# Uncertainty-based Decision Making Using Deep Reinforcement Learning

Xujiang Zhao*, Shu Hu*, Jin-Hee Cho†, Feng Chen*
* University at Albany - SUNY, Albany, NY, USA, {xzhao8, shu2, fchen5}@albany.edu
† Virginia Tech, Falls Church, VA, USA, jicho@vt.edu

***Abstract*—This work proposes an opinion inference algorithm in large graph network data using subjective, uncertain opinions. In the graph network data, an opinion is associated with an edge between two nodes where the edge indicates a known opinion while no edge refers to an unknown opinion for their relationship. The examples include the predictions of a road traffic condition (i.e., an edge indicates a road between two intersections and an opinion represents congested or non-congested) or trust relationships (i.e., an edge refers to a trust relationship between two users where an opinion indicates a user's trust in another user). To derive an unknown opinion between two nodes, we identify a set of best paths in the graph network data that can maximize decision performance (e.g., prediction accuracy). To solve this problem, we formulate each opinion using Subjective Logic (SL) and leverage a policy-based deep reinforcement learning (DRL) technique. We propose three DRL-based schemes combining SL and DRL where a reward is given based on a different type of uncertainty, including vacuity, dissonance, or monosonance. Via extensive simulation experiments, we investigate what type of uncertainty is a more critical factor than others in improving decision performance when a different uncertainty type is considered as a reward in DRL. We validated the outperformance of the proposed DRL-based schemes in terms of belief errors, prediction accuracy, and computation time based on both a semi-synthetic and real world datasets.**

***Index Terms*—subjective opinion, uncertainty, dissonance, monosonance, decision making, reinforcement learning**

## I. Introduction

In decision making research, uncertainty has been studied as one of critical factors that significantly affects decision performance. Belief theories mainly have been used to reason uncertainty in information caused by corrupted, deceptive, missing, conflicting, and/or incomplete evidence. As one of prominent belief theories, Subjective Logic (SL) has been developed to explicitly deal with uncertainty caused by a lack of information (a.k.a. vacuity or ignorance) [8]. However, the lack of information does not provide a wide spectrum of uncertainty dimensions because uncertainty can be derived from many different root causes. Recently, other dimensions of uncertainty in subjective opinions have been investigated in SL [9].

As artificial intelligence and/or machine/deep learning (ML/DL) has been realized as a powerful tool for decision making, combining belief models with mature, solid theories and ML/DL with high performance [12, 16] looks a promising direction to enhance decision making performance. In this work, we are interested in how subjective opinions with uncertainty can be used to infer unknown opinions for maximizing decision performance. In particular, we concern a large graph network data consisting of a set of nodes and a set of edges where an opinion is associated with an edge representing the relationship between two nodes. When some of opinions (edges) are known, we aim to accurately infer unknown opinions based on the structural relationships between nodes based on the fusion operators offered by SL (i.e., discounting and consensus operators [8]). For example, when $A$ trusts $B$ and $B$ trusts $C$, then we can infer $A$'s trust in $C$ based on the trust chain using the discounting operator in SL [8], as demonstrated in Fig. 1. However, what if there are multiple paths of trust chains, so $A$ can reach $D$, $E$, or $F$ to know about $C$ where all $D$, $E$, and $F$ knows $C$, $A$ can consider multiple paths to derive its trust in $C$. However, in order to make a best decision with low complexity, $A$ can select a partial set of the paths to maximize its decision accuracy with minimum cost. In order to solve this problem (i.e., path selection among multiple paths), we leverage a *policy-based deep reinforcement learning* (DRL) technique. Given a graph network data, we aim to identify a set of best paths maximizing decision accuracy. In our example above, $A$ needs to make a decision in choosing either $B$, $D$, $E$, or $F$ when it needs to choose only one path from itself ($A$) to $C$. In particular, we consider the reward in DRL using multiple types of uncertainty in which the reward considers minimum vacuity (due to a lack of evidence) or minimum dissonance (due to conflicting evidence) or maximum monosonance (minimum in both vacuity and dissonance) and investigate which dimension of uncertainty as a reward plays a most critical role in decision performance.

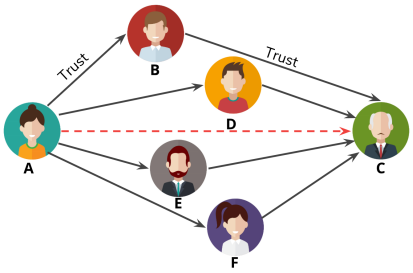



Fig. 1: An example opinion inference of a unknown trust relationship using trust chains.

This paper has the following **key contributions**:

- We formulated an opinion inference problem as a decision

making problem with uncertain, subjective opinions where multiple dimensions of uncertainty (i.e., vacuity, dissonance, or monosonance) are used to determine an optimal set of relational paths (i.e., trust chains) between nodes to infer unknown opinions. By leveraging fusion operators in SL (i.e., discounting and consensus operator) [8], we derive the unknown relations. The use of multiple dimensions of uncertainty in identifying an optimal set of the paths is novel and has not been studied in the existing studies.
- We combined a belief model (i.e., SL) with DRL (i.e., policy-based DRL) to identify a set of paths based on a reward of minimum uncertainty. We developed three different DRL schemes using different types of uncertainty (i.e., vacuity, dissonance, and monosonance). To the best of our knowledge, this work is the first to take a hybrid approach for decision making with uncertain, subjective opinions by combining a belief model with DRL.
- Via extensive simulation study, we identified what type of uncertainty more significantly influences decision performance in terms of prediction accuracy, errors in predicted beliefs, and computation time.

## II. Related Work

As one of well-known belief theories, SL explicitly considered an opinion's uncertainty derived from a lack of evidence, which is called *vacuity* (i.e., ignorance). In SL, when dealing with hyper opinions, *vagueness* is also discussed in terms of how each belief mass is distinguished in an opinion [8]. Recently, other dimensions of uncertainty have been discussed, such as *dissonance* (due to conflicting evidence), monosonance (considering both vacuity and dissonance), or *consonance* (due to evidence about composite subsets of state values) [9]. Uncertainty quantification has been explored in the machine learning research. Machine/deep learning (ML/DL) research focuses on considering *aleatoric uncertainty* (i.e., uncertainty due to statistical uncertainty derived from randomness or a.k.a. data uncertainty) and *epistemic uncertainty* (i.e., a.k.a. systematic or model uncertainty derived from the inherent limitations in measurements) using Bayesian neural networks (BNNs) for computer vision applications. Aleatoric uncertainty consists of two types of uncertainty: homoscedastic uncertainty (i.e., constant errors from different inputs) and heteroscedastic uncertainty (i.e., different errors from different inputs) [7]. A Bayesian deep learning framework was presented to estimate both aleatoric and epistemic uncertainty simultaneously in regression settings (e.g., depth regression) and classification settings (e.g., semantic segmentation) [10].

Reinforcement learning (RL) refers to goal-oriented algorithms aiming to learn how to maximize along a particular dimension over many steps. It can learn a policy and then tell an agent to take an action which can achieve a final goal in the future. Although it has achieved high success in some existing studies [12, 16], it is limited to solve low-dimensional problems and hard to fix complexity issues. Deep reinforcement learning (DRL) has been proposed to address the disadvantages of RL in solving a problem with huge state space or action space [4]. Learning algorithms have a significant impact on bringing huge success to solve sequential decision making and control problems. Well-known examples include deep model-free Q-learning for general Atari game playing [13], a DRL for a driver's decision making in traffic situation [5], and continuous control decision in 3D humanoid locomotion [6]. DRL also enhances learning decision policies directly from high-dimensional inputs using end-to-end RL.

The applications using RL or DRL are huge. The examples include designing a decision making algorithm for dynamic sensor networks using RL [19], solving a path discovery problem using DRL to learn dynamics of environments for network reconfiguration [18], and development of a DRL-based recommendation system for recommending news articles [20].

To the best of our knowledge, no prior work has investigated the effect of uncertainty on a sequential decision making using DRL combined with SL.

## III. Background

In this section, we provide the overview of SL and DRL which are mainly leveraged in this work.

### A. Subjective Logic

SL defines a subjective opinion by explicitly considering the dimension of uncertainty derived from vacuity (i.e., a lack of evidence). Although SL offers the capability to formulate binomial, multinomial, and hyper opinions, we will focus on a binomial opinion in this work. For a given binomial opinion towards proposition $x$, an opinion is expressed by two belief masses (i.e., belief $b$ and disbelief $d$) and one uncertainty mass (i.e., uncertainty, $u$). For simple notations, we will drop the notation $x$ in the rest of this paper. Denote an opinion by $\omega$, which is formulated by:

$$\omega = (b, d, u, \alpha) \tag{1}$$

where $b$ and $d$ can be thought as agree vs. disagree or pro vs. con on a given proposition where $\alpha$ refers to a base rate representing a prior knowledge without commitment such as neither agree nor disagree (or neither true or false) where $b + d + u = 1$ and $b, d, u, \alpha \in [0, 1]$.

A binomial subjective opinion can be calculated as follows:

$$b = \frac{r}{r+s+W},\ d = \frac{s}{r+s+W},\ u = \frac{W}{r+s+W}. \tag{2}$$

where $r, s$ is the amount of positive and negative evidence. $W$ is an amount of uncertainty evidence where $W = 2$ refers to complete uncertainty in the initial uncertainty (i.e., $u = 0.5$ with $r = s = 1$ and $W = 2$).

In SL, two types of operators are considered: the discounting operator ($\otimes$) and the consensus operator ($\oplus$). The discounting operator ($\otimes$) [8] derives trust when there is no direct relationship between two entities. For example, when $A$ trusts $B$, $B$ trusts $C$, then we can derive $A$'s trust in $C$ based on this transitive trust relationship. To be specific, given $i$'s trust in $j$ is $w_j^i = (b_j^i, d_j^i, u_j^i, \alpha_j^i)$ and $j$'s trust in

$k$ is $w_k^j = (b_k^j, d_k^j, u_k^j, \alpha_k^j)$, we can derive $i$'s trust in $k$ as $w_k^i = (b_k^i, d_k^i, u_k^i, \alpha_k^i) = w_j^i \otimes w_k^j$, which is given by:

$$b_k^i = b_j^i \otimes b_k^j = b_j^i b_k^j,\ d_k^i = d_j^i \otimes d_k^j = b_j^i d_k^j \qquad (3)$$
$$u_k^i = u_j^i \otimes u_k^j = d_j^i + u_j^i + b_j^i u_k^j,\ \alpha_k^i = \alpha_j^i \otimes \alpha_k^j = \alpha_k^j.$$

The consensus operator ($\oplus$) offers the capability to combine two different opinions (i.e., $w_k^i$ and $w_k^j$) towards a same entity where the two opinions are independent to each other [8]. That is, the combined opinion, $w_k^i \oplus w_k^j$, can be obtained by:

$$b_k^i \oplus b_k^j = \frac{b_k^i u_k^j + b_k^j u_k^i}{\zeta},\ d_k^i \oplus d_k^j = \frac{d_k^i u_k^j + d_k^j u_k^i}{\zeta} \qquad (4)$$
$$u_k^i \oplus u_k^j = \frac{u_k^i u_k^j}{\zeta},\ \alpha_k^i \oplus \alpha_k^j = \alpha_k^i.$$

where $\zeta = u_j^i + u_k^j - u_j^i u_k^j > 0$.

We use these two operators in a scenario that a decision maker needs to select a set of paths for collecting opinions in order to predict unknown opinions.

### B. Policy Based Deep Reinforcement Learning

RL adopts the standard Markov Decision Process (MDP) formalism. An MDP is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, which consists of a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a reward function $\mathcal{R}(s, a)$ with each state $s \in \mathcal{S}$ and each action $a \in \mathcal{A}$, $\mathcal{P}(s'|a, s)$ is the transition probability matrix , and a discount factor $\gamma$. According to the agents taking an action $a \in \mathcal{A}$ in a state.

The agent of reinforcement learning is represented as a policy network $\pi_\theta(s, a) = p(a|s, \theta)$ which means that the policy $\pi$ is the probability of taking action $a$ when at state $s$ and the neural network parameters are $\theta$. Now suppose we are in some state $s_t$, receive reward $r_t$, sample action $a_t \sim \pi_\theta(s_t, a)$ and $s_{t+1}$ is a sample from $\mathcal{P}(s'|s_t, a_t)$, the optimization problem for finding $\theta$ to maximize the sum of rewards is:

$$J_\theta(s) = E\left[\sum_{t=1}^{\infty} \gamma^t r_t | s_{t+1} \sim \mathcal{P}(s'|s_t, a_t), s_1 = s\right] \qquad (5)$$

where $\theta$ can be updated by using stochastic gradient descent. Compared to deep Q-learning Network (DQN) [13] which is a value-based DRL method, policy-based DRL methods are more appropriate for our uncertainty-based decision making scenario. As we present experiments for graph data in the Section V, the action space in a complexity relation graph can be very large. This may result in poor convergence properties of DQN. In addition, stochastic policy can be learned in the policy network which prevents the agent from being stuck at an intermediate state. However, the value-based methods try to learn a greedy policy which cannot solve the problem mentioned above. Therefore, in this work, we use the policy-based DRL to improve our SL-based opinion inference model.

### C. Dimensions of Uncertainty in SL

The concept of uncertainty has been discussed differently depending on domains [9, 15]. In this work, we adopt the concept of uncertainty and its variety based on SL which will be applied in developing a DRL-based decision making algorithm when the input is a large-scale graph network data. Since we use the three dimensions of uncertainty in SL as a reward in applying DRL to identify an optimal path to collect beliefs with minimum uncertainty, we limit our discussion on *vacuity*, *dissonance*, and *monosonance* for brevity.

*Vacuity* refers to a lack of evidence, meaning that uncertainty is introduced because of no or insufficient information. Note that in Section III-A, $u_v$ is denoted by $u$ in the original SL before being extended its uncertainty in [9] because the original SL only dealt with vacuity [8].

*Dissonance* reflects a situation when an analyst holds contradicting beliefs for a given proposition (e.g., $b = 0.5, d = 0.5, u_v = 0$). The dissonance can be also measured when an opinion is a hyper opinion where *vagueness* exists for a composite belief with multiple belief masses (i.e., a belief consists of multiple beliefs, called a composite belief). However, as we consider a binomial opinion (i.e., belief, disbelief, and vacuity), the dissonance captures the distance between belief and disbelief. We denote *dissonance* [9] by $u_d$, estimated by:

$$u_d = \sum_{k=1}^{K} \left( \frac{b_k \sum_{j=1, j\neq k}^{K} b_j \mathrm{Bal}(b_j, b_k)}{\sum_{j=1, j\neq k}^{K} b_j} \right) \qquad (6)$$

where $b_k$ and $b_j$ are the belief masses of the $k$-th and $j$-th categories (i.e., belief and disbelief in a binomial opinion), respectively. $K$ is the total number of beliefs, and the relative mass between a pair of belief masses $b_j$ and $b_k$ is expressed by the so called *balance*, $\mathrm{Bal}(b_j, b_k)$, which is obtained by:

$$\mathrm{Bal}(b_j, b_k) = \frac{1 - |b_j - b_k|}{(b_j + b_k)} \qquad (7)$$

Higher *dissonance* indicates higher uncertainty due to conflicting beliefs.

*Monosonance* refers to the degree that the opinion is solely supporting a singleton belief under low vacuity. We denote *monosonance* by $u_m$, which is estimated by:

$$u_m = \sum_{k=1}^{K} \left( \frac{b_k \sum_{j=1, j\neq k}^{K} b_j (1 - \mathrm{Bal}(b_j, b_k))}{\sum_{j=1, j\neq k}^{K} b_j} \right) \qquad (8)$$

Notice that the imbalance between beliefs (i.e., $1 - \mathrm{Bal}(b_i, b_j)$) increases, $u_m$ increases. Therefore, lower *monosonance* refers to high uncertainty due to conflicting beliefs.

## IV. Uncertainty-based Decision Making with DRL

In this section, we define a decision making problem based on different types of uncertainty and then present a deep reinforcement learning framework to solve it.

### A. Problem Scenario & Formulation

Our decision making problem is formulated by:

**Given:**

- $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ is an input network where $\mathbb{V} = \{1, \cdots, N\}$ is a set of nodes and $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ is a set of edges.
- $\boldsymbol{\omega}_{\mathbb{L}} = [\omega_i]_{i \in \mathbb{L}}$, where let $\mathbb{L} \subset \mathbb{E}$ be a subset of edges that have subjective opinions, and $\omega_i = (b_i, d_i, u_i, \alpha_i)$ be edge $i$'s subjective opinion.

**Predict:**

- $\boldsymbol{\omega}_{\mathbb{E}\setminus\mathbb{L}} = [\omega_i]_{i \in \mathbb{E}\setminus\mathbb{L}}$, a vector of unknown opinions.

Given a predicted opinion, the expected belief and disbelief [8] can be estimated as:

$$E_b = b + \alpha \cdot u, \quad E_d = d + \alpha \cdot u \tag{9}$$

where $E_b$ and $E_d$ are belief and disbelief that a decision maker actually uses for decision making when the perceived uncertainty is $u$ and the prior belief is $\alpha$.

### B. SL-based Deep Reinforcement Learning

In this section, we describe our proposed deep reinforcement learning framework for opinion inference as a decision making problem based on different types of uncertainty. The key problem is to identify paths based on the relationships between opinions in order to infer unknown opinions using the discounting and consensus operators in SL. We formulate this problem as a sequential decision making problem where a decision maker is an agent conducting learning based on DRL technique. Now we discuss how the agent makes a decision based on DRL as follows.

**External Learning Environment**: This environment specifies the dynamics of the interaction between an agent and an input graph. This environment is modeled as a Markov decision process (MDP). The MDP defines a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ where $\mathcal{S}$ is the state space, $\mathcal{A} = \{a_1, a_2, \cdots, a_n\}$ is the set of all available actions, $\mathcal{P}(S_{t+1} = s' | S_t = s, A_t = a)$ is the transition probability matrix from time $t$ to time $t+1$, and $R(s, a)$ is the reward function of every $(s, a)$ pairs.

**DRL Agent**: This agent is represented as a policy network $\pi(s, a; \theta) = p(a|s; \theta)$ where it maps the state vector to a stochastic policy. The parameter $\theta$ can be updated by a gradient descent method.

**Action**: Given a test edge $e_i = (n_s, n_d)$, the agent aims to find the most related paths from source node $n_s$ to target node $n_d$. Beginning with $n_s$, the agent uses the policy network to learn the most promising relation in order to extend its path at each step until reaching $n_d$. To keep the output dimension of the policy network consistent, the action space is defined as all nodes in a graph. After taking action $a_t$ at time $t$, we arrive at current node $n_t$ and update current opinion $\omega_{t+1} \longleftarrow \omega_t \otimes \omega_{a_t}$.

**States**: To keep the input dimension of the policy network consistent, we let the state $s_t = (f_t, f_d - f_t)$ as input for all possible states, where $f_t$ denotes the embedding of current node and $f_d$ denotes the embedding of target node. Here we use a random walk scheme to generate the embedding feature

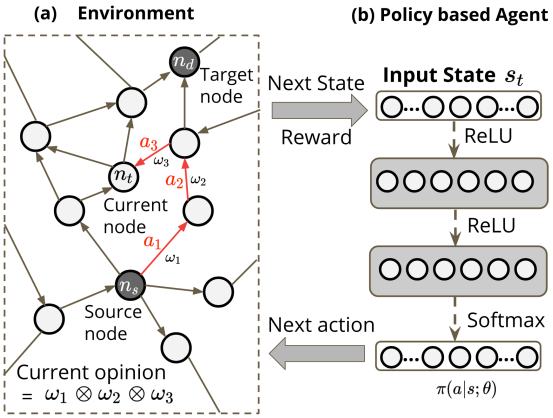



Fig. 2: An overview of the proposed DRL-based model: (a) The environment modeled by the MDP. The red arrows (i.e., actions taken) show the path identified by the DRL agent. (b) The structure of the policy network agent. At each step, by interacting with the environment, the agent learns a relation action to extend the opinion paths.

for each node, from the initial node $t$, random walk a fixed steps $P$ and concatenate all opinion, $f_t \in \mathbb{R}^{4P}$. At the initial state $f_t = f_s$.

**Reward**: For the agent to make an effective decision using the most predictive paths that provide the final opinion with high confidence, we use uncertainty as a reward ($r_{\text{uncertainty}}$) to maximize the prediction of effective decision in our proposed DRL framework as follows:

$$r_{\text{uncertainty}} = \begin{cases} 1 - u & \text{if } u = u_v \text{ or } u = u_d \\ u & \text{if } u = u_m. \end{cases} \tag{10}$$

To make the agent to find the target node $n_d$ efficiently, we set a global reward:

$$r_{\text{global}} = \begin{cases} +1 & \text{if path can reach } n_d \\ -1 & \text{otherwise.} \end{cases} \tag{11}$$

The agent is given an offline positive reward $+1$ if it reaches the target after a sequence of actions; otherwise, given a negative reward $-1$. The total reward for transition $s_{t+1} \leftarrow s_t$ is:

$$r_t = r_{\text{uncertainty}} + r_{\text{global}} \tag{12}$$

**Policy Network**: We use a fully-connected neural network to parameterize the policy function $\pi(s; \theta)$ that maps state vector $s$ to a probability distribution over all possible actions. The neural network consists of two hidden layers with each being followed by a rectifier nonlinearity layer (ReLU). The output layer is normalized using a softmax function (see Fig. 2)

### C. Training Pipeline

Following Eq. (5), in order to maximize the expected cumulative reward, we update the nerual network parameters $\theta$ by using Monte-Carlo Policy Gradient [17]. Therefore, Eq. (5) can be rewritten as:

$$J(\theta) = \mathbb{E}_{a \sim \pi(a|s;\theta)}(\sum_t r_t) = \sum_t \sum_{a \in \mathcal{A}} \pi(a|s_t; \theta) r_t \tag{13}$$

where $J(\theta)$ is an expected total reward for one episode. For each path $p$ with a sequence of relations $p_{n_s} \rightarrow ... \rightarrow p_{n_t} \rightarrow \cdots \rightarrow p_{n_d}$ from source node $n_s$ to target node $n_d$, we provide a reward for each step of a successful episode according to Eq. (10). The approximated gradient for updating the policy network is shown below:

$$\begin{aligned} \nabla_\theta J(\theta) &= \sum_t \sum_{a \in \mathcal{A}} \pi(a|s_t;\theta) \nabla_\theta \log \pi(a|s_t;\theta) r_t \\ &\approx \nabla_\theta \sum_t \log \pi(a = p_{n_t}|s_t;\theta) r_t \end{aligned} \tag{14}$$

Starting from the source node $n_s$, the agent picks an action according to the stochastic policy $\pi(a|s)$, which is a probability distribution over all actions, to extend its promising path. Since the agent is following a stochastic policy, the agent will not get stuck by repeating a wrong step. In addition, we set an upper bound *max_length* $L$ for the episode length in order to improve efficiency in the training procedure. Therefore, the episode will early stop if the agent cannot reach the target node $n_d$ in *max_length* steps, and give a penalty. After each episode, the policy network will be updated by Eq. (14). Algorithm 1 provides the details. In practice, $\theta$ is updated by using the Adam Optimizer [11] with L2 regularization.

**Algorithm 1:** DRL Uncertainty-based Path Selection for Evidence Collection

**Input:** $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ and $\{\omega_i\}_{i \in \mathbb{L}}$
**Output:** $\{\omega_i\}_{i \in \mathbb{E} \setminus \mathbb{L}}$

Initialize policy network with random weight $\theta$;
Initialize episode length $l = 0$;
**for** *episode = 1, ..., V* **do**
  Initialize source node $n_s$, target node $n_t$ and state $s_1$.
  **for** $t$ *= 1, ..., L* **do**
    Randomly select action $a \sim \pi(a|s_t)$
    Calculate current opinion $\omega_t$ via Eq. (3)
    Observe reward $r_t$, next state $s_{t+1}$
    **if** $r_{global} = -1$ **then**
      Save $(s_t, a)$ to $\mathcal{M}_{neg}$
    **if** $n_t = n_d$ **then**
      *Success* and *break*
    Penalize failed steps and update $\theta$ by $g \propto \nabla_\theta \sum_{\mathcal{M}_{neg}} log\pi(a = p_{n_t}|s_t;\theta)(-1)$
    **if** *Success* **then**
      $r_t = r_{\text{global}} + \lambda \cdot r_{\text{uncertainty}}$
      update $\theta$ via: $g \propto \nabla_\theta \sum_t log\pi(a = p_{n_t}|s_t;\theta) r_t$
Infer a test edge by selecting actions with $\pi(a|s,\theta)$.
**return** $\{\omega_i\}_{i \in \mathbb{E} \setminus \mathbb{L}}$

## V. Experimental Results & Analysis

### A. Experimental Setup

*1) Semi-synthetic Epinions dataset:* We use the *Epinions* dataset [1] representing a who-trust-who in an online social network. This is a directed network consisting of 47,676 users (i.e., vertices) and 467,468 relationships (i.e., edges). As there are no ground truth opinions available from the dataset, we use a benchmark simulation model [14] to generate synthetic opinions. The simulation model has the following main steps:

- **Initialization**: 10% of the edges are uniformly selected at random and set the trust of edges to 1's meaning that $i$ trusts $j$ where $i$ and $j$ are users in a given directed network.
- **Exploration**: 1,000 exploration steps are performed to update trust relationships based on the following trust rule:

$$\text{Trust}(a,b) = 1 \wedge \text{Trust}(b,c) = 1 \rightarrow \text{Trust}(a,c) = 1. \tag{15}$$

  The exploration step is used to generate synthetic trust observations on the edges of the network. For each exploration step, we uniformly select one edge at random, identify the rule instances associated with this edge, and generate one observation of the edge (i.e., 0 or 1) based on the probability of the rule instances in which 1 and 0 refer to trust and distrust, respectively. By repeating the exploration step 1,000 times, we generate a realization of trust relationships on the edges in the network, in which the observations of 1,000 randomly selected edges are generated while the other edges do not have any observations in this realization. We then conduct the $2^{nd}$ realization based on the previous one by randomly selecting 5% of the edges and swapping their most recent observations from 1 to 0 or from 0 to 1 that are considered as their new trust observations at the current realization. 1,000 exploration steps are conducted to generate observations to make them consistent with the trust rule. Following this procedure, we generate $2^{nd}, \cdots T^{th}$ realizations.
- **Performance evaluation**: After conducting the $T$ realizations, each edge then has up to $T$ trust observations and its opinion can be estimated based on its trust observations. We consider a set of candidate values of $T \in \{6, 10, 15, 25, 38\}$ corresponding to different uncertainty ranges, as explained below. In order to conduct performance evaluation for different network sizes, we randomly sample sub-networks with the number of nodes $N \in \{500, 1000, 5000, 10000\}$ from the original *Epinions* network, respectively. We randomly selected 20% edges for testing.

*2) Road traffic datasets:* We collected live road traffic data from June 1, 2013 to March 31, 2014 across one cities from INRIX [2], Philadelphia (PA), as summarized in Table I. The raw INRIX dataset collected live traffic speed information from trucks per five-minute interval. A road link has a live speed measurement at a specific time interval if it has at least one truck traversing this link at the time interval; otherwise, it will be a missing speed value. In addition, the reference speed information refers to each road link per hour interval where the reference speed means the "non-congested free flow speed" for each road segment [3]. It is calculated based upon the $85^{th}$ percentile of the measured speeds for all time periods over a few years where the reference speed serves as a threshold separating two traffic states, *congested* vs. *non-congested*. The road traffic dataset has 43 weeks in total. An hour is represented by a specific combination of hours of a day ($h \in \{6, 9, 12, \cdots, 21\}$), days of a week ($d \in \{1, 2, 3, 4, 5\}$), and weeks ($w \in \{1, 2, \cdots, 43\}$): $(h, d, w)$. We only considered work days from Monday ($d = 1$) to Friday ($d = 5$) and hours from 6AM ($h = 6$) to 9PM ($h = 21$).

TABLE I: Description of the two real-world datasets.

| Dataset name | # nodes | # edges | # weeks | # snapshots (hours) in total |
|---|---|---|---|---|
| **Epinions** | 47,676 | 477,468 | - | - |
| **Philadelphia** | 603 | 708 | 43 | 3440 |

**Ground truth opinions (beliefs and uncertainties) of training and testing edges in traffic dataset**. For traffic dataset, the opinion of a specific (training or testing) link $s$ at an hour $(h, d, w)$ is estimated based on the observations of the same hour in previous $T$ weeks $\{x_{s,h,d,w}, x_{s,h,d,w-1}, \cdots, x_{s,h,d,w-T+1}\}$ as the evidence, where $x_{s,h,d,w}$ refers to the congestion observation (i.e., 0 or 1) of the link $s$ at hour $(h, d, w)$ and $T$ refers to a predefined time window size. Note that some of the observations are not observed, as only a subset of the links were traversed by the delivery trucks. Denote by $T_s$ the number of observations within the $T$ weeks for the link $s$ and $0 \leq T_s \leq T$. The belief, disbelief, and vacuity variables $b_s$, $d_s$, and $u_s$ of a specific link $s$ are estimated by:

$$\begin{aligned} b_s &= \left(\sum_{t=0}^{T-1} x_{s,h,d,w-t} - W \cdot \alpha\right) / (T_s + W) \\ d_s &= \left(T - \sum_{t=0}^{T-1} x_{s,h,d,w-t} + W \cdot \alpha\right) / (T_s + W) \\ u_s &= W/(T_s + W), \end{aligned} \tag{16}$$

where we set the non-informative prior weight (i.e., an amount of uncertain evidence with $W = 2$) and the base rate (i.e., prior knowledge with $\alpha = 0.5$). As $T$ is the maximum number of possible observations a link can have within a time window of size $T$, it can be used to calculate a lower bound on the uncertainty of a link as $W/(T + W)$, and the upper bound will be $100\%$.

*3) Parameter settings:* The main parameters for all the datasets include $T$ (i.e., an observation time window size) and $P$ (i.e., a maximum number of paths to infer an unknown opinion). We tested different window sizes $T \in \{6, 10, 15, 25, 38\}$ corresponding to the different vacuity ranges. For all datasets, we only choose 20% test edges.

*4) Performance metrics:* Our experimental analysis focuses on the performance comparative study of our proposed schemes and a baseline scheme based on the following metrics: *Expected Belief MSE* (EB-MSE), precision accuracy, and computation time (in sec.).

EB-MSE and precision accuracy ($P_A$) are computed by:

$$\text{EB-MSE}(\boldsymbol{\omega}_{\mathbb{E}\setminus\mathbb{L}}) = \frac{1}{M} \sum\nolimits_{i \in \mathbb{E}\setminus\mathbb{L}} \left| E_{b_i} - E_{b_i^\star} \right| \tag{17}$$

$$P_A = \frac{1}{M} \sum\nolimits_{i \in \mathbb{E}\setminus\mathbb{L}} \phi(E_{b_i}, E_{b_i^\star}) \tag{18}$$

where $E_{b_i}$ or $E_{b_i^\star}$ refers to the predicted or true expected belief of a target test edge $i$, respectively. $\phi(E_{b_i}, E_{b_i^\star}) = 1$ when $E_{b_i} \leq 0.5, E_{b_i^\star} \leq 0.5$ or $E_{b_i} > 0.5, E_{b_i^\star} > 0.5$; otherwise $\phi(E_{b_i}, E_{b_i^\star}) = 0$. That is, $\phi(E_{b_i}, E_{b_i^\star}) = 1$ represents a correct decision while $\phi(E_{b_i}, E_{b_i^\star}) = 0$ means an incorrect decision. Computation time reflecting algorithmic complexity is measured using time unit (sec.). Note that smaller EB-MSE, larger $P_A$, and smaller computation time are more desirable.

*5) Comparing schemes:* In our experiments, we compare our proposed schemes with SL as a baseline model [8]. Our proposed scheme combines SL and DRL while using vacuity, monosonance, or dissonance as a reward (see Eq. (10)). We denote them by `SL-DRL-V`, `SL-DRL-M`, and `SL-DRL-D`, respectively.

*6) Parameter Tuning:* SL only has one hyper parameter that is the maximum length of its independent paths. Our proposed DRL-based methods, `SL-DRL-V`, `SL-DRL-M`, and `SL-DRL-D`, have three hyper parameters: $\eta$ (a learning rate), dropout (a parameter to lower complexity), and $\lambda$ (a trade-off parameter). We set $\lambda = 10$, $\eta = 0.0005$, and dropout = 0.5 for all the experiments.

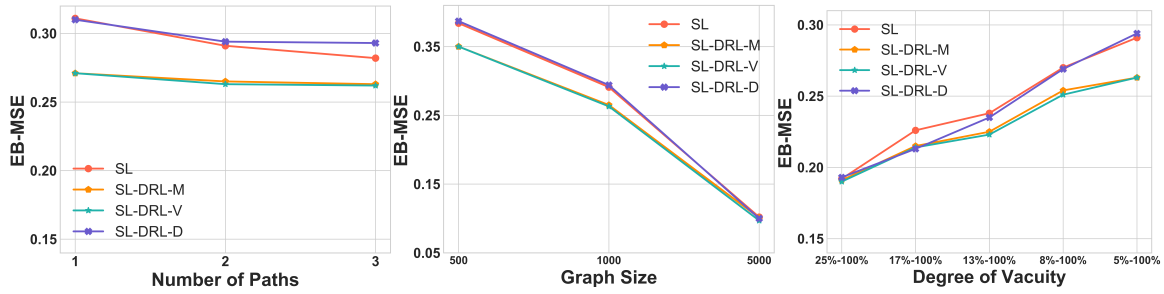
### *B. Experimental Results based on Semi-Synthetic Datasets*

Figs. 3 and 4 show the comparative analysis of our proposed schemes (i.e., `SL-DRL-V`, `SL-DRL-M`, `SL-DRL-D`) and a baseline scheme, `SL`, in terms of EB-MSE and precision accuracy ($P_A$), respectively, under the semi-synthetic *Epinions* dataset.

Figs. 3 (a) and 4 (a) demonstrate the effect of the number of paths ($N_P$) on EB-MSE and $P_A$ across all schemes. Obviously, `SL-DRL-V` and `SL-DRL-M` outperform among all, except that they perform comparably to `SL` for some of the settings (i.e., $N_P = 2, 3$) for $P_A$. We notice that both `SL-DRL-V` and `SL-DRL-M` have the flat tendency. Since DRL-based approaches select each path among multiple paths available based on the given reward, they select the best paths that can generate the best decision accuracy. This reward-based path selection leads to little difference even if more paths are used (i.e., even using a single path can lead to the best decision).

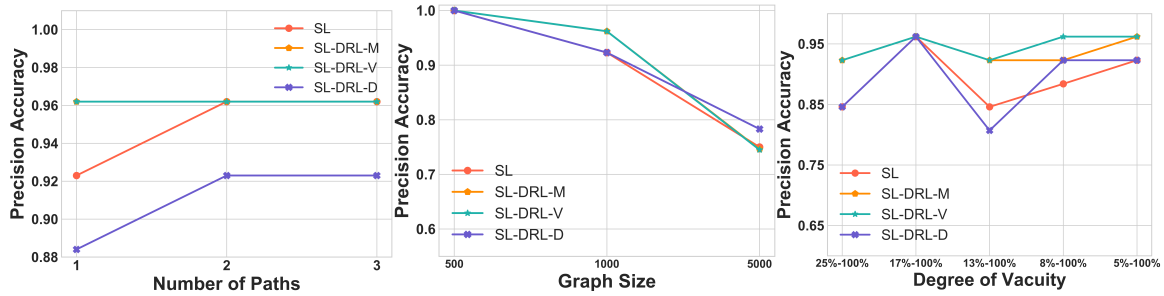
Figs. 3 (b) and 4 (b) show the effect of a network size $N$ on EB-MSE and $P_A$ under all comparing schemes. It is clear that `SL-DRL-V` and `SL-DRL-M` perform the best among all on both metrics. We observe that as a graph size (i.e., a number of nodes in a network) grows, both EB-MSE and $P_A$ decrease in `SL-DRL-V` and `SL-DRL-M`. This implies that for larger network data, it is less likely to predict the decision accurately due to relatively longer paths used, leading to higher vacuity ($u_v$) in the resulting opinions (i.e., a longer trust chain leads to more decay, as observed in the discounting operator in Eq. (3)).

Figs. 3 (c) and 4 (c) demonstrate that `SL-DRL-V` and `SL-DRL-M` outperform among all in terms of EB-MSE and $P_A$, respectively, with respect to varying the ranges of vacuity, including $[25\%, 100\%]$, $[17\%, 100\%]$, $[13\%, 100\%]$, $[8\%, 100\%]$ and $[5\%, 100\%]$. As the degree of vacuity increases (i.e., the smaller lower bound), while `SL-DRL-V` and `SL-DRL-M` maintain $P_A$, `SL` and `SL-DRL-D` are fluctuating in $P_A$. In `SL-DRL-V` and `SL-DRL-M`, the paths with minimum vacuity are considered; so even if there are paths with high vacuity, they can selectively pick the paths with minimum vacuity, leading to fairly constant performance across different degrees of vacuity. However, `SL` selects a path randomly and `SL-DRL-D` selects a path with minimum dissonance, which does not guarantee the minimum vacuity in both schemes, their prediction accuracy can be fluctuating. Recall that our




(a) EB-MSE under varying the number of paths with $T = 38, N = 1000$ (b) EB-MSE under varying the size of a graph with $T = 38, N_P = 2$ (c) EB-MSE under varying degree of vacuity with $N_P = 2, N = 1000$

Fig. 3: Performance comparison: EB-MSE under the semi-synthetic network based on the *Epinions* dataset.




(a) $P_A$ under varying the number of paths with $T = 38, N = 1000$ (b) $P_A$ under varying the size of a graph with $T = 38, N_P = 2$ (c) $P_A$ under varying the the degree of vacuity with $N_P = 2, N = 1000$

Fig. 4: Performance comparison: Precision accuracy ($P_A$) under the semi-synthetic network based on the *Epinions* dataset.

expected belief and disbelief (i.e., $E_b$ and $E_d$) reflect how to interpret uncertainty. Therefore, the fluctuating vacuity on the selected paths can naturally lead to the zigzag patterns of $P_A$.

Overall, the performance order in EB-MSE and $P_A$ on the *Epinion* dataset is: SL−DRL−V≈ SL−DRL−M > SL > SL−DRL−D, which demonstrate that vacuity and monosonance play an important role in opinion inference. On the other hand, SL−DRL−D performs the worst. This is because dissonance in SL [9] is estimated based on the relative difference between belief and disbelief. That is, even if there is a high vacuity, the minimum dissonance can be found. However, the precision accuracy is estimated based on the expected belief or disbelief (i.e., $E_b$ or $E_d$) using vacuity ($u_v$) and the base rate ($\alpha$). Hence, low dissonance does not necessarily result in better decision performance while high monosonance covers both low vacuity and low dissonance. Fig. 5 shows the log computation

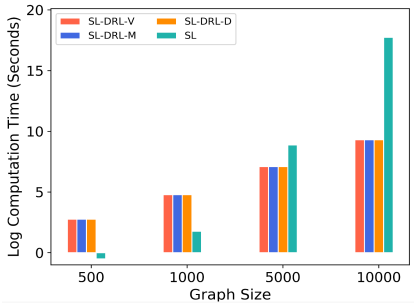


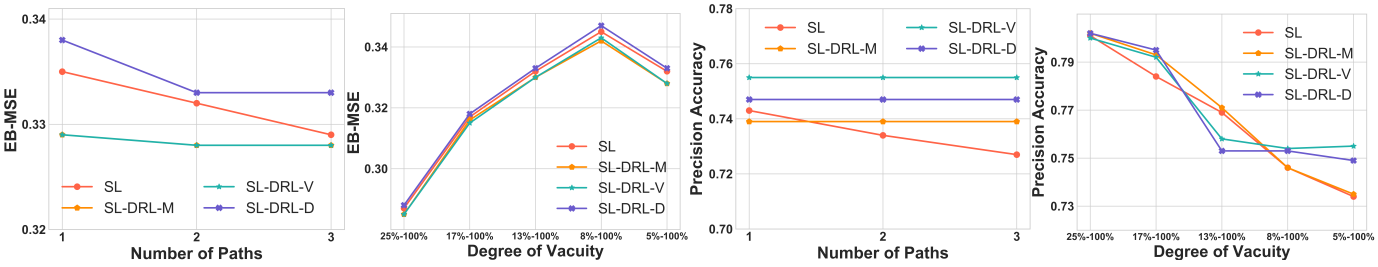

Fig. 5: Comparison of computation time on the *Epinion* dataset with $T = 38, N_P = 2$.

times as the number of nodes increases. Except SL whose computation time increases exponentially with network size increasing, the other schemes almost scale linearly with respect to the network size. SL−DRL−V and SL−DRL−M are the most efficient schemes among all. The overall performance order in computation time of all comparing schemes is: SL−DRL−V ≈ SL−DRL−M ≈ SL−DRL−D > SL.

### C. Experimental Results based on Real-World Datasets

Fig. 6 shows the comparative analysis of our proposed schemes and SL in terms of EB-MSE and $P_A$ under the read world dataset. Fig. 6 (c) and (d) demonstrates that SL−DRL−V and SL−DRL−M outperform among all based on EB-MSE and $P_A$ with respect to varying the ranges of uncertainty mass (i.e., vacuity), including $[25\%, 100\%]$, $[17\%, 100\%]$, $[13\%, 100\%]$, $[8\%, 100\%]$ and $[5\%, 100\%]$. As the lower bound of the vacuity range decreases, in both SL−DRL−V and SL−DRL−M, EB-MSE increases while $P_A$ decreases. This is quite in contrast with what we observed with the *Epinion* dataset. In the road traffic dataset for PA, as shown in Table I, the network itself is very sparse compared to the *Epinion* dataset. We suspect that a lack of available paths may lead to poor performance, which is further pronounced when the degree of vacuity increases.

Fig. 6 (a) and (b) demonstrate the effect of the number of paths on EB-MSE and $P_A$ across all comparing schemes. Obviously, SL−DRL−V outperforms among all. And both SL−DRL−V and SL−DRL−M show a flat tendency, demonstrating that our DRL-based model with a reward of vacuity




(a) EB-MSE under varying the number of paths with $T = 38$

(b) EB-MSE under varying the degree of vacuity with $N_p = 2$

(c) $P_A$ under varying the number of paths with $T = 38$

(d) $P_A$ under varying the degree of vacuity with $N_p = 2$

Fig. 6: Performance comparison: EB-MSE and prediction accuracy ($P_A$) on the road traffic dataset (real world dataset).

or monosonance can find the most useful path(s) for opinion inference, resulting in high $P_A$.

## VI. Conclusion & Future Work

In this work, we proposed a set of uncertainty-based decision rules to infer unknown subjective opinions by leveraging a deep reinforcement learning (DRL) technique when a uncertain, subjective opinion is formulated based on Subjective Logic on graph network data. We considered three different types of uncertainty, including vacuity, monosonance, and dissonance, to be used as a reward in DRL with the aim of identifying the most useful opinion paths that can lead to the best decision making on graph network data.

The **key findings** from this study are summarized as: (1) vacuity is the most important factor that can significantly impact decision accuracy; (2) monosonance considers both vacuity and dissonance where the effect of vacuity is more dominant than that of dissonance, resulting in the similar performance to the vacuity-based DRL; and (3) although dissonance captures the discrepancy between belief and disbelief in a given binomial opinion in SL, low dissonance does not necessarily mean low vacuity and so may not lead to the best decision performance.

In our future work, we plan to conduct the extension of our proposed work via meta reinforcement learning to be more efficient and effective in optimizing decision performance.

## Acknowledgments

This work is partially supported by ARL's Competitive Basic Research Program under Computational and Information Sciences Directorate and by the US Army Research Office under grant number W911NF1720129. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ARL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

## References

[1] "Epinions," http://www.trustlet.org/downloaded_epinions.html.
[2] "Inrix," http://inrix.com/publicsector.asp.
[3] "Reference speed for congestion evaluation," http://www.inrix.com/scorecard/methodology.asp/.
[4] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *arXiv preprint arXiv:1708.05866*, 2017.
[5] Z. Bai, B. Cai, W. Shangguan, and L. Chai, "Deep reinforcement learning based high-level driving behavior decision-making model in heterogeneous traffic," *arXiv preprint arXiv:1902.05772*, 2019.
[6] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *ICML*, 2016, pp. 1329–1338.
[7] Y. Gal, "Uncertainty in deep learning," *University of Cambridge*, 2016.
[8] A. Jósang, *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer Publishing Company, 2016.
[9] A. Josang, J.-H. Cho, and F. Chen, "Uncertainty characteristics of subjective opinions," in *FUSION*. IEEE, 2018, pp. 1998–2005.
[10] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *NIPS*, 2017, pp. 5574–5584.
[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[12] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *Proceedings. ICRA'04. 2004*, vol. 3. IEEE, 2004, pp. 2619–2624.
[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.
[14] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *International semantic Web conference*. Springer, 2003, pp. 351–368.
[15] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *NIPS*, 2018, pp. 3183–3193.
[16] S. Singh, D. Litman, M. Kearns, and M. Walker, "Optimizing dialogue management with reinforcement learning: Experiments with the njfun system," *Journal of Artificial Intelligence Research*, vol. 16, pp. 105–133, 2002.
[17] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.
[18] W. Xiong, T. Hoang, and W. Y. Wang, "Deeppath: A reinforcement learning method for knowledge graph reasoning," *arXiv preprint arXiv:1707.06690*, 2017.
[19] D. Zhang and H. Ma, "A q-learning-based decision making scheme for application reconfiguration in sensor networks," in *2007 11th International Conference on Computer Supported Cooperative Work in Design*. IEEE, 2007, pp. 1122–1127.
[20] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "Drn: A deep reinforcement learning framework for news recommendation," in *WWW*, 2018, pp. 167–176.