Bwibots: A platform for bridging the gap between ai and human–robot interaction research. *IJRR*, 36(5-7): 635–659, 2017.

Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *RSS*, volume 2008. Citeseer, 2008.

Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series, the handbook of brain theory and neural networks, 1998.

Michelle A Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks. *IEEE Transactions on Robotics*, 2020.

Matteo Leonetti, Luca Iocchi, and Peter Stone. A synthesis of automated planning and reinforcement learning for efficient, robust decision-making. *Artificial Intelligence*, 241:103–130, 2016.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023.

Nils J Nilsson. Shakey the robot. Technical report, SRI INTERNATIONAL MENLO PARK CA, 1984.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

Wei Shuai and Xiao-ping Chen. Kejia: towards an autonomous service robot with tolerance of unexpected environmental changes. *Frontiers of Information Technology & Electronic Engineering*, 20(3):307–317, 2019.

Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451, 2017.

Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *ICML*, 2018.

Jakob Suchan, Mehul Bhatt, and Srikrishna Varadarajan. Out of sight but not out of mind: An answer set programming based online abduction framework for visual sense-making in autonomous driving. In *28th International Joint Conference on Artificial Intelligence (IJCAI 2019), Macao, China, August 10-16, 2019*, pages 1879–1885. ijcai. org, 2019.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

Simon Ulbrich and Markus Maurer. Probabilistic online pomdp decision making for lane changes in fully automated driving. In *International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2063–2067. IEEE, 2013.

Manuela M. Veloso. The increasingly fascinating opportunity for human-robot-ai interaction: The cobot mobile service robots. *ACM Transactions on Human-Robot Interaction*, 7(1), May 2018.

Chen Wang, Shaoxiong Wang, Branden Romero, Filipe Veiga, and Edward Adelson. Swingbot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation. In *2020 IEEE/RSJ IROS*. IEEE, 2020.

Yi Wang, Shiqi Zhang, and Joohyung Lee. Bridging commonsense reasoning and probabilistic planning via a probabilistic action language. *Theory and Practice of Logic Programming*, 19(5-6):1090–1106, 2019.

Kyle Hollins Wray, Stefan J Witwicki, and Shlomo Zilberstein. Online decision-making for scalable autonomous systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4768–4774, 2017.

Kyle Hollins Wray, Bernard Lange, Arec Jamgochian, Stefan J Witwicki, Atsuhide Kobashi, Sachin Hagaribommanahalli, and David Ilstrup. Pomdps for safe visibility reasoning in autonomous vehicles. In *2021 IEEE ISR*, pages 191–195. IEEE, 2021.

Fangkai Yang, Daoming Lyu, Bo Liu, and Steven Gustafson. PEORL: integrating symbolic planning and hierarchical reinforcement learning for robust decision-making. In *IJCAI*, 2018.

Haodi Zhang, Zhenhao Chen, Junyang Chen, Yi Zhou, Defu Lian, Kaishun Wu, and Fangzhen Lin. Dynamic decision making framework based on explicit knowledge reasoning and deep reinforcement learning. *Journal of Software*, pages 0–0, 2022a.

Haodi Zhang, Zhichao Zeng, Keting Lu, Kaishun Wu, and Shiqi Zhang. Efficient dialog policy learning by reasoning with contextual knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11667–11675, 2022b.

Shiqi Zhang and Mohan Sridharan. A survey of knowledge-based sequential decision-making under uncertainty. *AI Magazine*, 43(2):249–266, 2022.

Shiqi Zhang and Peter Stone. Corpp: Commonsense reasoning and probabilistic planning, as applied to dialog with a mobile robot. In *AAAI*, volume 29, 2015.