IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 19, 2022, accepted May 12, 2022, date of publication May 18, 2022, date of current version May 26, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3176104

Explaining Intelligent Agent's Future Motion on Basis of Vocabulary Learning With Human Goal Inference

YOSUKE FUKUCHI[1], **MASAHIKO OSAWA**[2], **HIROSHI YAMAKAWA**[3,4,5], **AND MICHITA IMAI**[6], **(Member, IEEE)**

[1]National Institute of Informatics, Tokyo 101-8430, Japan
[2]College of Humanities and Sciences, Nihon University, Tokyo 156-8550, Japan
[3]Whole Brain Architecture Initiative, Tokyo 111-0051, Japan
[4]School of Engineering, The University of Tokyo, Tokyo 113-8654, Japan
[5]RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan
[6]Faculty of Science and Technology, Keio University, Kanagawa 223-8522, Japan

Corresponding author: Yosuke Fukuchi (fukuchi@ailab.ics.keio.ac.jp)

This work was supported in part by the Core Research for Evolutionary Science and Technology, Japan Science and Technology Agency (JST CREST), Japan, under Grant JPMJCR19A1; and in part by the Research Grant of Keio Leading-edge Laboratory of Science and Technology.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Ethics Committee of the Faculty of Science and Technology, Keio University, under Application No. 2021-114.

ABSTRACT Intelligent agents (IAs) that use machine learning for decision-making often lack the explainability about what they are going to do, which makes human-IA collaboration challenging. However, previous methods of explaining IA behavior require IA developers to predefine vocabulary that expresses motion, which is problematic as IA decision-making becomes complex. This paper proposes Manifestor, a method for explaining an IA's future motion with autonomous vocabulary learning. With Manifestor, an IA can learn vocabulary from a person's instructions about how the IA should act. A notable contribution of this paper is that we formalized the *communication gap* between a person and IA in the vocabulary-learning phase, that is, the IA's goal may be different from what the person wants the IA to achieve, and the IA needs to infer the latter to judge whether a motion matches that person's instruction. We evaluated Manifestor by investigating whether people can accurately predict an IA's future motion with explanations generated with Manifestor. We compared Manifestor's vocabulary with that from *optimal* acquired in a situation in which the communication-gap problem did not exist and that from *ablation*, which was learned with a false assumption that an IA and person shared a goal. The experimental results revealed that vocabulary learned with Manifestor improved people's prediction accuracy as much as with *optimal*, while *ablation* failed, suggesting that Manifestor can enable an IA to properly learn vocabulary from people's instructions even if a communication gap exists.

INDEX TERMS Explainable AI, human–agent interaction, intelligent agent, deep reinforcement learning.

I. INTRODUCTION

The development of machine-learning methods has allowed intelligent agents (IAs) to learn complex decision-making. Deep reinforcement learning (DRL) has broadened the applicability of IAs. However, while a growing number of studies are beginning to tackle problems when IAs mix with human society, there is still much room for improving the quality of interaction between humans and IAs.

This paper focuses on explaining what an IA is going to do. It is difficult for non-experts to understand an IA's complex decision-making process in a machine-learning module [1]; as a result, people become unable to predict the IA's behavior. Unpredictable behavior can cause unintended results or accidents. Moreover, for IAs to effectively work with people, both a person and IA should be able to understand each other's future behavior to decide roles to take in each situation [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Anandakumar Haldorai.

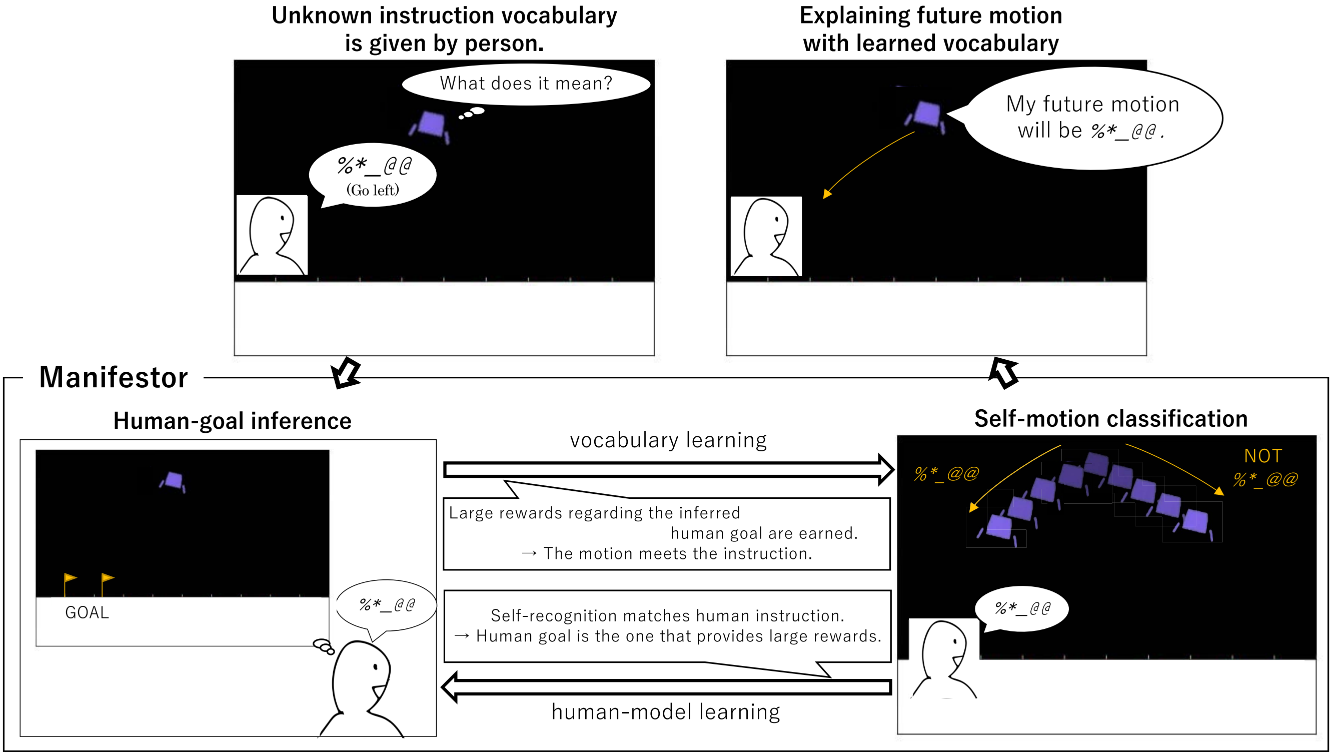

Unknown instruction vocabulary is given by person.
What does it mean?
%*_@@ (Go left)
Explaining future motion with learned vocabulary
My future motion will be %*_@@.
Manifestor
Human-goal inference
GOAL
%*_@@
vocabulary learning
Large rewards regarding the inferred human goal are earned. → The motion meets the instruction.
Self-recognition matches human instruction. → Human goal is the one that provides large rewards.
human-model learning
Self-motion classification
%*_@@
NOT %*_@@
%*_@@

FIGURE 1. Manifestor enables IA to learn vocabulary used in person's instructions and apply it to explain IA's future motion.

Methods were proposed for generating explanation of motions that an IA will show. Hayes *et al.* proposed a natural-language question-answering system that provides an explanation of what an IA does in a particular situation [1]. Waa *et al.* proposed a method for explaining not a one-shot action but a sequence of actions [3].

A problem with previous methods is that they require IA designers to predefine vocabulary that expresses an IA's behavior. Predefining vocabulary by hand is relatively easy in a simple environment such as grid world [4]. However, when an IA deals with robot motor control, for example, decision-making can be highly-frequent, high-dimensional, or sustaining time delay. We cannot simply correspond an IA action with a specific expression, which makes defining vocabulary much more complex.

We propose Manifestor, a method for explaining what an IA is going to do by autonomously enabling the IA to learn vocabulary that expresses its motions (Fig. 1). Manifestor enables an IA to learn vocabulary from instructions that people give to the IA. This setting is analogous to the instruction-following framework [5]–[7], in which an IA aims to learn a policy, or how to act, to follow a given instruction. Instruction following is typically formalized as an RL problem; that is, an IA earns more reward when its action fits more to the instruction.

As well as the difference in not at generating motions but explaining an IA's motions, a significant point of Manifestor lies in what we call the *communication gap*. In this paper, a communication gap refers to a problem in which the goal that a person wants an IA to achieve can be different from that of the IA, and the IA does not know which the person has. More specifically, an IA does not know which reward function is behind a person's instructions. Unlike instruction following in which an IA obtains reward feedback on whether its motion follows an instruction, an IA requires a meta-inference of the person's goal to learn the correspondences between its motion and vocabulary used in an instruction. Human-human interaction typically contains a communication gap because each person has her/his goals or intentions, and such mental states are more or less uncertain. A communication gap can also arise between an IA and person particularly if the person is not familiar with the design of the IA's decision-making.

Figure 1 illustrates our main idea. Manifestor solves the problem of vocabulary learning with communication gap as two inferences: (i) inference of a human goal allows an IA to learn vocabulary in a manner similar to instruction following. (ii) By comparing a human instruction with a classification result of an IA motion by the learned vocabulary, an IA can estimate which goal the human has, that is, when an IA recognizes that its motion matches a human instruction, a human goal is likely the one with which the motion earns more rewards. Manifestor enables an IA to learn vocabulary using two loss functions that represent each of the statements above.

This paper reports the results of experiments conducted to evaluate Manifestor. A numerical experiment focused on confirming the basis of Manifestor, and a user study experiment aimed at investigating whether explanation generated with Manifestor can improve the predictability of an IA's future motion. We compared Manifestor with two alternatives: *optimal* is trained in situations in which a person always gives instructions on the basis of an IA's true goal, so the IA does not need to consider the communication gap, corresponding to the instruction-following setting. *ablation* is trained with a false assumption that a communication gap did not exist. As a result, Manifestor showed similar performance as *optimal*, while *ablation* failed, suggesting that even if a communication gap exists, Manifestor enables an IA to correctly learn vocabulary and effectively explain its future behavior.

This paper is structured as follows. Section II describes the background of Manifestor from the perspective of both the explainable artificial intelligence (XAI) problem and vocabulary learning. We also formalize the communication-gap problem. Section III explains the design of Manifestor for learning vocabulary in situations with a communication gap. Section IV describes the details of our implementation for evaluating Manifestor. Section V reports the results of the numerical and user study experiments. Section VI discusses the limitations and future directions of Manifestor. Finally, Section VII concludes this paper.

II. BACKGROUND

A. GOAL-ORIENTED XAI

An XAI refers to an intelligent system that can explain its decision-making to people [8]. Some machine-learning models, particularly deep learning models, are composed of variables that people cannot easily interpret, which increases the need for XAI. Adadi and Berrada pointed out that XAI stems from at least four motivations: to justify AI decision-making, make AIs controllable, improve AIs, and enable people to discover insights from machine learning results [9].

The XAI problem is also critical for IAs that autonomously learn their policy with machine-learning methods. XAI for IAs is specifically called goal-oriented XAI [10] or explainable agency [11]. Goal-oriented XAI is necessary for people to control or improve an IA and avoid unintended behavior. It is also concerned with whether people can trust an IA [12], [13].

Puiutta & Veith applied the taxonomy of XAI [9] to goal-oriented XAI [14]. One factor is whether a method is intrinsic or post-hoc. Intrinsic methods are used for building a machine-learning model that is constitutionally interpretable. By using a decision-tree model or attention mechanism [15], it is easier for people to interpret an IA's decision-making process. Certain methods add constraints to a deep-learning-model structure so that decision-making models explicitly have human interpretable variables such as goals [16], [17]. Post-hoc methods, however, focus on generating explanations of incomprehensible models after training. Although post-hoc methods are not guaranteed to explain the literal decision-making process of an original machine-learning model, they do not affect model performance.

Another factor is whether an explanation is global or local. Global explanation provides a summary of an IA's general behavior [1], [18] whereas local explanation targets behavior in a specific situation. A major approach for local explanation is using a target model's saliency map, a visualization of input factors that strongly affected the model's decision-making [19]–[21]. Saliency maps provide the reason an IA took a specific action and can be a clue for people to predict IA behavior [20].

Manifestor provides post-hoc and local explanations. It focuses on generating explanation of an IA whose model has little interpretability for people. We consider a specific motion that an IA is going to show so that people can correctly predict the future.

B. EXPLAINING IA FUTURE ACTION

Most XAI studies focus on explaining *why* a decision is made, and little consideration has been taken for explaining *what* the decision will be. However, because an IA's action can cause unrecoverable effects, including physical changes in the environment, it is important to be able to explain its action before taking it. Explaining what an IA is going to do is also essential for cooperation with humans, because effective cooperation is based on mutual understanding of what others will do [2].

Hayes *et al.* proposed a question-answering system for explaining an IA's behavior [1]. It can answer a question of what an IA will do under specific circumstances. Strictly speaking, this is a global explanation based on a Markov decision-process model. Waa *et al.* proposed a method for explaining not a one-shot action but a sequence of actions [3].

However, they focused on an IA in a grid world, and challenges remain for applying it to another domain. A major challenge is that an IA's action is assumed to be easily associated with a symbolic expression for explaining to people. It becomes difficult to define vocabulary since an IA's decision-making is complex, making autonomous learning of vocabulary more promising.

C. ENABLING IA TO LEARN VOCABULARY

A simple machine-learning approach for grounding vocabulary with motion is supervised learning using a dataset of motion-caption pairs. Methods have been proposed to classify human activity using RGB (red, green, blue) cameras or depth sensors into caption labels [22]–[24], and a study focused on robot-motion captioning [25].

Typical trials for an IA to interactively learn vocabulary from people are in the instruction-following framework [5]–[7], [26], in which an IA seeks a policy for instructions given by people. Particularly in a reward-based approach [27], [28], an IA learns policy $\pi_{\text{instruction}}$ that maximizes expected reward given the environment state s_t and instruction u_t with RL methods:

$$a \sim \pi_{\text{instruction}}(a_t|s_t, u_t) \propto E_{\pi_{\text{instruction}}}[\sum_{0\leq\tau} \gamma^{\tau} r_{t+\tau}],$$

where r_t is a reward given at time t, and γ is a discount rate for future rewards. On the boundary between XAI and instruction following, Shu *et al.* proposed a hierarchical RL model that improves interpretability of the learned behavior by associating sub-policies with vocabulary used in an instruction [17].

Instructions from people can be also used to boost an IA's action learning in which an IA can solely achieve its goal without instructions. Interactive RL (IRL) aims at enabling an IA to quickly learn its policy from people's symbolic feedback [29]. Therefore, we consider a situation in which a person mentions how an IA should act.

In reward-based instruction following, it is assumed that an IA and person giving instructions share goal $g \in \mathcal{G}$. That is, an IA's goal g_{agent} is the same as a person's goal g_{human}. Here, g_{agent} is a variable that specifies the reward r for an IA's action in a specific environment status:

$$r = R(s, a, g_{agent}). \tag{1}$$

We call R a reward function and g_{human} a variable that is behind a person's instruction:

$$u = H(s, g_{human}). \tag{2}$$

In instruction following, larger rewards are given to an IA when its action more matches a person's instructions. However, when a non-expert person attempts to communicate with an autonomous IA, s/he can mention something other than g_{agent}, or $g_{agent} \neq g_{human}$, because s/he may not know exactly which goal the IA has or want the IA to work on another task. This paper focuses on such a *communication gap* to correctly interpret instructions given by people.

Manifestor is an extension of our previously proposed prototype method called Instruction-based Behavior Explanation (IBE) [30], [31]. IBE also uses vocabulary used in instructions from a person for explaining an IA's future motion and empirically demonstrated that its explanation improves the predictability of IA behavior. The largest significance of Manifestor over IBE is that it handles the communication gap between a person and IA, whereas IBE does not. Moreover, Manifestor quantitatively formalizes vocabulary learning with two loss functions while IBE relies on manual design of thresholds for determining whether an IA behavior matches a given instruction.

This paper tackles an extreme case in which a person only provides instructions and does not provide any other feedback such as whether the motion matches an instruction. This is not realistic for actual application because feedback boosts the learning process, but we chose this case to explore the possibility of vocabulary acquisition with as little additional information from a person as possible.

D. LunarLander-v2 AND INSTRUCTIONS

We used LunarLander-v2 provided in Open AI Gym [32] as a task for which an IA acts and in which a person gives instructions. An IA aims to land a rocket on an objective landing spot by manipulating main and side thrusters located on the rocket's bottom and left and right sides, respectively. A possible action $a \in \mathcal{A}$ is choosing which thruster to ignite to accelerate or decelerate the rocket. An IA can choose no thruster as well, with which the rocket moves in accordance with gravity and inertia. The environment state $s \in \mathcal{S}$ represents the rocket's location, velocity, degree of tilt, with which an IA choose its action. Rewards are calculated on the basis of the distance to a landing spot, deceleration, and decrease in tilt for each time step. One-shot positive/negative reward is given when the rocket succeeds/fails in landing.

In this paper, goals correspond to the location of a landing spot. We modified LunarLander-v2 so that we could change the landing-spot location for each trial, whereas the original has a fixed landing point at the center of the ground.[1]

Compared with a grid world, an action of LunarLander-v2 is not intuitive for people [33], so it is difficult to correspond an IA action with human vocabulary. One reason is that an action causes different results depending on the s. For example, the effect of rocket ignition on its velocity depends on the rocket's angle. In addition, a rocket behavior that people can recognize is not based on a single action but a sequence of actions because an action is chosen at high frequency (20 ms), which sustains time delay.

Following previous studies [30], [31], we defined a simple rule of generating an instruction for an IA:

$$H(s, g) = \begin{cases} \text{Go left. (if } s.x > g.x_{right}) \\ \text{Go right. (if } s.x < g.x_{left}) \\ \text{Fall straight down. (else),} \end{cases} \tag{3}$$

where $s.x$ is the location of the rocket on the horizontal axis, and $g.x_{left}$ and $g.x_{right}$ represent the left and right end of the landing spot g, respectively. An instruction is based only on the locations of the rocket and human goal, and inertia of the rocket is not taken into account. A person is assumed to consistently give instructions with g_{human}, which is fixed in an episode. In LunarLander-v2, an episode is from a rocket beginning to land to completing the landing.

III. MANIFESTOR

A. OVERVIEW

Manifestor is a method for explaining an IA's future motion by learning vocabulary from people. It generates future-motion explanation by predicting the transition of environment states caused by an IA and translating it to human vocabulary. Manifestor interprets a person's instruction about how s/he wants an IA to act while inferring her/his

[1] The implementation is available online (https://github.com/fuku5/multi_lunar_lander)

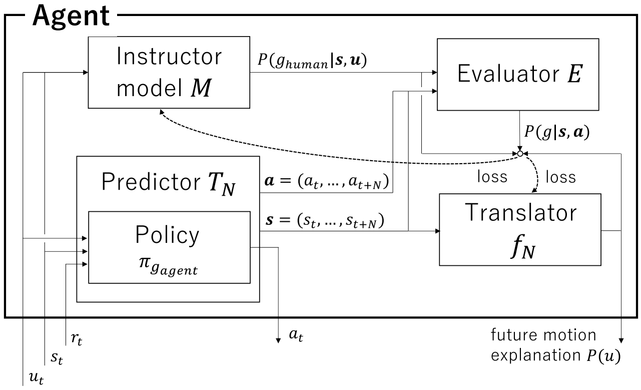

Agent
Instructor model M
$P(g_{human}|\boldsymbol{s},\boldsymbol{u})$
Evaluator E
$P(g|\boldsymbol{s},\boldsymbol{a})$
Predictor T_N
$\boldsymbol{a} = (a_t, \ldots, a_{t+N})$
loss
loss
Translator f_N
Policy $\pi_{g_{agent}}$
$\boldsymbol{s} = (s_t, \ldots, s_{t+N})$
r_t
s_t
u_t
a_t
future motion explanation $P(u)$

FIGURE 2. Components of Manifestor.

goal and grounds vocabulary used in instructions. Therefore, it becomes unnecessary for IA designers to manually define vocabulary for explanation.

B. MODULES

Manifestor is composed of five modules: policy π_g, predictor T_N, translator f_N, instructor model M, and evaluator E (Fig. 2). π_g determines an action a on the basis of s under g in the same manner with typical RL methods. The T_N predicts a sequence of actions ($\boldsymbol{a}_{t,N} = (a_t, a_{t+1}, \ldots, a_{t+N})$) and transitions in environment state ($\boldsymbol{s}_{t,N} = (s_t, s_{t+1}, \ldots, s_{t+N})$) in N steps on the basis of π_g.

$$T_N(\pi_g, s_t) = (\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N}).$$

The f_N outputs a probability distribution that represents the correspondence between transition $\boldsymbol{s}_{t,N}$ and vocabulary u:

$$u \sim f_N(u|\boldsymbol{s}_{t,N})$$

The M infers a person's goal g_{human} from her/his instructions:

$$g_{human} \sim M(g_{human}|\boldsymbol{s}_{0,\tau}, \boldsymbol{u}_{0,\tau}),$$

where τ is the length of an episode, and $\boldsymbol{u}_{0,\tau} = (u_0, u_1, \ldots, u_\tau)$ is a sequence of instructions in an episode. This formalization is based on the assumption that g_{human} is fixed in an episode. Finally, the E represents to what extent a transition and sequence of actions caused by the IA policy ($\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N}$) fits a g:

$$g \sim E(g|\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N})$$

We focus on the training of the f_N and M but do not discuss the problems with T_N or E such as how to train it and prediction accuracy.

C. LOSS FUNCTIONS

We define loss functions for training the f_N and M. These loss functions represent the two ideas shown in Fig. 1. That is, we can correspond instruction vocabulary and a motion in a similar manner if the goal behind the instruction is given, and the goal behind an instruction can be inferred on the basis of how much a self-classification result of an IA motion matches the instruction. These ideas are used for training f_N and M, respectively.

1) TRANSLATOR f_N

Let us first consider a situation in which a person and IA share the goal, that is, $g_{agent} = g_{human}$. A person provides instruction u_t on the basis of s_t and g_{human} (cf. Eq. 2). An IA chooses actions afterwards for N steps, and we obtain a sequence of actions $\boldsymbol{a}_{t,N}$ and an environment transition $\boldsymbol{s}_{t,N}$. We define a loss function for the f_N, i.e., L_{f_N}:

$$L_{f_N} = -E(g_{human}|\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N}) \cdot \log f_N(u_t|\boldsymbol{s}_{t,N}). \tag{4}$$

The f_N is trained to minimize L_{f_N}. With this loss function, $\boldsymbol{s}_{t,N}$ more strongly corresponds to u_t the more $(\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N})$ accords with g_{human}. The E is based on rewards $R(s, a, g)$ that will be given when assuming each possible goal.

$$E(g|\boldsymbol{s}, \boldsymbol{a}) = \underset{g\in\mathcal{G}}{\mathrm{softmax}}\Big(\sum_{s,a\in\boldsymbol{s},\boldsymbol{a}} R(s, a, g)\Big).$$

Next, let us suppose a situation with a communication gap, or $g_{agent} \neq g_{human}$. We extended the former loss function as follows:

$$L^+_{f_N} = -\sum_{g\in\mathcal{G}} (M(g|\boldsymbol{s}_{0,\tau}, \boldsymbol{u}_{0,\tau}) \cdot E(g|\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N})) \cdot \log f_N(u_t|\boldsymbol{s}_{t,N}), \tag{5}$$

where $L^+_{f_N}$ depends on a person's goal inferred by M. The $\sum(M \cdot E)$ in Equation 5 corresponds to $E(g_{human}|\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N})$ in Equation 4. It takes into account all $g \in \mathcal{G}$ as a possible human-goal candidate because g_{human} is hidden from the IA. The sum of $M \cdot E$ is the expected value of $E(g_{human}|\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N})$ when we consider g_{human} as a random variable.

2) INSTRUCTOR MODEL M

Equation 6 shows the loss function for training the M, i.e., L_M:

$$L_M = -\frac{1}{\beta} f_N(u_t|\boldsymbol{s}_{t,N}) \cdot \sum_{g\in\mathcal{G}} (E(g|\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N}) \cdot \log M(g|\boldsymbol{s}_{0,\tau}, \boldsymbol{u}_{0,\tau})), \tag{6}$$

which expresses our idea that (a) when self-recognition of an agent's motion matches an instruction, (b) a human goal should be the one with which the motion earns large rewards. The terms f_N and $\sum(E \cdot \log M)$ represent (a) and (b), respectively. We focused on the co-occurrence of (a) and (b). That is, $\sum(E \cdot \log M)$ should be maximized (or $-\sum(E \cdot \log M)$ should be minimized) when f_N is large.

Equation 6 expresses the co-occurrence but has a loophole. It can also be minimized by decreasing the two terms individually. Therefore, we added β for a penalty factor to avoid this loophole and focus on the co-occurrence:

$$\beta = E_t[f_N(u_t|\boldsymbol{s}_{t,N})] \cdot E_t\Big[\sum_{g\in\mathcal{G}} (E(g|\boldsymbol{s}_{t,N}, \boldsymbol{a}_{t,N}) \cdot \log M(g|\boldsymbol{s}_{0,\tau}, \boldsymbol{u}_{0,\tau}))\Big].$$

We do not consider $\beta = 0$ because it cannot not theoretically occur when we use the softmax function for the outputs of f_N, E, or M.

IV. IMPLEMENTATION

A. TRAINING PROCEDURE

The f_N and M require each other's inference for training (Eqs. 5 and 6). Namely, their training is interdependent, and we could not stabilize the learning results when simultaneously training the two in our trials. To focus on validating our formalization of $L^{+}_{f_N}$ and L_M, we simplified the learning process by introducing assumptions and splitting the training process into three phases.

In the first phase, instructions are divided into n groups with an unsupervised learning method regardless of L_M. Specifically, we used the encoder-decoder model:

$$\hat{g} \sim Encoder(\hat{g}|\boldsymbol{s}_{0,\tau}, \boldsymbol{u}_{0,\tau}), \tag{7}$$

$$u_t \sim Decoder(u_t|s_t, Encoder(\boldsymbol{s}_{0,\tau}, \boldsymbol{u}_{0,\tau})), \tag{8}$$

where $\hat{g} \in \hat{\mathcal{G}}$ is the result of the unsupervised learning method.

The second phase is training the f_N on the basis of $\hat{g}$. The unsupervised learning method does not provide the relationship between $\hat{g} \in \hat{\mathcal{G}}$ and $g \in \mathcal{G}$, so there can be multiple combinations. Let us consider a mapping $m : \hat{\mathcal{G}} \rightarrow \mathcal{G}$. Therefore, we can build the M.

$$M_m(g|\boldsymbol{s}_{0,\tau}, \boldsymbol{u}_{0,\tau}) = \sum_{\hat{g} \in \hat{\mathcal{G}}} \delta(g, m(\hat{g})) \cdot Encoder(\hat{g}|\boldsymbol{s}_{0,\tau}, \boldsymbol{u}_{0,\tau}), \tag{9}$$

where $\delta(a, b)$ is 1 if $a = b$ and 0 otherwise. When we assume $|\mathcal{G}| = |\hat{\mathcal{G}}| = 3$ and that there is a one-to-one correspondence between $\hat{g}$ and g, we can consider $3! = 6$ mappings. In this phase, we trained the f_N using Eq. 5 for all possible Ms with each mapping.

In the third phase, we evaluate all the Ms with Eq. 6. The final training result is from the f_N trained with the best mapping, which minimizes Eq. 6. The training of the M is simplified as a problem of choosing the best mapping m.

B. MODELS

In this paper, the f_N is a Transformer-Encoder model [34] to handle time series data. The model can be trained with a gradient method on the basis of Eq. 5. We inserted a [CLS] token [35] at the beginning of the input and transformed the output as a probability distribution of u with multi-layer perceptron and the softmax function.

The *Encoder* of the M (Eq. 7) is implemented with a model similar to the f_N, but it receives a sequence of both s and u. The output of the *Encoder* expresses a probability distribution of $\hat{g}$ behind instructions. The decoder (Eqn. 8) is a multilayer perceptron.

V. EVALUATION

A. OVERVIEW

Two experiments were conducted to evaluate Manifestor, i.e., a numerical experiment for confirming the basis of Manifestor, and a user study experiment investigating whether explanation generated with Manifestor can contribute to improving the predictability of an IA's future motion for people.

We compared Manifestor with *optimal* and *ablation* to investigate its performance against a communication gap. *optimal* is trained with instructions on the basis of the g_{agent} and L_{f_N}. It does not need to take into account a communication gap, thus should provide optimal results. Manifestor is trained with instructions on the basis of the g_{human}, the same as with *ablation*, which falsely ignores a communication gap using L_{f_N}. *optimal* and *ablation* simulate the results made by the contemporary instruction-following methods [27], [28]. They demonstrate what occurs when we introduce current methods in situations with or without a communication gap.

B. NUMERICAL EXPERIMENT

1) AIMS

The numerical experiment was conducted to validate Manifestor by investigating the following two questions:

i) Can we choose the best mapping m^* for the M with L_M?

ii) Do the training results acquired on the basis of $L^{+}_{f_N}$ in a situation with a communication gap match the *optimal* vocabulary acquired with L_{f_N} in a situation without a communication gap?

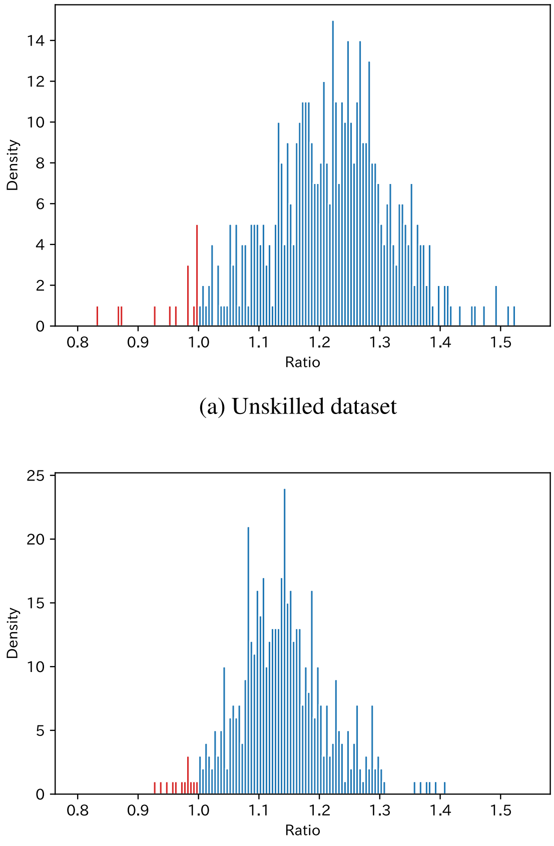
These questions are for validating our idea expressed with L_M and L_{f_N}, respectively.

2) PROCEDURE

An IA policy was trained with Advantage Actor-Critic (A2C), one of the most representative algorithms for DRL. From this policy, we created datasets for training and evaluating Manifestor. The datasets were composed of tuples of an s, an instruction based on a g_{human}, and an instruction based on a g_{agent}. A g_{human} is randomly chosen for each episode. We prepared two datasets, unskilled and skilled datasets, on the basis of a policy trained for 500,000 and 150 million time steps, respectively, to supplementally investigate the effects of IA-policy performance on vocabulary learning.

We set $N = 100$ (five seconds), which we determined for the following user study experiment considering the balance between the difficulty of predicting where a rocket lands and the time left for letting people understand the context for prediction on the basis of our pilot experiment. A dataset has 3,200 episodes, and we used half for training and the other half for evaluation.

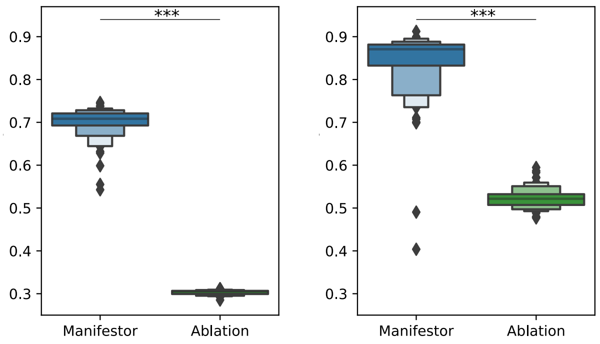
For question i), we executed the procedure shown in Subsection IV-A with 100 different random seeds. Training with each random seed produces six Ms and f_Ns, and there is the mapping m^* with which the corresponding M most accurately predicts the ground truth of g_{human}. We calculated the ratio of



(a) Unskilled dataset

(b) Skilled dataset

FIGURE 3. **Histograms of ratio distributions. Blue lines show density of ratio greater than 1, and red lines elsewise.**



(a) Unskilled dataset

(b) Skilled dataset

FIGURE 4. **Accuracy of Manifestor and *ablation*. ($***: p < .001$). See Appendix B for statistical details.**

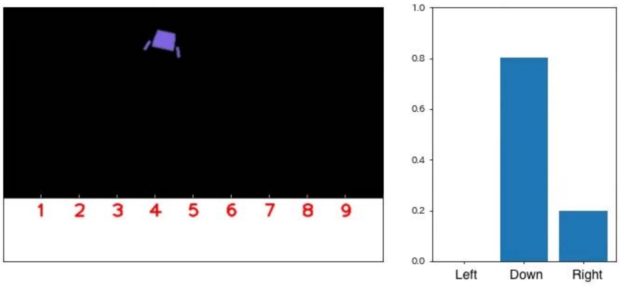


FIGURE 5. **Interface for user study experiment.**

L_{M_m} ($m \neq m^*$) to $L_{M_{m^*}}$ as a measure of the loss function. A ratio of more than 1 means that the loss function can select the best mapping.

For question ii), we calculated the accuracy of Manifestor, where accuracy means how much the outputs of Manifestor's f_N match those of *optimal*. We used the f_N with the ground truth mapping to focus on $L^+_{f_N}$ and remove the effects of L_M. For comparison, we calculated the accuracy of *ablation*.

3) RESULTS

Figure 3 shows the results for question i). Similar results, except for the breadth of the distributions, were obtained with both unskilled and skilled datasets. From 500 samples, we could successfully distinguish the best mapping in 485 and 487 samples (97.0 and 97.4 %). The mean ratios were 1.22 (95% CI[2] 1.01, 1.42) and 1.14 (95% CI 0.99, 1.29), respectively.

Figure 4 illustrates the accuracy of Manifestor and *ablation*. With both datasets, Manifestor was significantly more accurate than *ablation* (Mann-Whitney's U test). The median accuracy values of Manifestor were .700 and .870, whereas those of *ablation* were .303 and .522 with the unskilled and skilled datasets, respectively. A possible reason is that the unskilled dataset has very few successful and many relatively better examples for human instructions, so it was difficult to clearly ground vocabulary to motions.

The numerical experiment results supported both questions, so we conclude that the two loss functions for Manifestor can effectively handle the communication-gap problem.

C. USER STUDY EXPERIMENT

1) AIMS

The user study experiment was conducted to evaluate Manifestor in more practical situations. We investigated whether future-motion explanation generated with Manifestor can improve the predictability of IA behavior for people.

2) PROCEDURE

Participants were asked to predict where a rocket would land. Figure 5 illustrates the interface shown to the participants. It showed the rocket's movement until five seconds (100 frames) before it landed along with explanation of its future motion generated with Manifestor or the other methods which we used in subsection V-B (*optimal* and *ablation*). We also prepared a baseline condition in which only the rocket

[2] Confidence interval.

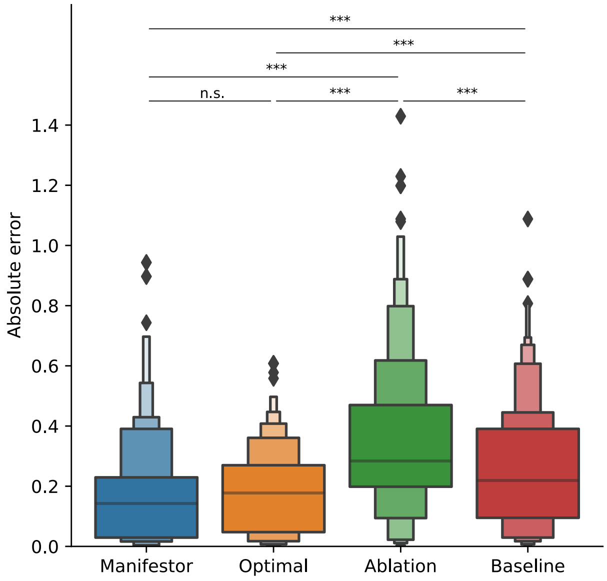


FIGURE 6. Absolute errors of participants' predictions. See Appendix B for statistical details.

movement and no explanation was provided. The explanation was shown as a bar graph.

We recruited 100 participants (26 female and 74 male; aged 21-67, $M = 41.3$, $SD = 8.6$) with compensation of 120 JPY from Lancers,[3] a crowdsourcing platform in Japan. The experiment was conducted on a web site. The participants were first provided pertinent information, and all the participants consented to the participation. Before the main task, we asked four simple questions to test the participants' comprehension about the task and removed 14 participants for evaluation. The experiment was conducted in a between-participant design, and 18, 20, 22, 26 participants were assigned to Manifestor, *optimal*, *ablation*, and baseline condition, respectively. In the main task, the participants were requested to answer where the rocket landed by indicating the index written on the moon ground (Fig. 5). Twenty episodes were shown in random order.

3) HYPOTHESES

We made two hypotheses to validate whether Manifestor can effectively explain an IA's future motion by managing the communication-gap problem:

(H1) Manifestor improves predictability as much as *optimal*.

(H2) *Ablation* does not improve predictability.

4) RESULTS

Figure 6 illustrates the absolute errors of the participants' predictions with statistical results. One tick error in Fig. 5 equals 0.2. The Kruskal-Wallis test revealed significant differences among the three methods and the baseline condition ($p < .001$). For a post-hoc analysis, we applied the Mann-Whitney's U test with Bonferroni correction to the results. We found significant differences among all combinations except for that between Manifestor and *optimal*.

Both Manifestor and *optimal* reduced the error compared with the baseline condition. The mean absolute errors were 0.172, 0.176, and 0.260 for Manifestor, *optimal*, and baseline condition, respectively. The effect size r was .25 between Manifestor and baseline condition and .22 between *optimal* and baseline condition. We found no significant difference between Manifestor and *optimal*. The r between the two was .06. These results support H1, meaning that even though a communication gap exists, Manifestor can enable an IA to learn vocabulary and generate future-motion explanations that improve the predictability of an IA motion as much as vocabulary learned in ideal situations in which a communication gap does not exist.

ablation failed to improve predictability of an IA motion and rather misled participants. The mean absolute error of *ablation* was 0.346. The r was .16 between baseline condition and *ablation*, and .37 between Manifestor and *ablation*. These results support H2, which confirms that the communication-gap problem needs to be solved in our settings to properly learn vocabulary from people.

VI. LIMITATIONS AND FUTURE WORK

We empirically demonstrated that Manifestor can enable an IA to properly learn vocabulary in situations with communication gaps and contribute to improving the predictability of IA motion with the learned vocabulary. However, the implementation of Manifestor and experimental settings mainly focused on validating our idea formalized as loss functions (Eq. 4-6), and further consideration is required to apply them to actual human-IA interaction.

We defined a rule for generating human instructions (Eq. 3), but a previous study on IRL revealed that human feedback signals are infrequent, inconsistent, or suboptimal [36]. It would be promising to improve Manifestor on the basis of IRL models that can handle such characteristics of human instructors.

As we mentioned in Subsection II-C, we assumed that a person only gives instructions and never provides feedbacks such as whether an IA's motion followed what s/he said. However, using feedback from people and Manifestor are complementary; feedback gives a boost to acquiring an M of Manifestor while an M reduces the need of feedback. For future work, we are planning to integrate other information provided by an instructor to both accelerate the training process of Manifestor and generate IA motions.

Contextual information is also helpful for developing an M. A g_{human} is randomly chosen for each episode in this paper, but the g_{human} can depend on context, and if so, context can be a hint to infer it. In particular, the behavior of a person who gives instructions provides plenty of information

[3] https://www.lancers.jp/

about her/his goal. An IA motion can also affect the g_{human} because a person sometimes tries to infer an artificial agent's mental states on the basis of mere observations of IA behavior [37], [38]. When a person tries to infer a g_{agent} to provide instructions, an IA may need to infer how its own motion is considered by people.

Another direction for improving Manifestor is to refer to semantics to interpret human instructions. Manifestor learns vocabulary without prior knowledge about what it means. However, an IA can more efficiently and properly interpret vocabulary by using a lexicon or language model trained with corpus data [39], [40]. Combining such information with Manifestor is promising for reducing the human cost of interacting with an IA for vocabulary learning.

Our implementation of Manifestor has an assumption of $g_{human} \in \mathcal{G}$, that is, the g_{human} is derived from a set of an IA's possible goals; thus, the IA's evaluator can evaluate whether a motion matches the g_{human}. However, a non-expert may ask an IA to work on a task that is overlooked in design or beyond the capabilities of the IA. Applying inverse RL methods [41] to a person may be a promising means to specify the $g_{human} \neq \mathcal{G}$ and build an evaluator that evaluate an IA motion on the basis of the inferred reward function.

Manifestor relies on the predictor, but predicting the transition of the environment is still an important domain of research. It is challenging particularly for the real world because it tends to be nondeterministic and highly complex. An actively researched domain is video prediction [42]–[44]. However, an IA needs to handle action-conditional prediction because its actions affect the environment [45]. Model-based RL that attempts to integrate a world dynamics model into an IA's decision-making can provide a direction for implementing an action-conditional prediction model for a more complex environment [46]. Prediction accuracy depends on N, or the length of prediction. We need to further investigate how long Manifestor structurally affords to generate future-motion explanation.

VII. CONCLUSION

We proposed Manifestor, a method for explaining an IA's future motion on the basis of vocabulary learning. Manifestor enables IAs to learn vocabulary that expresses their motions from a person's instructions of how they should act. By inferring their goals behind instructions, Manifestor can manage the communication-gap problem in which a person and an IA do not share their goals. The numerical and user study experiments demonstrated that Manifestor can generate future-motion explanation of an IA with learned vocabulary and improve the predictability of IA behavior even if communication gaps exist.

CODE AVAILABILITY

Our implementation is available online (https://github.com/fuku5/Manifestor).

APPENDIX A
EXPERIMENTAL SETUP DETAILS

A. SOFTWARES

Table 1 lists the softwares used in the experiments.

TABLE 1. Software and version.

Software	Version
python	3.7.13
torch	1.11.0
pfrl	0.3.0
gym	0.21.0

B. FLOW OF NUMERICAL EXPERIMENT

Algorithm 1 shows the flow of the numerical experiment.

Algorithm 1 Training and Evaluating Manifestor

```
$\pi \leftarrow$ a policy of an A2C agent
// Preparation
Build a dataset of tuples $(s, a, g_{agent}, g_{human})$
dataset_training, dataset_evaluation $\leftarrow$ dataset.split()
$\vec{s}, \vec{a}, \vec{g}_{agent}, \vec{g}_{human} \leftarrow$ dataset_training
$\vec{u} \leftarrow H(\vec{s}, \vec{g}_{human})$
$\vec{u}' \leftarrow H(\vec{s}, \vec{g}_{agent})$

**for** $i = 1, 2, \ldots,$ to NUM_SEED **do**
    // For drawing histogram
    Train *Encoder* with $(\vec{s}, \vec{u})$
    $m^* \leftarrow \mathrm{argmax}_m$ Accuracy$(M_m; \vec{g}_{human})$
    **for** possible $m$ **do**
        Build $M_m$ with *Encoder* and $m$ (See Eq. 9.)
        $f_{N,\mathbf{Manifestor},m} \leftarrow$ Train $f_N$ with $(L^+_{f_N}, M_m, \vec{s}, \vec{a}, \vec{u})$
        Calculate $L_{M_m}$ with dataset_evaluation
    **end for**
    Calculate $L_{M_m}/L_{M_{m^*}}$ for each $m(\neq m^*)$

    // For comparing accuracy
    $f_{N,\mathbf{optimal},m^*} \leftarrow$ Train $f_N$ with $(L_{f_N}, M_{m^*}, \vec{s}, \vec{a}, \vec{u}')$
    $f_{N,\mathbf{ablation},m^*} \leftarrow$ Train $f_N$ with $(L_{f_N}, M_{m^*}, \vec{s}, \vec{a}, \vec{u})$
    Calculate how much the outputs of $f_{N,\mathbf{Manifestor},m^*}$
    match those of $f_{N,\mathbf{optimal},m^*}$
    Calculate how much the outputs of $f_{N,\mathbf{ablation},m^*}$ match
    those of $f_{N,\mathbf{optimal},m^*}$
**end for**
```

C. TRAINING A2C

An A2C agent was trained for the modified LunarLander-v2 (Subsection II-D) using pfrl, a DRL library [47]. Table 2 lists the hyperparameters for the training.

D. TRAINING MANIFESTOR

The *Encoder* (Eq. 7) is based on the Transformer-Encoder implementation from the PyTorch library. Both s_t and u_t are embedded into 64-dimensional vectors and concatenated before entering the *Encoder*. A sequence of (s, u) with a

TABLE 2. Hyperparamters of A2C agent.

Hyperparameter	Value
Update steps	5
Discount factor γ	0.995
Optimizer	RMSprop
RMSprop epsilon	1e-5
Learning rate	7e-4
Hidden activation	ReLU
Hidden sizes	[512, 512, 512]

TABLE 3. Hyperparamters of *Encoder*.

Hyperparameter	Value
The number of Transformer-Encoder layers	2
The number of Transformer-Encoder heads	2
Dimension of the Transformer-Encoder input	128
Dimension of Transformer-Encoder feedforward network model	1024
Dropout rate	0.5
Hidden sizes for *Decoder*	[256, 256]

TABLE 4. Hyperparamters of f_N.

Hyperparameter	Value
The number of Transformer-Encoder layers	2
The number of Transformer-Encoder heads	2
Dimension of the Transformer-Encoder input	32
Dimension of Transformer-Encoder feedforward network model	1024
Dropout rate	0.5

TABLE 5. Numerical experiment - Mann-Whitney's U test (unskilled dataset).

	U	p
Manifestor - *ablation*	0	$1.3 \cdot 10^{-34}$

TABLE 6. Numerical experiment - Mann-Whitney's U test (skilled dataset).

	U	p
Manifestor - *ablation*	194	$3.9 \cdot 10^{-32}$

[CLS] token at the beginning is input to the Transformer-Encoder model, which outputs vectors for each sequence element. The output vector for [CLS] is transformed to three-dimensional vector with a perceptron and the softmax function. This is the output of *Encoder* and expresses the probability of $\hat{g} \in \hat{\mathcal{G}}$. Table 3 lists the hyperparameters for the *Encoder* and *Decoder*.

The f_N is implemented with a similar model to the *Encoder*. The differences are that the input is only $\boldsymbol{s}_{t,N}$ and that the output vector is considered the probability that $\boldsymbol{s}_{t,N}$ is expressed with vocabulary $u \in \mathcal{U}$. Table 4 lists the hyperparameters for the f_N.

TABLE 7. User study experiment - Kruskal-Wallis test.

H	p
188.7	$1.1 \cdot 10^{-40}$

TABLE 8. User study experiment - Mann-Whitney's U test with Bonferroni correction.

	U	p
Manifestor - *optimal*	66767	.25
Manifestor - *ablation*	42317	$2.4 \cdot 10^{-29}$
Manifestor - baseline	64850	$2.6 \cdot 10^{-14}$
optimal - *ablation*	49779	$4.1 \cdot 10^{-27}$
optimal - baseline	76565	$1.9 \cdot 10^{-11}$
ablation - baseline	135701	$1.9 \cdot 10^{-6}$

APPENDIX B
STATISTICS DETAILS

See Tables 5–8.

REFERENCES

[1] B. Hayes and J. A. Shah, "Improving robot controller transparency through autonomous policy explanation," in *Proc. ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, Mar. 2017, pp. 303–312.

[2] B. Hayes and B. Scassellati, "Challenges in shared-environment human–robot collaboration," in *Proc. Collaborative Manipulation Workshop ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, vol. 8, Jan. 2013, pp. 1–9.

[3] J. Waa, J. V. Diggelen, K. Bosch, and M. Neerincx, "Contrastive explanations for reinforcement learning in terms of expected consequences," in *Proc. Workshop Explainable AI IJCAI Conf.*, Stockholm, Sweden, vol. 37, 2018, pp. 1–6.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[5] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov, "Gated-attention architectures for task-oriented language grounding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[6] D. Misra, J. Langford, and Y. Artzi, "Mapping instructions and visual observations to actions with reinforcement learning," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1–16.

[7] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sunderhauf, I. Reid, S. Gould, and A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3674–3683.

[8] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *ITU J., ICT Discoveries*, vol. 1, no. 1, pp. 1–10, 2017.

[9] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[10] S. Anjomshoae, A. Najjar, D. Calvaresi, and K. Främling, "Explainable agents and robots: Results from a systematic literature review," in *Proc. 18th Int. Conf. Auton. Agents Multiagent Syst.* Richland, WA, USA: Int. Found. Auton. Agents Multiagent Syst., 2019, pp. 1078–1088.

[11] P. Langley, B. Meadows, M. Sridharan, and D. Choi, "Explainable agency for intelligent autonomous systems," in *Proc. 31st AAAI Conf. Artif. Intell.* Palo Alto, CA, USA: AAAI Press, 2017, pp. 4762–4763.

[12] M. Edmonds, F. Gao, H. Liu, X. Xie, S. Qi, B. Rothrock, Y. Zhu, Y. N. Wu, H. Lu, and S.-C. Zhu, "A tale of two explanations: Enhancing human trust by explaining robot behavior," *Sci. Robot.*, vol. 4, no. 37, Dec. 2019, Art. no. eaay4663.

[13] K. Weitz, D. Schiller, R. Schlagowski, T. Huber, and E. André, "'Do you trust me?': Increasing user-trust by integrating virtual agents in explainable AI interaction design," in *Proc. 19th ACM Int. Conf. Intell. Virtual Agents*. New York, NY, USA: ACM, 2019, pp. 7–9, doi: 10.1145/3308532.3329441.

[14] E. Puiutta and E. M. S. P. Veith, "Explainable reinforcement learning: A survey," in *Machine Learning and Knowledge Extraction*, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds. Cham, Switzerland: Springer, 2020, pp. 77–95.

[15] D. Hein, A. Hentschel, T. Runkler, and S. Udluft, "Particle swarm optimization for generating interpretable fuzzy reinforcement learning policies," *Eng. Appl. Artif. Intell.*, vol. 65, pp. 87–98, Oct. 2017, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197617301537

[16] B. Beyret, A. Shafti, and A. A. Faisal, "Dot-to-dot: Explainable hierarchical reinforcement learning for robotic manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5014–5019.

[17] T. Shu, C. Xiong, and R. Socher, "Hierarchical and interpretable skill acquisition in multi-task reinforcement learning," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, Apr. 2018, pp. 1–14. [Online]. Available: https://openreview.net/forum?id=SJJQVZW0b

[18] G. Liu, O. Schulte, W. Zhu, and Q. Li, "Toward interpretable deep reinforcement learning with linear model U-trees," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases* (Lecture Notes in Computer Science), vol. 11052, M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, Eds. Dublin, Ireland: Springer, Sep. 2018, pp. 414–429, doi: 10.1007/978-3-030-10928-8_25.

[19] P. Tamagnini, J. Krause, A. Dasgupta, and E. Bertini, "Interpreting blackbox classifiers using instance-level visual explanations," in *Proc. 2nd Workshop Hum.-Loop Data Anal.* New York, NY, USA: ACM, 2017, pp. 1–6, doi: 10.1145/3077257.3077260.

[20] R. Iyer, Y. Li, H. Li, M. Lewis, R. Sundar, and K. Sycara, "Transparency and explanation in deep reinforcement learning neural networks," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.* New York, NY, USA: ACM, Dec. 2018, pp. 144–150, doi: 10.1145/3278721.3278776.

[21] A. Mott, D. Zoran, M. Chrzanowski, D. Wierstra, and D. J. Rezende, "Towards interpretable reinforcement learning using attention augmented agents," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[22] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[23] J. K. Aggarwal and L. Xia, "Human activity recognition from 3D data: A review," *Pattern Recognit. Lett.*, vol. 48, pp. 70–80, Oct. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865514001299

[24] M. Barnachon, S. Bouakaz, B. Boufama, and E. Guillou, "Ongoing human action recognition with motion capture," *Pattern Recognit.*, vol. 47, no. 1, pp. 238–247, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320313002720

[25] K. Yoshino, K. Wakimoto, Y. Nishimura, and S. Nakamura, "Caption generation of robot behaviors based on unsupervised learning of action segments," in *Conversational Dialogue Systems for the Next Decade* (Lecture Notes in Electrical Engineering), vol. 704, L. F. D'Haro, Z. Callejas, and S. Nakamura, Eds. Singapore: Springer, 2021, doi: 10.1007/978-981-15-8395-7_17.

[26] H. Mei, M. Bansal, and M. R. Walter, "Listen, attend, and walk: Neural mapping of navigational instructions to action sequences," in *Proc. 30th AAAI Conf. Artif. Intell.* Palo Alto, CA, USA: AAAI Press, 2016, pp. 2772–2778.

[27] P. Shah, M. Fiser, A. Faust, J. C. Kew, and D. Hakkani-Tur, "FollowNet: Robot navigation by following natural language directions with deep reinforcement learning," *CoRR*, vol. abs/1805.06150, pp. 1–7, May 2018.

[28] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, M. Wainwright, C. Apps, D. Hassabis, and P. Blunsom, "Grounded language learning in a simulated 3D world," *CoRR*, vol. abs/1706.06551, pp. 1–22, Jun. 2017.

[29] J. Luketina, N. Nardelli, G. Farquhar, J. Foerster, J. Andreas, E. Grefenstette, S. Whiteson, and T. Rocktäschel, "A survey of reinforcement learning informed by natural language," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1–9.

[30] Y. Fukuchi, M. Osawa, H. Yamakawa, and M. Imai, "Autonomous self-explanation of behavior for interactive reinforcement learning agents," in *Proc. 5th Int. Conf. Hum. Agent Interact.* New York, NY, USA: ACM, Oct. 2017, pp. 97–101, doi: 10.1145/3125739.3125746.

[31] Y. Fukuchi, M. Osawa, H. Yamakawa, and M. Imai, "Application of instruction-based behavior explanation to a reinforcement learning agent with changing policy," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2017, pp. 100–108.

[32] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," 2016, *arXiv:1606.01540*.

[33] S. Reddy, A. Dragan, and S. Levine, "Where do you think you're going?: Inferring beliefs about dynamics from behavior," in *Proc. NeurIPS*, 2018, pp. 1–12.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[36] S. Griffith, K. Subramanian, J. Scholz, C. L. Isbell, and A. L. Thomaz, "Policy shaping: Integrating human feedback with reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2013. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/e034fb6b66aacc1d48f445ddfb08da98-Paper.pdf

[37] Y. Fukuchi, M. Osawa, H. Yamakawa, T. Takahashi, and M. Imai, "Bayesian inference of self-intention attributed by observer," in *Proc. 6th Int. Conf. Hum.-Agent Interact.* New York, NY, USA: ACM, Dec. 2018, pp. 3–10, doi: 10.1145/3284432.3284438.

[38] Y. Fukuchi, M. Osawa, H. Yamakawa, T. Takahashi, and M. Imai, "Conveying intention by motions with awareness of information asymmetry," *Frontiers Robot. AI*, vol. 9, pp. 1–9, Feb. 2022. [Online]. Available: https://www.frontiersin.org/article/10.3389/frobt.2022.783863

[39] D. K. Misra, J. Sung, K. Lee, and A. Saxena, "Tell me Dave: Context-sensitive grounding of natural language to manipulation instructions," *Int. J. Robot. Res.*, vol. 35, nos. 1–3, pp. 281–300, Jan. 2016, doi: 10.1177/0278364915602060.

[40] H. Chen, H. Tan, A. Kuntz, M. Bansal, and R. Alterovitz, "Enabling robots to understand incomplete natural language instructions using commonsense reasoning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 1963–1969.

[41] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artif. Intell.*, vol. 297, Aug. 2021, Art. no. 103500. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370221000515

[42] Y. Ye, M. Singh, A. Gupta, and S. Tulsiani, "Compositional video prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10353–10362.

[43] N. Wichers, R. Villegas, D. Erhan, and H. Lee, "Hierarchical long-term video prediction without supervision," in *Proc. 35th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 80, J. Dy and A. Krause, Eds., Jul. 2018, pp. 6038–6046. [Online]. Available: https://proceedings.mlr.press/v80/wichers18a.html

[44] M. Oliu, J. Selva, and S. Escalera, "Folded recurrent neural networks for future video prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 716–731.

[45] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in Atari games," in *Advances in Neural Information Processing Systems*, vol. 28, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/6ba3af5d7b2790e73f0de32e5c8c1798-Paper.pdf

[46] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based reinforcement learning: A survey," *CoRR*, vol. abs/2006.16712, pp. 1–61, Jun. 2020.

[47] Y. Fujita, P. Nagarajan, T. Kataoka, and T. Ishikawa, "ChainerRL: A deep reinforcement learning library," *J. Mach. Learn. Res.*, vol. 22, no. 77, pp. 1–14, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-376.html

YOSUKE FUKUCHI was born in Japan, in 1994. He received the B.E. and M.E. degrees in computer science from Keio University, Yokohama, Japan, in 2017 and 2019, respectively.

From 2019 to 2021, he was an Assistant Professor at Keio University. From 2021 to 2022, he was a Project Researcher with the Keio Leading-edge Laboratory of Science and Technology (KLL). He is currently a Project Researcher at the National Institute of Informatics, Tokyo, Japan. His research interests include human–agent interaction, artificial intelligence, and theory of mind. He is a member of the Japanese Society for Artificial Intelligence.

MASAHIKO OSAWA received the Ph.D. degree in computer science from Keio University, in 2020.

He was a Research Fellow (DC1) at the Japan Society of the Promotion of Science, from 2017 to 2020. He is currently an Assistant Professor of Nihon University. His dream is making DORAEMON. His research interests include machine learning, autonomous robots, human–agent interaction, cognitive science, biologically inspired cognitive architecture, and computational neuro science. He is a member of the Japanese Society for Artificial Intelligence, the Japanese Neural Network Society, the Japanese Cognitive Science Society, the Asia Pacific Neural Network Assembly, and ACM.

HIROSHI YAMAKAWA received the M.S. degree in physics and the Ph.D. degree in engineering from The University of Tokyo, Japan, in 1989 and 1992, respectively.

In 1992, he joined Fujitsu Laboratories Ltd. In 2014, he founded the Dwango AI Laboratory, where he was the Director, until March 2019. In 2015, he co-founded the Whole Brain Architecture Initiative (WBAI), non-profit organization, where he is currently the Chairperson. He is also a Project Researcher at the Graduate School of Engineering, The University of Tokyo; a Visiting Professor at the Graduate School, The University of Electro-Communications; the Director of the Intelligent Systems Division (Visiting Professor), Institute of Informatics, Kinki University; and the Chief Visiting Researcher at the RIKEN Center for Biosystems Dynamics Research. He is an AI researcher interested in brain. His research interests include brain-inspired artificial general intelligence, concept formation, neurocomputing, and opinion aggregation technology.

MICHITA IMAI (Member, IEEE) received the Ph.D. degree in computer science from Keio University, in 2002.

In 1994, he joined NTT Human Interface Laboratories. In 1997, he joined ATR Media Integration and Communications Research Laboratories. From 2009 to 2010, he was a Visiting Scholar at The University of Chicago. He is currently a Professor with the Faculty of Science and Technology, Keio University, and a Researcher with ATR Intelligent Robot Laboratories. His research interests include autonomous robots, human–robot interaction, speech dialogue systems, humanoids, and spontaneous behaviors. He is a member of the Information and Communication Engineers Japan (IEICE-J), the Information Processing Society of Japan, the Japanese Cognitive Science Society, the Japanese Society for Artificial Intelligence, the Human Interface Society, and ACM.