How Can AI Reason Your Character?

Dongsu Lee [1] **Minhae Kwon** [1]

Abstract

Inference of decision preferences through others' behavior observation is a crucial skill for artificial agents to collaborate with humans. While some attempts have taken in this realm, the inference speed and accuracy of current methods still need improvement. The main obstacle to achieving higher accuracy lies in the stochastic nature of human behavior, a consequence of the stochastic reward system underlying human decision-making. To address this, we propose the development of an *instant inference network* (IIN), surmising the partially observable agents' *stochastic character*. The agent's character is parameterized by weights assigned to reward components in reinforcement learning, resulting in a singular policy for each character. To train the IIN for inferring diverse characters, we develop a *universal policy* comprising a set of policies reflecting different characters. Once the IIN is trained to cover diverse characters using the universal policy, it can return character parameters instantly by receiving behavior trajectories. The simulation results confirm that the inference accuracy of the proposed solution outperforms state-of-the-art algorithms, despite having lower computational complexity.

1. Introduction

Inferring others' decision preferences through their behavior cues is a pivotal skill in the context of social decision-making. In particular, endowing artificial intelligence (AI) agents with this ability is critical for effective human-AI collaboration, where the AI is required to instantly determine the optimal action to support human partners who are encountered on the fly.

Consider the case of autonomous vehicles and human drivers coexisting. These human drivers have different driving preferences (Park et al., 2020; Griesche et al., 2016), e.g., one who prioritizes speed and another who values safety by keeping a safe distance from neighboring vehicles. In such social contexts, the ability to infer instantly and accurately the driving character of human drivers is crucial for autonomous vehicles to determine their upcoming actions. To parameterize the behavior preference of a decision-maker systematically, we introduce the *character* concept of an agent in this paper. We define the character of an agent as a weight vector on the reward components in reinforcement learning.

Individual behavioral patterns can be described as a character based on the different weights assigned to reward components that determine what is more valuable. These characters arise from accumulated past experiences, intrinsic motivation, and extrinsic reward signals (Craik, 1967). Recent advances in neuroscience have revealed that character exhibits *stochasticity* due to epistemic and aleatoric uncertainties (Chater et al., 2020; McNamee & Wolpert, 2019; Bressloff et al., 2016). To infer the character of humans, the inference model should be capable of handling stochastic character, which arises from the stochasticity in the agent's reward function.

Several approaches, including Theory of mind (ToM) (Nguyen & Gonzalez, 2020; Rabinowitz et al., 2018), instant-based learning (IBL) (Gonzalez & Quesada, 2003), and inverse reinforcement learning (IRL) (Hantous et al., 2022; Chan & van der Schaar, 2020), have been developed to infer an agent's specific character from their behavior. While these approaches can infer some internal parameters, they have two critical limitations in a social context. Firstly, existing methods build a single inference network for a single character. This means that we need multiple networks, as many as the character types. Secondly, most existing works do not allow real-time inference since they use iterative optimization approaches. To address these limitations, we propose an IIN that can infer a wide range of characters.

In this study, our goal is to build two agents: (1) partially observable Markov decision process (POMDP) agent that embeds stochastic character, trained with partial noisy observation; and (2) a meta-agent equipped with the IIN. The meta-agent collects observation-action trajectories of a target agent, and the trajectories are then inputted into the IIN.

[1]Soongsil University, Seoul, Korea. Correspondence to: Minhae Kwon <minhae@ssu.ac.kr>.

ICML (International Conference on Machine Learning) Workshop on Artificial Intelligence and Human-Computer Interaction, Honolulu, Hawaii, USA.

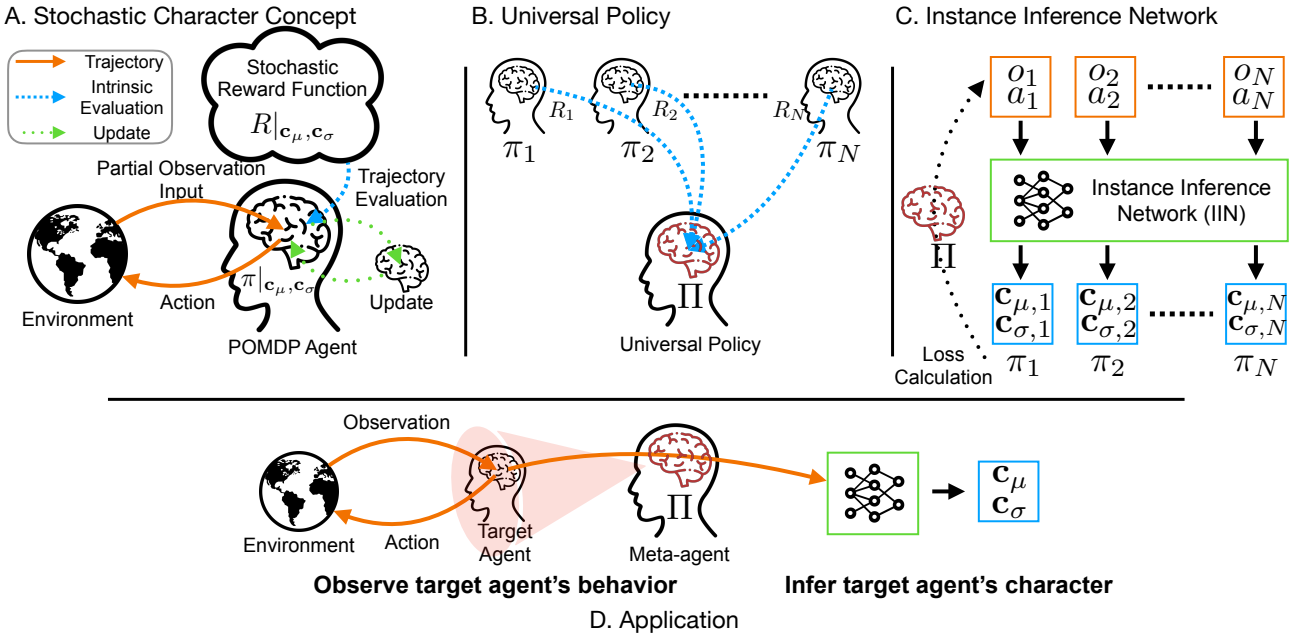

Figure 1. **A**. The policy π of a POMDP agent captures stochastic character $\mathbf{c}_\mu, \mathbf{c}_\sigma$ by using own reward function R. **B**. The universal policy $\Pi = \{\pi_1, \pi_2, \cdots, \pi_N\}$ comprises all probable policies based on each character. **C**. IIN's input and output are observation-action pair and stochastic character, respectively. Universal policy identifies the ground truth character of each trajectory. **D**. In deployment phase, the meta agent can collect the observation action pairs observing world and behavior; then the IIN instantly infers the agent's character.

The IIN returns the inferred character distribution of the target agent.

To build the POMDP agent with stochastic character, we formalize a reinforcement learning problem under partially observable conditions (Figure 1.**A**). The agent's character is determined by weights assigned to multiple reward components drawn from a distribution that induces stochasticity. The trained policy exhibits behavior preferences aligned with the POMDP agent's stochastic character. Next, we train the IIN of the meta-agent using a universal policy that generalizes over a wide range of character spaces (Figure 1.**B**). The universal policy generates training data for the IIN by providing observation-action trajectories as inputs and the mean and variance of the character distribution as outputs. We design the IIN with long short-term memory (LSTM) (Blom et al., 2020; Wu et al., 2015), enabling the inferred character to be incrementally improved over time by integrating accumulated observation-action trajectories (Figure 1.**C**). In summary, the meta-agent observes the behavior of a target POMDP agent and infers the character distribution of the target using the proposed IIN (Figure 1.**D**).

Summary of Contributions:

- We introduce a stochastic agent model to train an agent with stochastic character in a POMDP problem (Section 2.1).
- We build a universal policy that is generalized across probable stochastic characters and provide an efficient training method without much sampling (Section 2.2 & 2.3).
- We propose an instant inference model that can inversely infer a stochastic character of others with high accuracy, even on short-term trajectories, and can gradually update the inferred character as the observing period increases (Section 3).
- To verify the effectiveness of the proposed method, we evaluate the performance of the inference model over the observation noise and trajectory length of a meta-agent (Section 4).

2. Universal Policy with Stochastic Character

In this section, we formulate the problems of a POMDP agent and a meta-agent to achieve two objectives: 1) building a POMDP agent with a stochastic character, and 2) building a meta-agent that can collect trajectories and infer a stochastic character of the POMDP agent. We formalize the first task as a POMDP and define a stochastic agent model with a stochastic distribution. Next, we train a uni-

versal policy that includes a set of policies based on diverse stochastic characters for building the meta-agent.

2.1. Partially Observable Markov Decision Process Agent with Stochastic Character

The POMDP is defined as a tuple $M = \langle \mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, \Omega, R, \gamma \rangle$ that comprises of a state $s_t \in \mathcal{S}$, an observation $o_t \in \mathcal{O}$, an action $a_t \in \mathcal{A}$, a state transition probability $\mathcal{T}(s_{t+1}|s_t, a_t)$, an observation transition probability $\Omega(o_t|s_t)$, a reward function R, and a temporal discounted factor γ. In detail, the reward r_t can be defined as a linear combination of n-th element of character vector c_n and reward component $\mathcal{R}_n(s_t, a_t, s_{t+1})$ as follows:

$$r_t = R(s_t, a_t, s_{t+1}; \mathbf{c}_\mu, \mathbf{c}_\sigma) = \sum_{n=1}^{N} c_n \mathcal{R}_n(s_t, a_t, s_{t+1}),$$

$$\text{where } \; c_n \sim f(c_n; c_{n,\mu}, c_{n,\sigma}, \mathrm{a}, \mathrm{b})$$

Since we consider the stochastic character, n-th element of character vector $c_n \in \mathbf{c} = [c_1, c_2, \cdots, c_N]^{\mathrm{T}} \in \mathcal{C}$ from character distribution $f(c_n; c_{n,\mu}, c_{n,\sigma}, \mathrm{a}, \mathrm{b})$ is sampled, where a mean of n-th element of character vector $c_{n,\mu} \in \mathbf{c}_\mu = [c_{1,\mu}, c_{2,\mu}, \cdots, c_{N,\mu}]^{\mathrm{T}}$, standard deviation of n-th element of character vector $c_{n,\sigma} \in \mathbf{c}_\sigma = [c_{1,\sigma}, c_{2,\sigma}, \cdots, c_{N,\sigma}]^{\mathrm{T}}$, and N denotes a size of character vector $|\mathbf{c}|$ (i.e., the number of components that make up character). Here, we define the character distribution as follows.

Definition 2.1 (Character distribution)**.** *A character distribution is defined as a truncated Gaussian distribution*

$$f(c_n; c_{n,\mu}, c_{n,\sigma}, \mathrm{a}, \mathrm{b}) = \frac{\zeta(\frac{c_n - c_{n,\mu}}{c_{n,\sigma}})}{c_{n,\sigma}(Z(\frac{\mathrm{b} - c_{n,\mu}}{c_{n,\sigma}}) - Z(\frac{\mathrm{a} - c_{n,\mu}}{c_{n,\sigma}}))},$$

where $\zeta(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$ *means the probability density function of the Gaussian distribution,* $Z(x) = \frac{1}{2}(1 + \mathrm{erf}(x/\sqrt{2}))$ *is its cumulative distribution function, and character variable* c_n *is constrained to be in the interval* $[\mathrm{a}, \mathrm{b}]$.

Herein, we consider the truncated Gaussian distribution as the character distribution to design a character defined in finite character space.[1]

The goal of the POMDP agent is to find a policy that can maximize the reward. The reward maximization strategy can be changed according to the combination of character parameters, and this difference makes diverse policies. Therefore, to find the policy that captures this diversity, we define a stochastic agent model as follows.

[1]Note that Gaussian distribution can be set to $a = -\infty$ and $b = +\infty$. It is worth noting that the proposed method still works with broader family of unimodal distributions (e.g.,Gamma and Poisson). The empirical results are shown in Figure 5.

Definition 2.2 (Stochastic agent model)**.** *A stochastic agent model is defined* $\mathrm{A} = (R, \pi, Q, f)$*, where* $r_t = R(s_t, a_t, s_{t+1}; \mathbf{c}_\mu, \mathbf{c}_\sigma)$ *means a stochastic character reward,* $\pi_\phi(a_t|o_t; \mathbf{c}_\mu, \mathbf{c}_\sigma)$ *and* $Q_\psi^{\pi_\phi}(o_t, a_t; \mathbf{c}_\mu, \mathbf{c}_\sigma)$ *denote a stochastic character policy and value function, and* $f(c_n; c_{n,\mu}, c_{n,\sigma}, \mathrm{a}, \mathrm{b})_{\forall n}$ *represents a character distribution.*

The POMDP agent interacts with the environment to learn an optimal policy $\pi_\phi^*(a_t|o_t; \mathbf{c}_\mu, \mathbf{c}_\sigma)$, where ϕ represents a policy network parameters. Specifically, the purpose of the agent is to maximize the expected cumulative reward $\mathbb{E}_{\pi_\phi}\Big[\sum_t \gamma^t R(s_t, a_t, s_{t+1}; \mathbf{c}_\mu, \mathbf{c}_\sigma)\Big]$. This objective function define the value function $Q_\psi^{\pi_\phi^*}(o_t, a_t; \mathbf{c}_\mu, \mathbf{c}_\sigma)$, where ψ denotes a Q-function network parameters. We contemplate the neural network for approximation because calculating and maximizing an Q-function about all combinations of observation and action requires a high cost in the continuous space.

2.2. Meta-agent in Partial Noisy Environment

In this subsection, we provide the role of the meta-agent as a *trajectory collector*. The meta-agent assumes the responsibility of observing the target POMDP agent's trajectory to infer their underlying character in partial noisy environment. To elaborate, the meta-agent observes noisy information $(\tilde{o}_t, \tilde{a}_t)$ on (s_t, a_t)—a state and an action of the POMDP agent—, where $\tilde{o}_t \sim \tilde{\Omega}_o(\tilde{o}_t|s_t) = \mathcal{N}(\Omega(o_t|s_t), \tilde{\sigma})$ represents noisy information on the state s_t, and $\tilde{a}_t \sim \tilde{\Omega}_a(\tilde{a}_t|a_t) = \mathcal{N}(a_t, \tilde{\sigma})$ represents noisy information on the action a_t. Here, $\tilde{\sigma}$ denotes the standard deviation of noise on meta-agent's observation.

2.3. Universal Policy

To endow the inference ability to meta-agent, we need to universal policy $\Pi_\phi(a_t|o_t; \mathbf{c}_\mu, \mathbf{c}_\sigma)$ that includes a set of policies $\{\pi_1(\mathbf{c}_\mu, \mathbf{c}_\sigma), \pi_2(\mathbf{c}_\mu, \mathbf{c}_\sigma), \cdots \pi_N(\mathbf{c}_\mu, \mathbf{c}_\sigma)\}$ over the character space $\mathcal{C}$. The universal policy can switch or mimic behavioral patterns over a given stochastic character. After the meta-agent collects the trajectories of the POMDP target agent, the universal policy contributes to uncovering the trajectories' character. We introduce how to build the universal policy in the following Subsection 2.4, and then we construct the inference network in Section 3.

2.4. Training Universal Policy

To build a universal policy with a stochastic character, we contemplate the following two considerations. One is to approximate the policy and Q-function in continuous observation and action spaces, and the other is to estimate the expected character of stochastic character for sample efficiency.

Algorithm 1 Train a Universal Policy

Require: Total episodes K, total timesteps of an episode T

Initialization: Network parameters ϕ, ψ
for episode $k = 1, K$ **do**
  Reset s_1 and get $o_1 \sim \Omega(o_1|s_1)$
  Sample character distribution parameters $\mathbf{c}_\mu, \mathbf{c}_\sigma$
  for timestep $t = 1, T$ **do**
    Execute $a_t \sim \Pi_\phi(a_t|o_t; \mathbf{c}_\mu, \mathbf{c}_\sigma)$
    Get s_{t+1}, r_t, and o_{t+1}
    Calculate the $Q_\psi^{\Pi_\phi}(\cdot)$
    Update ψ by back-propagating TD-error ϵ
    Update ϕ by back-propagating $-Q_\psi^{\Pi_\phi}$
    $t \leftarrow t + 1$
  end for
end for

First, we utilize the deep RL algorithm based on actor-critic networks as a function approximator to estimate the policy and value function in continuous space. The actor-network ϕ approximates the universal policy over the continuous observation and character parameters. The critic-network ψ approximates the value function over the continuous observation, character parameters, and action. The update of each network is based on a Temporal Difference error (TD-error) $\epsilon = R(s_t, a_t, s_{t+1}; \mathbf{c}_\mu, \mathbf{c}_\sigma) + \gamma Q_\psi^{\Pi_\phi}(o_{t+1}, a_{t+1}; \mathbf{c}_\mu, \mathbf{c}_\sigma) - Q_\psi^{\Pi_\phi}(o_t, a_t; \mathbf{c}_\mu, \mathbf{c}_\sigma)$ and negative value function $-Q_\psi^{\Pi_\phi}(\cdot)$ by using the gradient descent method. We provide the pseudocode in Algorithm 1.

Next, we consider an expected character value $\bar{\mathbf{c}} = [\bar{c}_1, \bar{c}_2, \cdots, \bar{c}_N]$ when computing reward $R(s_t, a_t, s_{t+1}; \mathbf{c}_\mu, \mathbf{c}_\sigma)$. This is because it needs various experiences to train the policy, which can capture the stochastic character distribution. Such a training task can be addressed the sampling-based method, which requires a lot of character sampling and estimation steps to approximate the character distribution. Besides, this approach requires many samples but cannot guarantee a reliable estimation. In other words, we consider the reward based on estimated character, which can be theoretically obtained when assuming the agent experiences enough, in order to address the problem.

$$
\begin{aligned}
&\mathbb{E}_{c_n \sim f(c_n), n \in \{1,2,\cdots,N\}} \left[R(s_t, a_t, s_{t+1}; \mathbf{c}_\mu, \mathbf{c}_\sigma) \right] \\
&= \mathbb{E}_{c_n \sim f(c_n), n \in \{1,2,\cdots,N\}} \left[\sum_{n=1}^{N} c_n \mathcal{R}_n(s_t, a_t, s_{t+1}) \right] \\
&= \sum_{n=1}^{N} \mathcal{R}_n(s_t, a_t, s_{t+1}) \mathbb{E}_{c_n \sim f(c_n)} \left[c_n \right] \\
&= \sum_{n=1}^{N} \mathcal{R}_n(s_t, a_t, s_{t+1}) \int c_n f(c_n; c_{n,\mu}, c_{n,\sigma}, \mathrm{a}, \mathrm{b}) dc_n \\
&= \sum_{n=1}^{N} \mathcal{R}_n(s_t, a_t, s_{t+1}) c_{n,\mu} \qquad (1)
\end{aligned}
$$

Therefore, the policy and Q-value networks do not need to experience many character samples to form the representation of a stochastic character.

Algorithm 2 IIN: Training Phase

Require: IIN $\Theta(\cdot)$, meta-agent's observation noise standard deviation $\tilde{\sigma}_{train}$, length of trajectory l_{train}, dataset $\bar{\mathcal{D}}$, and learning rate α

Ensure:
Initialize IIN parameters
while Not stop criterion **do**
  Sample trajectory data by the length of l_{train}
  $(o_t, a_t; \mathbf{c}_\mu, \mathbf{c}_\sigma) \sim \bar{\mathcal{D}}$
  Add noise on trajectory data
  $\tilde{o}_t \sim \mathcal{N}(\Omega(o_t|s_t), \tilde{\sigma}_{train})$ and $\tilde{a}_t \sim \mathcal{N}(a_t, \tilde{\sigma}_{train})$
  Pass the IIN
  $\hat{\mathbf{c}}_\mu = \frac{1}{l_{train}} \sum_{t=1}^{l_{train}} \hat{\mathbf{c}}_{\mu,t}$ and $\hat{\mathbf{c}}_\sigma = \frac{1}{l_{train}} \sum_{t=1}^{l_{train}} \hat{\mathbf{c}}_{\sigma,t}$
  Calculate the loss
  $\mathcal{L}(f(\mathbf{c}; \mathbf{c}_\mu, \mathbf{c}_\sigma, \mathrm{a}, \mathrm{b}))$
  $= H(f(\mathbf{c}; \mathbf{c}_\mu, \mathbf{c}_\sigma, \mathrm{a}, \mathrm{b}) | f(\mathbf{c}; \hat{\mathbf{c}}_\mu, \hat{\mathbf{c}}_\sigma, \mathrm{a}, \mathrm{b}))$
  Update the IIN parameters
  $\Theta \leftarrow \Theta - \alpha \Delta_\Theta \mathcal{L}(\cdot)$
end while

3. Instance Inference Network

In this section, we propose the IIN that can instantly infer the stochastic character of a POMDP agent. To achieve this functionality, we provide the objective function of the proposed inference model and train the proposed IIN by utilizing the meta-agent with the trained universal policy in Section 2.

3.1. Training of the IIN

We introduce the objective and configuration of the proposed IIN in this subsection. Subsequently, we provide the loss function to achieve the objective of the IIN.

The objective of the IIN is to inversely find the true character distribution $f(\mathbf{c}; \mathbf{c}_\mu, \mathbf{c}_\sigma, \mathrm{a}, \mathrm{b})$ of a POMDP agent by observing its trajectory. Accomplishing such an objective should require trajectory data over various characters. We cannot access the character distribution of the POMDP agents before the inference; therefore, we take advantage of the meta-agent. The meta-agent has the universal policy, i.e., the set of various policies, and can thereby generate differ-

ent trajectory data[2] over various characters for the training of the IIN.

Next, the IIN should be capable of instant inference and continuous updates on the character of others. To implement this functionality, we configure the IIN Θ as the many-to-many LSTM network (Hochreiter & Schmidhuber, 1997; Sak et al., 2014). Specifically, the input of the IIN is the observation-action pair (o_t, a_t),[3] and the output is the mean $\hat{\mathbf{c}}_\mu$ and standard deviation $\hat{\mathbf{c}}_\sigma$ of the stochastic character. Herein, the IIN uses the length of trajectory l_{train} to calculate $\hat{\mathbf{c}}_\mu$ and $\hat{\mathbf{c}}_\sigma$.[4] In other words, the proposed network considers the trajectory $(o_t, a_t)_{i+1:i+l_{train}}$ as long as trajectory length l_{train}. Algorithm 2 summarizes the training process of the IIN. We provide the illustrative structure of the IIN in Appendix I

3.2. Objective Function of the IIN

To train the IIN, we define an objective function of the IIN as the Kullback–Leibler (KL) divergence between the true $f(\mathbf{c}; \mathbf{c}^*_\mu, \mathbf{c}^*_\sigma, \mathrm{a}, \mathrm{b})$ and inferred character distribution $f(\mathbf{c}; \hat{\mathbf{c}}_\mu, \hat{\mathbf{c}}_\sigma, \mathrm{a}, \mathrm{b})$, where $\hat{\mathbf{c}}_\mu, \hat{\mathbf{c}}_\sigma$ represent the output of the proposed IIN $\Theta(o_t, a_t)$. Specifically, finding arguments $\hat{\mathbf{c}}_\mu, \hat{\mathbf{c}}_\sigma$ that minimize the KL-divergence between the true and inferred character distribution is identical to finding arguments that minimize the cross-entropy between the true and inferred one $H(f(\mathbf{c}; \mathbf{c}^*_\mu, \mathbf{c}^*_\sigma, \mathrm{a}, \mathrm{b}) | f(\mathbf{c}; \hat{\mathbf{c}}_\mu, \hat{\mathbf{c}}_\sigma, \mathrm{a}, \mathrm{b}))$. The *proof* for this property is provided in Appendix E.

Subsequently, suppose an agent has a deterministic character(i.e., when the true character distribution is the Dirac delta distribution). In that case, the objective function of the IIN can be simplified as **Theorem** 3.1, as a special case.

Theorem 3.1. *If the true character distribution is the Dirac delta distribution, i.e.,* $f(\mathbf{c}; \mathbf{c}^*_\mu, \mathbf{c}^*_\sigma, \mathrm{a}, \mathrm{b}) = \delta(\mathbf{c} - \mathbf{c}^*_\mu)$*, the inferred character distribution satisfies the Gaussian distribution, and* $\mathbf{c}_\sigma$ *is a constant, then the optimization problem* $\hat{\mathbf{c}}_\mu, \hat{\mathbf{c}}_\sigma = \arg\min_{\mathbf{c}_\mu, \mathbf{c}_\sigma} \mathcal{L}(f(\mathbf{c}; \mathbf{c}_\mu, \mathbf{c}_\sigma, \mathrm{a}, \mathrm{b}))$ *can be simplified as* $\hat{\mathbf{c}}_\mu = \arg\min_{\mathbf{c}_\mu} \frac{1}{N} \sum_{n=1}^{N} (c^*_{n,\mu} - c_{n,\mu})^2$.

We provide *proof* of theorem 3.1 in the Appendix F, and this theorem is on board with the following remark.

Remark 3.2. *If a POMDP agent with deterministic character is considered, the loss function of IIN (i.e., KL-divergence) can be replaced by mean squared error.*

[2]We provide the dataset-generating method in Appendix D.

[3]The meta-agent infers a character using the noisy information $(\tilde{o}_t, \tilde{a}_t)$ on a trajectory of a POMDP agent. Therefore, we consider the noisy information in the training phase for building a robust model. To distinguish notation, we use $\tilde{\sigma}_{train}$ and $\tilde{\sigma}_{test}$.

[4]We also consider the length of trajectory in the inference phase. Thus, we use l_{train} and l_{test} to distinguish notation.

Algorithm 3 IIN: Inference Phase

Require: Pre-trained IIN $\Theta(\cdot)$, meta-agent's observation noise standard deviation $\tilde{\sigma}_{test}$, length of trajectory l_{test}, and buffer $\mathcal{B}$

Ensure:

for timestep $t = 1, T$ **do**
  Collect the observation $\tilde{o}_t \sim \mathcal{N}(\Omega(o_t|s_t), \tilde{\sigma}_{test})$
  Collect the action $\tilde{a}_t \sim \mathcal{N}(a_t, \tilde{\sigma}_{test})$
  Infer the character distribution $\hat{\mathbf{c}}_{\mu,t}, \hat{\mathbf{c}}_{\sigma,t} = \Theta(\tilde{o}_t, \tilde{a}_t)$
  Store $(\hat{\mathbf{c}}_{\mu,t}, \hat{\mathbf{c}}_{\sigma,t})$ in $\mathcal{B}$
  Calculate the estimated character $\hat{\mathbf{c}}_\mu = \frac{1}{|\mathcal{B}|} \sum_{t=1}^{|\mathcal{B}|} \hat{\mathbf{c}}_{\mu,t}$
  $\hat{\mathbf{c}}_\sigma = \frac{1}{|\mathcal{B}|} \sum_{t=1}^{|\mathcal{B}|} \hat{\mathbf{c}}_{\sigma,t}$
end for

3.3. Instance Inference on Character Distribution

Once the IIN is fully trained across character space, the network can instantly infer a character of a POMDP agent. The meta-agent with pre-trained IIN observes the trajectory of the POMDP agent, infers a character, and updates the inferred character over a period. Specifically, the meta-agent estimates the character of the POMDP agent at every timestep. We provide the pseudocode for the operation of the IIN in the inference phase in Algorithm 3.

4. Experiments

To select an appropriate task for verification purposes, we consider multi-agent scenarios where agents with diverse heterogeneous characters can exist. We select the autonomous driving task where several vehicles, which have various driving patterns, are on the road. Herein, the diversity of driving character means the driving preferences of drivers (i.e., aggressive and defensive driving). This task is suitable for our setting because the drivers can observe partial state information and need to interact with others instantly (Cooper et al., 2002; Eboli et al., 2017; Rosbach et al., 2019).

Specifically, we consider a scenario where several vehicles are on the multi-lane roundabout road. Each agent decides acceleration and lane-changing as given their partial observation. To express the driving character of each agent, we parameterize the three terms of the reward function, i.e, the character vector is as follows, $\mathbf{c} = [c_1, c_2, c_3]$. We provide more details about experiments in Appendix G.

In the following subsections, we provide extensive simulation results to evaluate the performance of the proposed

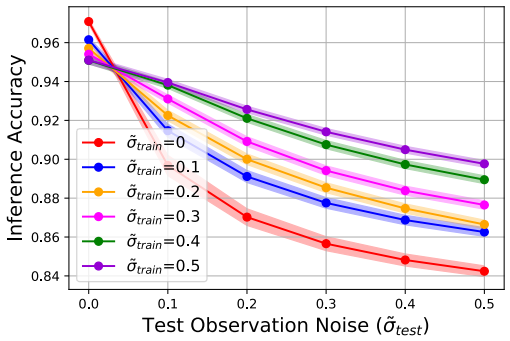

Figure 2. The inference accuracy of IIN according to the magnitude of observation noise in the training and test phases. In this simulation, the sequence length is set as $l = 100$.

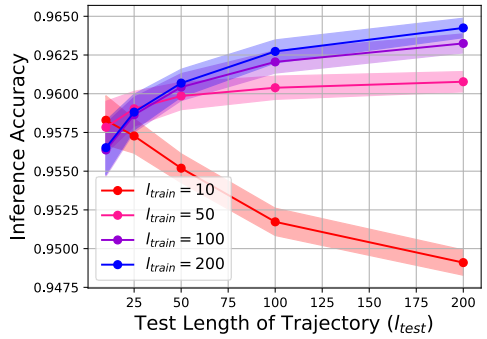

Figure 3. The inference accuracy of IIN over the length of trajectory in the training and test phases. In this simulation, the meta-agent's observation noise standard deviation is set to 0.

solution. We first analyze the performance of character inference as the sequence length and observation noise of a meta-agent increase. Next, we compare the proposed and other methods in terms of inference accuracy and time complexity. In all experimental results, we provide the average performance and confidence interval with three standard deviations over 10^3 inferences. The marker and dimmed area of the Figure represent the average performance and confidence interval, respectively. We define the inference accuracy as $1 - \frac{1}{N}\|\mathbf{c}_\mu - \hat{\mathbf{c}}_\mu\|_2^2$.

4.1. Performance Evaluation

In this subsection, we analyze the performance of the proposed solution as observation noise and sequence length increase.

Figure 2 shows the inference accuracy of the proposed solution as the observation noise considered in the inference phase increases. This figure confirms that the inference accuracy declines as increasing the standard deviation of the observation noise in the inference phase. Interestingly, the higher the standard deviation of the observation noise is, the more robust the proposed network is trained. We conclude that taking the observation noise in the training phase improves the inference accuracy when there is observation noise in the inference phase.

Next, Figure 3 depicts how the inference accuracy of the proposed model varies with the sequence length l_{test} of the test. When the sequence length in the training phase is more than 50, the inference accuracy increases as the sequence length in the inference phase grows. On the other hand, when the sequence length in the training phase is 10, the inference accuracy decreases as the sequence length in the test phase increases.

4.2. Performance Comparison

In this subsection, we compare the proposed solution to the other approaches in terms of inference accuracy and time complexity. We consider the following methods.

- Proposed (LSTM): The IIN includes the LSTM layers. It allows the inference process to consider time series property in observation-action trajectories.
- Proposed (Feedforward): The IIN includes only Feedforward layers. It uses a single data point for the inference, not the trajectory sequence.
- IRC (Kwon et al., 2020): Monte Carlo maximization estimation based inference method. It is to find the stochastic character distribution that best explains the given trajectories by maximizing their log-likelihood. This method includes iterations to perform the gradient ascent process.
- Variational Inference (VI) (Blei et al., 2017): Bayesian theorem based inference method. It is to approximate the posterior $p(\mathbf{c}|o, a)$ as a different probability $q(\mathbf{c})$ regardless of distribution type. The objective of this method is to minimize the KL divergence.

Table 1 shows the inference accuracy on the character and time complexity for four approaches. We exhibit three results of the LSTM-based proposed solution and the IRC for comparing the performance over different trajectory lengths ($l \in 10, 100, 200$). In contrast, the feedforward-based proposed one and the VI display a single result over each data point ($l = 1$), not the data sequence.

The simulation results confirm that the LSTM-based proposed solution with $l = 200$ performs the best. As expected, the inference accuracy of the LSTM-based proposed solution and the IRC gradually grows as the trajectory length increases. This is because a longer trajectory stands out as a

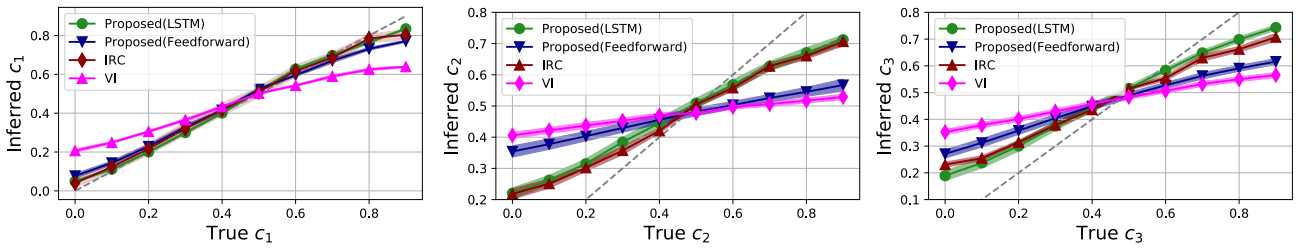

Figure 4. The inferred character parameters of the agent. The grey dotted line represents a diagonal line, where the estimated result is the same as the true one. Each marker depicts the estimated characters, and each dimmed area means the confidence interval for three standard deviations. Green, blue, and pink colors represent the proposed, MLP, and VI, respectively. **Left**: Inference on c_1, **Center**: Inference on c_2, **Right**: Inference on c_3.

Table 1. Inference accuracy (mean $\pm$ three standard deviations) and time complexity for four algorithms. In this table, l represents the train and test length of the trajectory (i.e., $l_{train} = l_{test}$), d denotes the dimension of the input feature, and k means the number of gradient ascent iterations in IRC. Note that run time of proposed (LSTM) is based on a sample (200 samples = $1.153s$).

Algorithm	Inference Accuracy			Time Complexity	Run-Time
	l=10	l=100	l=200		
Proposed (LSTM)	0.9564 $\pm$0.0017	0.9621 $\pm$0.0007	0.9633 $\pm$0.0006	$O(ld^2)$	0.006s
IRC	0.7874 $\pm$0.0883	0.9441 $\pm$0.0451	0.9604 $\pm$0.0344	$O(kld^2)$	91.472s
Proposed (Feedforward)	0.9533 $\pm$0.0019			$O(d^2)$	0.003s
VI	0.8900 + $\pm$0.0022			$O(d^2)$	0.004s

clearer difference in the behavioral pattern according to the character. Next, the feedforward-based proposed solution presents competitive performance with the LSTM-based one, although it does not utilize the time-series property. Thus, if a meta-agent cannot continuously observe the target, the feedforward-based proposed solution can be a good alternative. The IRC with $l = 200$ shows the second-best performance regarding mean accuracy; however, its reliance on the Monte Carlo method introduces high variance. Lastly, the VI exhibits the lowest performance, despite sharing similarities with the feedforward-based one.

Figure 4 confirms that the LSTM-based proposed solution's inferences are the closest to the ground truth among all algorithms. Trajectory-based approaches (e.g., the LSTM-based and the IRC) perform better than data point-based approaches (e.g., feedforward-based and the VI) for all c_1, c_2, c_3. Notably, the inference accuracy of c_1 is higher than c_2 and c_3. This is because the impact of each reward term on the action is different, and c_1 plays the dominant role to determine the action.[5]

Next, we analyze the computational complexity of all algorithms. In Table 1, k, l, and d represent the number of iterations, the trajectory length, and the dimension of features, respectively. All algorithms have quadratic time complexity about the dimension of features. Trajectory-based approaches require neural network operations as long as the trajectory length l. Additionally, the IRC performs k iterations for the gradient ascent process. The difference in time complexity between the proposed and IRC confirms that the proposed is faster than the IRC and explains why the IRC cannot instantly perform the inference, and the real run-time is consistent with the big O analysis. Details about time complexity analysis are provided in Appendix H.

In conclusion, the proposed solution comprehensively outperforms others in terms of inference accuracy and time complexity.

4.3. Performance on Exponential Family

In this subsection, we explore the possibility of IIN's operation for unimodal distributions belongings to the exponential family, other than the truncated Gaussian distribution. Specifically, we select the following distributions: the normal Gaussian, the Poisson, and the Gamma distribution. The character is defined as a parameter, which constitutes each distribution.

Figure 5 shows the inferred results of the LSTM-based proposed solution for each distribution. The proposed solution can still work for the unimodal distributions, being included in the exponential family, and infer decent performance. Interestingly, the trend of inference performance for each distribution is slightly different, e.g., the normal Gaussian

[5]We provide the intuition about behavioral differences according to the character in Appendix G.2

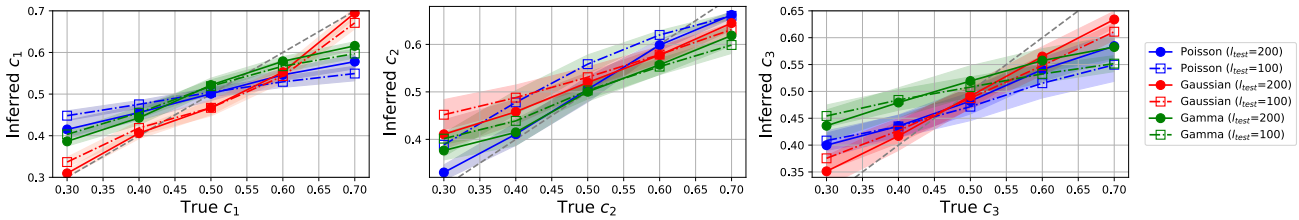

Figure 5. The inferred character parameters for each distribution, belong to the exponential family. A square with a dotted line and a circle with a solid line denotes the inferred performance of LSTM-based proposed with $l_{test} = 200$ and $l_{test} = 100$, respectively. Blue, red, and green mean the Poisson, the normal Gaussian, and the Gamma, respectively.

outperforms others for c_1 and c_3 while the Poisson outperforms for c_2. Consequently, this result demonstrates that the proposed to motivate employing a truncated Gaussian can apply to various exponential family distributions.

5. Related Works

Inverse Reinforcement Learning: The algorithm for inversely inferring the generative model of an agent's behavior can be broadly grouped into two folds: IRL (Ng et al., 2000; Ramachandran & Amir, 2007; Hantous et al., 2022) and IOC (Mombaur et al., 2010; Finn et al., 2016; Maroger et al., 2022). The IRL aims to learn the reward function, and IOC aims to learn the internal dynamics. In other words, the IRL assumes the known dynamics model to learn the reward/cost function based on observed information, and IOC assumes the reward function to infer the internal dynamics model of an agent. In (Herman et al., 2016), the authors propose the inverse solution, which can infer the reward function and dynamics model in the MDP case. In (Kwon et al., 2020) and (Wu et al., 2020), the authors propose the IRC that expands the solution in partially observable settings. As a task similar to IRC, there is a Bayesian Theory of Mind (BToM) (Baker et al., 2011; 2017; Lee et al., 2019). The commonality between IRC and BToM is in assuming an agent which is rational but has a possibility of error. However, there has been few research into inverse systems that can be run in real-time.

Machine Theory of Mind: As a cognitive theory, the Theory of Mind (ToM) refers to the ability to understand others by surmising what is happening in their mind. In various preliminary studies, the IRC (Kwon et al., 2020; Wu et al., 2020) and BToM (Baker et al., 2017; Lee et al., 2019) attempted to infer internal models. The IRC and BToM are motivated by the ToM, where the BToM focused on behavioral models and the IRC aimed to connect dynamic models to brain mechanics. Moreover, the MToM, a research topic for transferring these cognitive concepts to machines, is currently being studied (Rescorla, 2015; Rabinowitz et al., 2018; Nguyen & Gonzalez, 2020). In (Rabinowitz et al., 2018), the authors have opened the door to the possibility of the MToM research as a meta-learning strategy that explicitly learns an inference method. In (Shum et al., 2019), the authors provide the Composable Team Hierarchies (CTH), which is a generative model of multi-agent action comprehension based on a representation for these latent relationships. Moreover, in (Nguyen & Gonzalez, 2021), the authors improve the Instance-Based Learning Theory (IBLT) that observes others and makes the cognitive model and discuss the MToM's potential that could work in human-AI collaboration. However, many previous works require a high time cost to infer the internal model of others or are limited to the inference of a specific agent. To overcome these limitations, we proposed a real-time inference model that can infer all characters.

6. Discussion and Limitations

In this paper, we build a POMDP agent and meta-agent using the stochastic agent model in reinforcement learning, which captures the stochasticity of character. The meta-agent can collect observation-action pairs by observing a target POMDP agent. We train the IIN by leveraging universal policy and collected trajectory data; then, the meta-agent infers the target agent's character using pre-trained IIN. Consequently, we confirm that the proposed solution comprehensively outperforms the compared inference approaches.

To build an AI system that can effectively collaborate and interact with humans or AIs, it must be able to infer the characters of its human or AI partners quickly. These inferred characters should then be incorporated into the AI's decision-making process. We believe that the proposed framework can serve as a helpful guide for achieving these objectives.

The proposed whole framework, on the other hand, contains a few limitations needed for alleviation. We make two assumptions for experimental convenience: 1) the meta-agent obtains the other agent's observation through communication without cost, and 2) the probable character set must be predefined for the construction of universal policy and IIN. These assumptions can constrain practicality, and we will try to mitigate them further.

7. Acknowledge

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support programs (IITP-2022-2020-0-01602; No. 2021-0-00739, Development of Distributed/Cooperative AI based 5G+ Network Data Analytics Functions and Control Technology) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

Baker, C., Saxe, R., and Tenenbaum, J. Bayesian theory of mind: Modeling joint belief-desire attribution. In *Proceedings of the annual meeting of the cognitive science society*, 2011.

Baker, C. L., Jara-Ettinger, J., Saxe, R., and Tenenbaum, J. B. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1(4):1–10, 2017.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Blom, T., Feuerriegel, D., Johnson, P., Bode, S., and Hogendoorn, H. Predictions drive neural representations of visual events ahead of incoming sensory information. *Proceedings of the National Academy of Sciences*, 117(13): 7510–7515, 2020.

Bressloff, P. C., Ermentrout, B., Faugeras, O., and Thomas, P. J. Stochastic network models in neuroscience: A festschrift for jack cowan. introduction to the special issue, 2016.

Chan, A. J. and van der Schaar, M. Scalable bayesian inverse reinforcement learning. In *International Conference on Learning Representations*, 2020.

Chater, N., Zhu, J.-Q., Spicer, J., Sundh, J., León-Villagrá, P., and Sanborn, A. Probabilistic biases meet the bayesian brain. *Current Directions in Psychological Science*, 29 (5):506–512, 2020.

Cooper, R. A., Thorman, T., Cooper, R., Dvorznak, M. J., Fitzgerald, S. G., Ammer, W., Song-Feng, G., and Boninger, M. L. Driving characteristics of electric-powered wheelchair users: how far, fast, and often do people drive? *Archives of physical medicine and rehabilitation*, 83(2):250–255, 2002.

Craik, K. J. W. *The nature of explanation*, volume 445. CUP Archive, 1967.

Eboli, L., Mazzulla, G., and Pungillo, G. How drivers' characteristics can affect driving style. *Transportation research procedia*, 27:945–952, 2017.

Finn, C., Levine, S., and Abbeel, P. Guided cost learning: Deep inverse optimal control via policy optimization. In *International conference on machine learning*, pp. 49–58. PMLR, 2016.

Gonzalez, C. and Quesada, J. Learning in dynamic decision making: The recognition process. *Computational & Mathematical Organization Theory*, 9(4):287–304, 2003.

Griesche, S., Nicolay, E., Assmann, D., Dotzauer, M., and Käthner, D. Should my car drive as i do? what kind of driving style do drivers prefer for the design of automated driving functions. In *Braunschweiger Symposium*, volume 10, pp. 185–204, 2016.

Hantous, K., Rejeb, L., and Hellali, R. Detecting physiological needs using deep inverse reinforcement learning. *Applied Artificial Intelligence*, pp. 1–25, 2022.

Herman, M., Gindele, T., Wagner, J., Schmitt, F., and Burgard, W. Inverse reinforcement learning with simultaneous estimation of rewards and dynamics. In *Artificial Intelligence and Statistics*, pp. 102–110. PMLR, 2016.

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

Kwon, M., Daptardar, S., Schrater, P. R., and Pitkow, X. Inverse rational control with partially observable continuous nonlinear dynamics. *Advances in neural information processing systems*, 33:7898–7909, 2020.

Lee, J. J., Sha, F., and Breazeal, C. A bayesian theory of mind approach to nonverbal communication. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 487–496. IEEE, 2019.

Maroger, I., Stasse, O., and Watier, B. Inverse optimal control to model human trajectories during locomotion. *Computer Methods in Biomechanics and Biomedical Engineering*, 25(5):499–511, 2022.

McNamee, D. and Wolpert, D. M. Internal models in biological control. *Annual review of control, robotics, and autonomous systems*, 2:339, 2019.

Mombaur, K., Truong, A., and Laumond, J.-P. From human to humanoid locomotion—an inverse optimal control approach. *Autonomous robots*, 28(3):369–383, 2010.

Ng, A. Y., Russell, S. J., et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 2, 2000.

Nguyen, T. N. and Gonzalez, C. Cognitive machine theory of mind. Technical report, Carnegie Mellon University, 2020.

Nguyen, T. N. and Gonzalez, C. Theory of mind from observation in cognitive models and humans. *Topics in Cognitive Science*, 2021.

Park, S. Y., Moore, D. J., and Sirkin, D. What a driver wants: User preferences in semi-autonomous vehicle decision-making. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2020.

Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S. A., and Botvinick, M. Machine theory of mind. In *International conference on machine learning*, pp. 4218–4227. PMLR, 2018.

Ramachandran, D. and Amir, E. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pp. 2586–2591, 2007.

Rescorla, M. The computational theory of mind. 2015.

Rosbach, S., James, V., Großjohann, S., Homoceanu, S., and Roth, S. Driving with style: Inverse reinforcement learning in general-purpose planning for automated driving. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2658–2665. IEEE, 2019.

Sak, H., Senior, A. W., and Beaufays, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014.

Shum, M., Kleiman-Weiner, M., Littman, M. L., and Tenenbaum, J. B. Theory of minds: Understanding behavior in groups through inverse planning. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 6163–6170, 2019.

Wu, S.-W., Delgado, M. R., and Maloney, L. T. Motor decision-making. *Brain mapping: an encyclopedic reference*, 3:417–427, 2015.

Wu, Z., Kwon, M., Daptardar, S., Schrater, P., and Pitkow, X. Rational thoughts in neural codes. *Proceedings of the National Academy of Sciences*, 117(47):29311–29320, 2020.