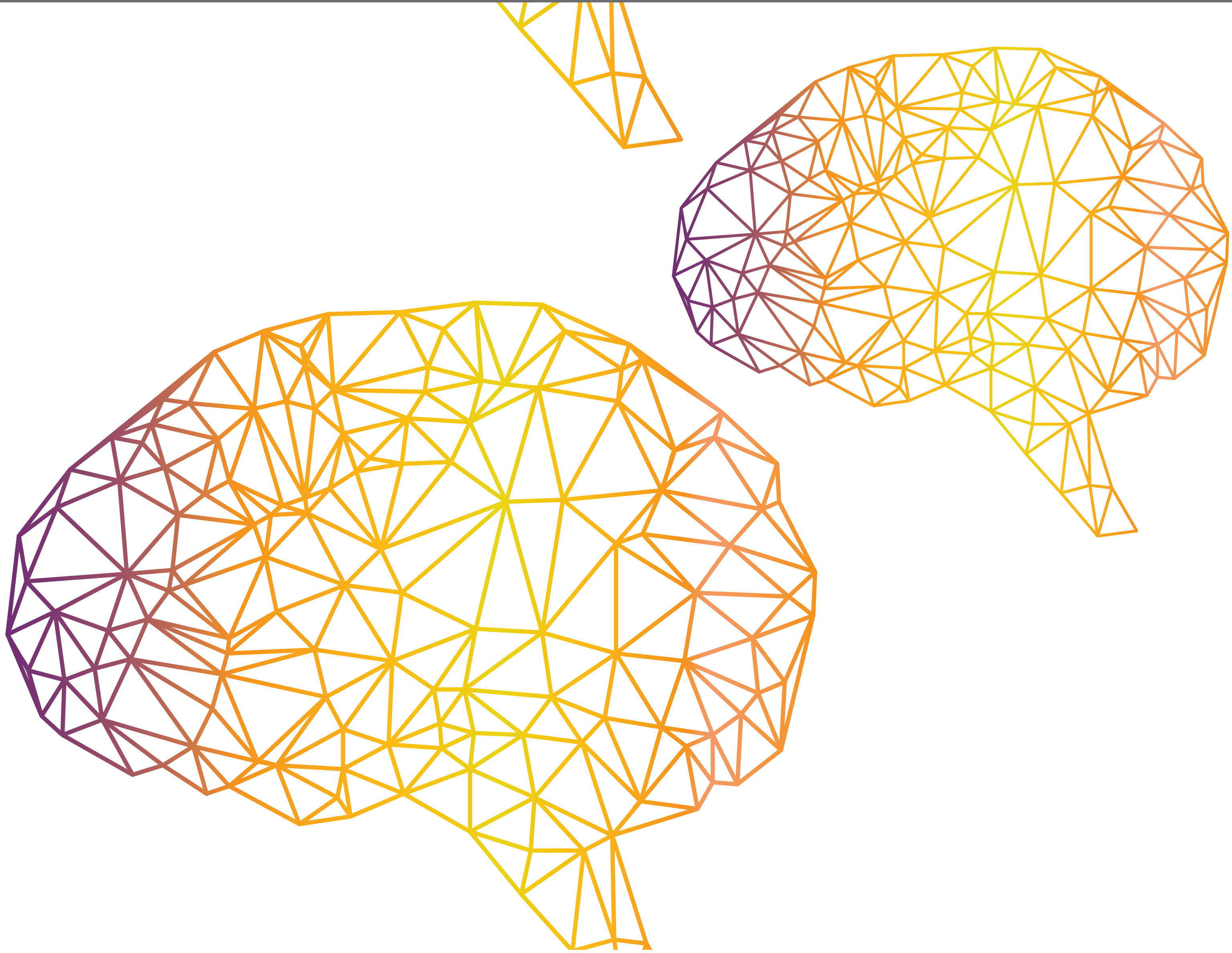
MACHINE LEARNING METHODS FOR HIGH-LEVEL COGNITIVE CAPABILITIES IN ROBOTICS

EDITED BY: Emre Ugur, Tetsuya Ogata, Yiannis Demiris, Tadahiro Taniguchi
and Takayuki Nagai

PUBLISHED IN: Frontiers in Neurorobotics



frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88963-261-9
DOI 10.3389/978-2-88963-261-9

MACHINE LEARNING METHODS FOR HIGH-LEVEL COGNITIVE CAPABILITIES IN ROBOTICS

Topic Editors:
Emre Ugur, Boğaziçi University, Turkey
Tetsuya Ogata, Waseda University, Japan
Yiannis Demiris, Imperial College London, United Kingdom
Tadahiro Taniguchi, Ritsumeikan University, Japan
Takayuki Nagai, Osaka University, Japan

Citation: Ugur, E., Ogata, T., Demiris, Y., Taniguchi, T., Nagai, T., eds. (2019). Machine Learning Methods for High-Level Cognitive Capabilities in Robotics. Lausanne: Frontiers Media SA. doi: 10.3389/978-2-88963-261-9

Table of Contents

frontiers in Neurorobotics

EDITORIAL
published: 22 October 2019
doi: 10.3389/fnbot.2019.00083

Editorial: Machine Learning Methods for High-Level Cognitive Capabilities in Robotics

Tadahiro Taniguchi[1]*, *Emre Ugur*[2], *Tetsuya Ogata*[3], *Takayuki Nagai*[4] *and Yiannis Demiris*[5]

[1] *Department of Information Science and Engineering, Ritsumeikan University, Kyoto, Japan,* [2] *Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey,* [3] *Department of Intermedia Art and Science, School of Fundamental Science and Engineering, Waseda University, Tokyo, Japan,* [4] *Department of Systems Innovation, Graduate School of Engineering Science, Osaka University, Osaka, Japan,* [5] *Department of Electrical and Electronic Engineering, Imperial College London, London, United Kingdom*

Keywords: machine leading, cognitive robotics, language acquisition, neural networks, cognitive architecture, probabilistic models, robot learning

Editorial on the Research Topic

Machine Learning Methods for High-Level Cognitive Capabilities in Robotics

1. INTRODUCTION

Adaptive learning and emergence of integrative cognitive system that involve not only low-level but also high-level cognitive capabilities are crucially important in robotics (Cangelosi et al., 2010; Cangelosi and Schlesinger, 2015; Ugur and Piater, 2015; Tani, 2016; Taniguchi et al., 2016, 2018). Recent advancement in machine learning methods, e.g., deep learning and hierarchical Bayesian modeling, enables us to develop cognitive systems that integrate multi-level sensory-motor and cognitive capabilities. Low-level cognitive capabilities includes sensory perception, physical control, and behavioral motion generation, while high-level cognitive capabilities include logical inference, planning, and language acquisition. To create robots that can deal with uncertainty in our daily environment, developing machine learning methods that can integrate low-level and high-level is essential. Following the successfully organized session "the Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics 2016" held in IEEE-IROS 2016[1], we organized this Research Topic. We aimed to publish original papers about the state-of-the-art machine learning methods that contribute to modeling sensory-motor and cognitive capabilities in robotics.

2. ABOUT THE RESEARCH TOPIC

We are pleased to present 9 research articles, related to motor and behavior learning, concept formation, language acquisition, and cognitive architecture. In this section, we briefly introduce each paper.

First, three papers focused on action and behavior learning. Imitation learning is an important topic related to the integration of high-level and low-level cognitive capability because it enables a robot to acquire behavioral primitives from social interaction including observation of human behaviors. Nakajo et al. proposed a machine learning method for viewpoint transformation

OPEN ACCESS

Approved by:
Florian Röhrbein,
Technical University of Munich,
Germany

****Correspondence:***
Tadahiro Taniguchi
taniguchi@ci.ritsumei.ac.jp

Received: *19 August 2019*
Accepted: *25 September 2019*
Published: *22 October 2019*

Citation:
Taniguchi T, Ugur E, Ogata T, Nagai T and Demiris Y (2019) Editorial: Machine Learning Methods for High-Level Cognitive Capabilities in Robotics. Front. Neurorobot. 13:83. doi: 10.3389/fnbot.2019.00083

[1] The Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics 2016: http://mlhlcr2016.tanichu.com/

and action mapping using a neural network having encoder-decoder architecture, i.e., sequence to sequence. In imitation learning, demonstrator and imitator have different perspectives. The method deals with the problem and produced a successful result. Nakamura et al. proposed a new machine learning method called Gaussian process-hidden semi-Markov model (GP-HSMM). GP-HSMM can segment continuous motion trajectories without defining a parametric model for each primitive. That comprises Gaussian process, which is a regression method based on Bayesian non-parametric, and hidden semi-Markov model. This method enables a robot to find motion primitives from complex human motion in an imitation learning scenario. Manipulation using the left and right arms is an essential capability for a cognitive robot. Zhang et al. proposed a neural-dynamic based synchronous-optimization scheme manipulators. It was demonstrated that the method enables a robot to track complex paths.

Second, two papers focused on the relationship between action and object concept. Andries et al. proposes the formalism for defining and identifying affordance equivalence. The concept of affordance can be regarded as a relationship between an actor, an action performed by this actor, an object on which the action is performed, and the resulting effect. Learning affordance, i.e., inter-dependency between action and object concept, is an important topic in this field. Taniguchi et al. proposed a new active perception method based on multimodal hierarchical Dirichlet process, which is a hierarchical Bayesian model for multimodal object concept formation method. The important aspect of the approach is that the policy for active perception is derived based on the result of unsupervised learning without any manually designed label data and reward signals.

Third, three papers are related to language acquisition and concept formation. Hagiwara et al. proposed hierarchical spatial concept formation method based on hierarchical multimodal latent Dirichlet allocation (hMLDA). They demonstrated that a robot could form concept for places having hierarchical structure, e.g., "around a table" is a part of "dining room," using hMLDA, and became able to understand utterances indicating places in a domestic environment given by a human user. Yamada et al. described representation learning method that enables a robot to understand not only action-related words, but also logical words, e.g., "or," "and" and "not." They introduced an neural network having an encoder-decoder architecture, and obtained successful and suggestive results. Taniguchi et al. proposed a new multimodal cross-situational learning method for language acquisition. A robot became able to estimate of each word in relation with modality via which each word is grounded.

The final paper presents a framework for cognitive architecture based on hierarchical Bayesian models. Nakamura et al. proposed Symbol Emergence in Robotics tool KIT (SERKET) that can integrate many cognitive modules developed using hierarchical Bayesian models, i.e., probabilistic generative models, effectively without re-implementation of each module. Integration of low-level and high-level cognitive capability and developing an integrative cognitive system requires researchers and developers to construct very complex software modules, and this is expected to cause practical problems. Serket can be regarded as a practical solution for the problem, and expected to push the research field forward.

3. NEXT STEP

With the tremendous success of the past three Special issues of this Research Topic, we organized follow-up workshops[2] and a Research Topic[3]. Two survey papers related to the series of workshops have already been published (Taniguchi et al., 2018; Tangiuchi et al., 2019). We will also organize a workshop with the special emphasis on deep probabilistic generative models[4] We believe that in order to create an artificial cognitive system, i.e., a robot, it is important to integrate low-level and high-level cognitive capabilities based on machine learning-based methods. We hope that this special issue will contribute to accelerating the robotics and machine learning studies that aims to create human-like cognitive systems that can behave in our real-world environment in collaboration with people.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of participants in this special issue.

[2] The 2nd Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics 2017: http://mlhlcr2017.tanichu.com/. The Workshop on Language and Robotics: http://iros2018.emergent-symbol.systems/.

[3] Research Topic Language and Robotics: https://www.frontiersin.org/research-topics/8861/language-and-robotics.

[4] The Workshop on Deep Probabilistic Generative Models for Cognitive Architecture in Robotics 2019: https://sites.google.com/site/dpgmcar2019/.

REFERENCES

Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S, Nehaniv, C., Fischer, K., et al. (2010). Integration of action and language knowledge: a roadmap for developmental robotics. *IEEE Trans. Auton. Ment. Dev.* 2, 167–195. doi: 10.1109/TAMD.2010.2053034

Cangelosi, A., and Schlesinger, M. (2015). *Developmental Robotics: From Babies to Robots*. Cambridge, MA: MIT Press.

Tangiuchi, T., Mochihashi, D., Nagai, T., Uchida, S., Inoue, N., Kobayashi, I., et al. (2019). Survey on fron- tiers of language and robotics. *Adv. Robot.* 33,700–730. doi: 10.1080/01691864.2019.1632223

Tani, J. (2016). *Exploring Robotic Minds: Actions, Symbols, and Consciousness as Self-Organizing Dynamic Phenomena*. Oxford, UK: Oxford University Press.

Taniguchi, T., Nagai, T., Nakamura, T., Iwahashi, N., Ogata, T., and Asoh, H. (2016). Symbol emergence in robotics: a survey. *Adv. Robot.* 30,706–728. doi: 10.1080/01691864.2016.1164622

Taniguchi, T., Ugur, E., Hoffmann, M., Jamone, L., Nagai, T., Rosman, B., et al. (2018). Symbol emergence in cognitive developmental systems: a survey. *IEEE Trans. Cogn. Dev. Syst.* doi: 10.1109/TCDS.2018.2867772. [Epub ahead of print].

Ugur, E., and Piater, J. (2015). "Bottom-up learning of object categories, action effects and logical rules: from continuous manipulative exploration to symbolic planning," in *IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA), 2627–2633.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers
in Neurorobotics

ORIGINAL RESEARCH
published: 19 December 2017
doi: 10.3389/fnbot.2017.00066

Cross-Situational Learning with Bayesian Generative Models for Multimodal Category and Word Learning in Robots

Akira Taniguchi[1]*, *Tadahiro Taniguchi*[1] *and Angelo Cangelosi*[2]

[1] *Emergent Systems Laboratory, Ritsumeikan University, Kusatsu, Japan* [2] *The Centre for Robotics and Neural Systems, Plymouth University, Plymouth, United Kingdom*

In this paper, we propose a Bayesian generative model that can form multiple categories based on each sensory-channel and can associate words with any of the four sensory-channels (action, position, object, and color). This paper focuses on cross-situational learning using the co-occurrence between words and information of sensory-channels in complex situations rather than conventional situations of cross-situational learning. We conducted a learning scenario using a simulator and a real humanoid iCub robot. In the scenario, a human tutor provided a sentence that describes an object of visual attention and an accompanying action to the robot. The scenario was set as follows: the number of words per sensory-channel was three or four, and the number of trials for learning was 20 and 40 for the simulator and 25 and 40 for the real robot. The experimental results showed that the proposed method was able to estimate the multiple categorizations and to learn the relationships between multiple sensory-channels and words accurately. In addition, we conducted an action generation task and an action description task based on word meanings learned in the cross-situational learning scenario. The experimental results showed that the robot could successfully use the word meanings learned by using the proposed method.

Keywords: Bayesian model, cross-situational learning, lexical acquisition, multimodal categorization, symbol grounding, word meaning

OPEN ACCESS

Edited by:
Frank Van Der Velde,
University of Twente, Netherlands

Reviewed by:
Maxime Petit,
Imperial College London,
United Kingdom
Yulia Sandamirskaya,
University of Zurich, Switzerland

Correspondence:
Akira Taniguchi
a.taniguchi@em.ci.ritsumei.ac.jp

Received: *17 July 2017*
Accepted: *21 November 2017*
Published: *19 December 2017*

Citation:
Taniguchi A, Taniguchi T and Cangelosi A (2017) Cross-Situational Learning with Bayesian Generative Models for Multimodal Category and Word Learning in Robots. Front. Neurorobot. 11:66. doi: 10.3389/fnbot.2017.00066

1. INTRODUCTION

This paper addresses the study of robotic learning of the word meanings inspired by the process of language acquisition of humans. We developed an unsupervised machine learning method to enable linguistic interaction between humans and robots. Human infants can acquire word meanings by estimating the relationships between multimodal information and words in a variety of situations. For example, if an infant grasps a green cup by hand, let us consider the way the parent describes the actions of the infant to the infant using a sentence such as "grasp green front cup." In this case, the infant does not know the relationship between words and situations because it has not acquired the meanings of words. In other words, the infant cannot determine whether the word "green" indicates an action, an object, or a color. However, it is believed that the infant can learn that the word "green" represents the green color by observing the co-occurrence of the word "green" with objects of green color in various situations. This is known as cross-situational learning (CSL), which has been both studied in children (Smith et al., 2011) and modeled in simulated agents and robots (Fontanari et al., 2009). The CSL is related to the symbol grounding problem (Harnad, 1990), which is a challenging and significant issue in robotics.

The generalization ability and the robustness of observation noise to process situations that have never been experienced are important in cognitive robotics. The study of language acquisition by infants led to the proposal of a hypothesis of taxonomic bias (Markman and Hutchinson, 1984) that infants tend to understand a word as the name of a category to which the target object belongs rather than a proper noun. This hypothesis could also be considered to play an important role in CSL. In this study, we assume that words are associated with categories based on taxonomic bias. By associating words with categories, it becomes possible for a human to generalize and process words. Therefore, humans can use words for communication in new situations. To develop this ability, the robot needs to form categories from observation information autonomously. We develop this ability by categorization based on the Bayesian generative model. Another hypothesis regarding the lexical acquisition by an infant was mutual exclusivity bias (constraint) (Markman and Wachtel, 1988). In studies on lexical acquisition, this hypothesis was considered to be particularly important for CSL (Twomey et al., 2016). Mutual exclusivity bias assumes that the infant considers the name of an object to correspond to one particular category only. In other words, multiple categories do not correspond to that word simultaneously. In Imai and Mazuka (2007), it was suggested that once an infant decides whether a word refers to the name of an object or a substance, the same word is not applied across the ontological distinction such as objects and substances. In this study, we extend the mutual exclusivity constraint to the CSL problem in complex situations. We aim to develop a novel method that can acquire knowledge of multiple categories and word meanings simultaneously. In addition, we verify whether the effect of mutual exclusivity is biased toward lexical acquisition by constructing a model assuming different constraints.

In addition, humans can perform the instructed action using acquired knowledge. For example, the parent places some objects in front of an infant and speaks "grasp green right ball" to the infant. In this case, the infant can use the acquired word meanings to select the green ball to the right of some objects and perform the action of grasping. Furthermore, humans can explain self-action with the sentence using the acquired knowledge. For example, if the infant knows the word meanings after grasping a blue box in front of it, the infant can speak "grasp blue front box" to another person. Understanding instructions and describing situations are crucial problems that are also required to build a cognitive robot.

In this paper, the goal is to develop an unsupervised machine-learning method for learning the relationships between words and the four sensory-channels (action, object, color, and position) from the robot's experience of observed sentences describing object manipulation scenes. In the above example, sentences containing four words for four sensory-channels are shown. However, in the scenario described in this study, sentences of less than four words are allowed. In addition, the position sensory-channel corresponds to the original position of the object. In other words, we assume that the environment is static. We assume that the robot can recognize spoken words without errors, as this work focuses specifically on (1) the categorization for each sensory-channel, (2) the learning of relationships between words and sensory-channels, and (3) the grounding of words in multiple categories. In addition, we demonstrate whether the robot can carry out its actions and the sentence description of its action by conducting experiments using the CSL results. The main contributions of this paper are as follows:

- We proposed an unsupervised machine-learning method based on a Bayesian generative model that makes it possible to learn word meanings, i.e., the relationships between words and categories, from complex situations.
- We demonstrated that word meanings learned by using the proposed method are effective for generating an action and description of a situation.

The remainder of this paper is organized as follows. In Section 2, we discuss previous studies on lexical acquisition by a robot and CSL that are relevant to our study. In Section 3, we present a proposed Bayesian generative model for CSL. In Sections 4 and 5, we discuss the effectiveness of the proposed method in terms of three tasks, i.e., cross-situational learning, action generation, and an action description task, in a simulation and a real environment, respectively. Section 6 concludes the paper.

2. RELATED WORK

2.1. Lexical Acquisition by Robot

Studies of language acquisition also constitute a constructive approach to the human developmental process (Cangelosi and Schlesinger, 2015), the language grounding (Steels and Hild, 2012), and the symbol emergence (Taniguchi et al., 2016c). One approach to studying language acquisition focuses on the estimation of phonemes and words from speech signals (Goldwater et al., 2009; Heymann et al., 2014; Taniguchi et al., 2016d). However, these studies used only continuous speech signals without using co-occurrence based on other sensor information, e.g., visual, tactile, and proprioceptive information. Therefore, the robot was not required to understand the meaning of words. Yet, it is important for a robot to understand word meanings, i.e., grounding the meanings to words, for human–robot interaction (HRI).

Roy and Pentland (2002) proposed a computational model by which a robot could learn the names of objects from images of the object and natural infant-directed speech. Their model could perform speech segmentation, lexical acquisition, and visual categorization. Hörnstein et al. (2010) proposed a method based on pattern recognition and hierarchical clustering that mimics a human infant to enable a humanoid robot to acquire language. Their method allowed the robot to acquire phonemes and words from visual and auditory information through interaction with the human. Nakamura et al. (2011a,b) proposed multimodal latent Dirichlet allocation (MLDA) and a multimodal hierarchical Dirichlet process (MHDP) that enables the categorization of objects from multimodal information, i.e., visual, auditory, haptic, and word information. Their methods enabled more accurate object categorization by using multimodal information. Taniguchi et al. (2016a) proposed a method for simultaneous estimation of self-positions and words from noisy sensory information and an uttered word. Their method integrated ambiguous speech recognition results with the self-localization method for learning spatial concepts. However, Taniguchi et al. (2016a) assumed that the name of a place would be learned from an uttered word. Taniguchi et al. (2016b) proposed a nonparametric Bayesian spatial concept acquisition method (SpCoA) based on place categorization and unsupervised word segmentation. SpCoA could acquire the names of places from spoken sentences including multiple words. In the above studies, the robot was taught to focus on one target, e.g., an object or a place, by a tutor using one word or one sentence. However, considering a more realistic problem, the robot needs to know which event in a complicated situation is associated with which word in the sentence. The CSL, which is extended from the aforementioned studies on the lexical acquisition, is a more difficult and important problem in robotics in comparison. Our research concerns the CSL problem because of its importance in relation to the lexical acquisition by a robot.

2.2. Cross-Situational Learning

2.2.1. Conventional Cross-Situational Learning Studies

Frank et al. (2007, 2009) proposed a Bayesian model that unifies statistical and intentional approaches to cross-situational word learning. They conducted basic CSL experiments with the purpose of teaching an object name. In addition, they discussed that the effectiveness of mutual exclusivity for CSL in probabilistic models. Fontanari et al. (2009) performed object-word mapping from the co-occurrence between objects and words by using a method based on neural modeling fields (NMF). In “modi” experiments using iCub, their findings were similar to those reported by Smith and Samuelson (2010). The abovementioned studies are CSL studies that were inspired by studies based on experiments with human infants. These studies assumed a simple situation such as learning the relationship between objects and words as the early stage of CSL. However, the real environment is varied and more complex. In this study, we focus on the problem of CSL in utterances including multiple words and observations from multiple sensory-channels.

2.2.2. Probabilistic Models

Qu and Chai (2008, 2010) proposed a learning method that automatically acquires novel words for an interactive system. They focused on the co-occurrence between word-sequences and entity-sequences tracked by eye-gaze in lexical acquisition. Qu and Chai’s method, which is based on the IBM-translation model (Brown et al., 1993), estimates the word-entity association probability. However, their studies did not result in perfect unsupervised lexical acquisition because they used domain knowledge based on WordNet. Matuszek et al. (2012) presented a joint model of language and perception for grounded attribute learning. This model enables the identification of which novel words correspond to color, shape, or no attribute at all. Celikkanat et al. (2014) proposed an unsupervised learning method based on latent Dirichlet allocation (LDA) that allows many-to-many relationships between objects and contexts. Their method was able to predict the context from the observation information and plan the action using learned contexts. Chen et al. (2016) proposed an active learning method for cross-situational learning of object-word association. In experiments, they showed that LDA was more effective than non-negative matrix factorization (NMF). However, they did not perform any HRI experiment using the learned language. In our study, we perform experiments that use word meanings learned in CSL to generate an action and explain a current situation.

2.2.3. Neural Network Models

Yamada et al. (2015, 2016) proposed a learning method based on a stochastic continuous-time recurrent neural network (CTRNN) and a multiple time-scales recurrent neural network (MTRNN). They showed that the learned network formed an attractor structure representing both the relationships between words and action and the temporal pattern of the task. Stramandinoli et al. (2017) proposed partially recurrent neural networks (P-RNNs) for learning the relationships between motor primitives and objects. Zhong et al. (2017) proposed multiple time-scales gated recurrent units (MTGRU) inspired by MTRNN and long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997). They showed that the MTGRU could learn long-term dependencies in large-dimensional multimodal datasets by conducting multimodal interaction experiments using iCub. The learning results of the above studies using neural networks (NNs) are difficult to interpret because time-series data is mapped to continuous latent space. These studies implicitly associate words with objects and actions. Generally, NN methods require a massive amount of learning data in many cases. On the other hand, the learning result is easier to interpret when Bayesian methods rather than NN methods are used. In addition, Bayesian methods require less data to learn efficiently. We propose a Bayesian generative model that can perform CSL, including action learning.

2.2.4. Robot-to-Robot Interaction

Spranger (2015) and Spranger and Steels (2015) proposed a method for the co-acquisition of semantics and syntax in the spatial language. The experimental results showed that the robot could acquire spatial grammar and categories related to spatial direction. Heath et al. (2016) implemented mobile robots (Lingodroids) capable of learning a lexicon through robot-to-robot interaction. They used two robots equipped with different sensors and simultaneous localization and mapping (SLAM) algorithms. These studies reported that the robots created their lexicons in relation to places and the distance in terms of time. However, these studies did not consider lexical acquisition by HRI. We consider HRI to be necessary to enable a robot to learn human language.

2.2.5. Multimodal Categorization and Word Learning

Attamimi et al. (2016) proposed multilayered MLDA (mMLDA) that hierarchically integrates multiple MLDAs as an extension of Nakamura et al. (2011a). They performed an estimation of the relationships among words and multiple concepts by weighting the learned words according to their mutual information as a post-processing step. In their model, the same uttered words are generated from three kinds of concepts, i.e., this model has three variables for same word information in different concepts. We consider this to be an unnatural assumption as the generative model for generating words. However, in our proposed model, we assume that the uttered words are generated from one variable. We consider our proposed model to involve a more natural assumption than Attamimi's model. In addition, their study did not use data that were autonomously obtained by the robot. In Attamimi et al. (2016), it was not possible for the robot to learn the relationships between self-actions and words because human motions obtained by the motion capture system based on Kinect and a wearable sensor device attached to a human were used as action data. In our study, the robot learns the action category based on subjective self-action. Therefore, the robot can perform a learned action based on a sentence of human speech. In this paper, we focus on complicated CSL problems arising from situations with multiple objects and sentences including words related to various sensory-channels such as the names, position, and color of objects, and the action carried out on the object.

3. MULTICHANNEL CATEGORIZATIONS AND LEARNING THE MEANING OF WORDS

We propose a Bayesian generative model for cross-situational learning. The proposed method can estimate categories of multiple sensory-channels and the relationships between words and sensory-channels simultaneously.

3.1. Overview of the Scenario and Assumptions

Here, we provide an overview of the scenario on which we focus and some of the assumptions in this study. **Figure 1** shows an overview of the scenario. The robot does not have any specific knowledge of objects, but it can recognize that objects exist on the table, i.e., the robot can segment the object and then extract the features of the segmented object. In addition, we assume that the robot can recognize the sentence uttered by the tutor without error. The training procedure consists of the following steps:

1. The robot is in front of the table on which the objects are placed. Multiple objects are placed separately on the table.
2. The robot selects an object from the objects on the table. The robot pays visual attention to a selected object, and then, exerts an action on the selected object, e.g., "grasp," "touch," "reach," and "look-at."
3. The human tutor utters a sentence including words about the object at which the robot is gazing and also a word about the action performed by the robot, e.g., "grasp front green cup."
4. The robot obtains multimodal information regarding all objects on the table in the current situation, e.g., the object features, positions, colors, and self-action. The robot processes the sentence to discover the meanings of the words.

This process (steps 1–4) is carried out many times in different situations.

We assume that the robot does not know the relationships between the words and sensory-channels in advance. This study does not consider grammar, i.e., a unigram language model is assumed. The robot learns word meanings and multiple categories by using visual, tactile, and proprioceptive information, as well as words.

In this study, we consider two-level cross-situational learning (CSL-I and II). The first level (CSL-I) is the selection of an object related to a tutor utterance from multiple objects on the table.

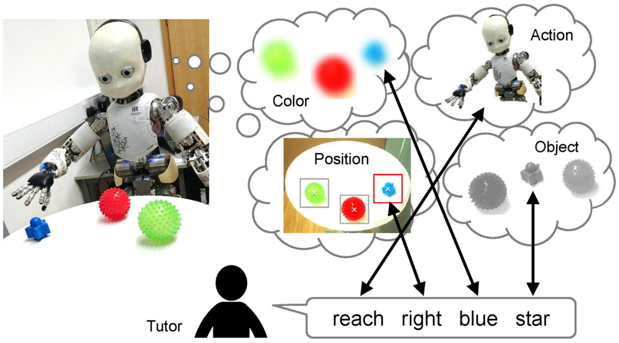


FIGURE 1 | Overview of the cross-situational learning scenario as the focus of this study; the robot obtains multimodal information from multiple sensory-channels in a situation and estimates the relationships between words and sensory-channels.

The second level (CSL-II) is the selection of the relationship between the specific word in the sentence and a sensory-channel in the multimodal information. In the first level, we assume joint attention. Tomasello and Farrar (1986) showed that the utterance referring to the object on which the child's attention was already focused is more effective in language acquisition. The above scenario enables the tutor to identify the object of attention, i.e., the object at which the robot is gazing. Furthermore, we assume that the robot considers the tutor to be speaking a sentence concerning the object of attention. This assumption of joint attention can avoid the problem of the selection of an object. The second level is the main problem in this study. Many previous studies on CSL-I have been reported (Frank et al., 2007, 2009; Fontanari et al., 2009; Morse et al., 2010); however, there are not the case for studies on CSL-II. The study discussed in this paper focused on solving the crucial problem of CSL-II.

In this study, we assume a two-level mutual exclusivity constraint (Markman and Wachtel, 1988) (MEC-I and II) regarding the selection of the sensory-channel. The first level (MEC-I) is the mutual exclusivity of sensory-channels with a word, i.e., one word is allocated to one category in one sensory-channel. The second level (MEC-II) is the mutual exclusivity between sensory-channels indicated by words, i.e., one word related to each sensory-channel is spoken only once in a sentence (or is not spoken). MEC-II is a stronger constraint than MEC-I. The proposed method can include both levels of mutual exclusivity.

3.2. Generative Model and Graphical Model

The generative model of the proposed method is defined as equations (1–10). **Figure 2** shows a graphical model representing the probabilistic dependencies between variables of the generative model. Basically, the categorization for each sensory-channel is based on the Gaussian mixture model (GMM). In this model, the probability distribution of words is represented by the categorical distribution. The categorization of words in sentences is similar to that of LDA. The latent variable of a word shares the latent variable of any one of the sensory-channels in GMMs, signifying that a word and a category in a particular sensory-channel are generated from the same latent variable.

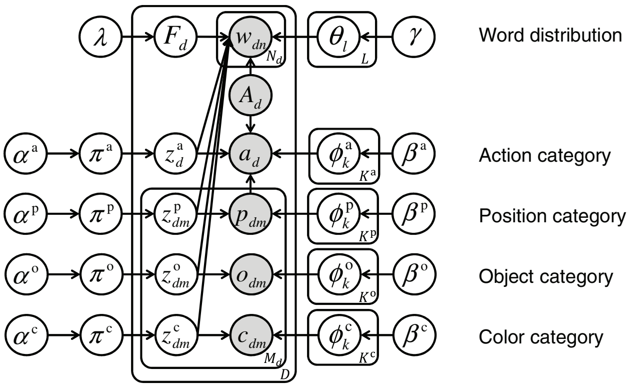


FIGURE 2 | Proposed graphical model for multichannel categorizations and for learning word meaning; the action, position, color, and object categories are represented by a component in Gaussian mixture models (GMMs). A word distribution is related to a category on GMMs. Gray nodes represent observed variables. Each variable is explained in the description of the generative model in Section 3.2.

We describe the generative model as follows:

$$F_d \sim \mathrm{Unif}(\lambda) \tag{1}$$

$$\theta_l \sim \mathrm{Dir}(\gamma) \tag{2}$$

$$\pi \sim \mathrm{GEM}(\alpha) \tag{3}$$

$$z_{dm} \sim \mathrm{Cat}(\pi) \tag{4}$$

$$w_{dn} \sim \mathrm{Cat}\left(\theta_{l=(F_{dn}, z^{F_{dn}}_{dA_d})}\right) \tag{5}$$

$$\phi_k \sim \mathrm{GIW}(\beta) \tag{6}$$

$$o_{dm} \sim \mathrm{Gauss}(\phi^{\mathrm{o}}_{z^{\mathrm{o}}_{dm}}) \tag{7}$$

$$c_{dm} \sim \mathrm{Gauss}(\phi^{\mathrm{c}}_{z^{\mathrm{c}}_{dm}}) \tag{8}$$

$$p_{dm} \sim \mathrm{Gauss}(\phi^{\mathrm{p}}_{z^{\mathrm{p}}_{dm}}) \tag{9}$$

$$a_d \sim \mathrm{Gauss}(\phi^{\mathrm{a}'}_{z^{\mathrm{a}}_{d}}), \tag{10}$$

where the discrete uniform distribution is denoted as Unif(·), the categorical distribution is denoted as Cat(·), the Dirichlet distribution is denoted as Dir(·), the stick-breaking process (SBP) (Sethuraman, 1994) is denoted as GEM(·), the Gaussian-inverse-Wishart distribution is denoted as GIW(·), and the multivariate Gaussian distribution is denoted as Gauss(·). See Murphy (2012) for specific formulas of the above probability distributions. In this paper, variables omitting superscript represent general notation, e.g., $\pi \in \{\pi\} = \{\pi^{\mathrm{a}},\ \pi^{\mathrm{p}},\ \pi^{\mathrm{o}}, \pi^{\mathrm{c}}\}$, and variables omitting subscripts represent collective notation, e.g., $F = \{F_1, F_2, \ldots F_D\}$. The number of trials is D. The number of objects on the table is M_d in the d-th trial. The number of words in the sentence is N_d in the d-th trial. The n-th word in the d-th trial is denoted as w_{dn}, which is represented by the bag-of-words (BoW). The model allows sentences containing zero to four words. The model associates the word distributions θ with categories z_{dm} on four sensory-channels, namely, the action a_d, the position p_{dm} of the object on the table, the object feature o_{dm}, and the object color c_{dm}. In this study, we define the action a_d as a static action feature, i.e., proprioceptive and tactile features, when the robot completes an action. An index of the object of attention selected by the robot from among the multiple objects on the table is denoted as $\mathrm{A}_d = m$. The sequence representing the respective sensory-channels associated with each word in the sentence is denoted as F_d, e.g., $F_d = (\mathrm{a}, \mathrm{p}, \mathrm{c}, \mathrm{o})$. The number of categories for each sensory-channel is K. An index of the word distribution is denoted as l. The set of all the word distributions is denoted as $\theta = \{\theta_{l=(F_{dn}, z^{F_{dn}}_{dm})} | F_{dn} \in \{\mathrm{o}, \mathrm{c}, \mathrm{p}, \mathrm{a}\}, z^{F_{dn}}_{dm} \in \{1, 2, \ldots, K^{F_{dn}}\}\}$. The index of the category of the sensory-channel F_{dn} and the object A_d is denoted as $z^{F_{dn}}_{dA_d}$. Then, the number of word distributions L is the sum of the number of categories of all the sensory-channels, i.e., $L = K^{\mathrm{a}} + K^{\mathrm{p}} + K^{\mathrm{o}} + K^{\mathrm{c}}$. The action category ϕ^{a}_{k}, the position category ϕ^{p}_{k}, the object category ϕ^{o}_{k}, and the color category ϕ^{c}_{k} are represented by a Gaussian distribution. The mean vector and the covariance matrix of the Gaussian distribution are denoted as $\phi_k = \{\mu_k, \Sigma_k\}$. We define $\phi^{\mathrm{a}'}_{z^{\mathrm{a}}_{d}}$ as the parameter of the Gaussian distribution that added the object position p_{dA_d} to the element of the mean vector representing the relative coordinates between the hand position and the target position. The position information of the object $\mathrm{A}_d = m$ is denoted as p_{dA_d}. Therefore, $\phi^{\mathrm{a}'}_{z^{\mathrm{a}}_{d}}$ is the parameter obtained by converting the target hand position to the absolute coordinate system based on $\phi^{\mathrm{a}}_{z^{\mathrm{a}}_{d}}$ (the parameter of the action category represented in the relative coordinate system) and p_{dA_d} (the position of the object of attention). The mixture weights of the categories for each sensory-channel are denoted as π^{a}, π^{p}, π^{o}, and π^{c}. The hyperparameter λ of the uniform distribution, i.e., equation (1), has the mutual exclusivity constraint that determines that each sensory-channel is represented only once in each sentence. The hyperparameter of the mixture weights π is denoted as α. The hyperparameter of the Gaussian-inverse-Wishart distribution is denoted as $\beta = \{m_0, \kappa_0, V_0, \nu_0\}$. The hyperparameter of the Dirichlet distribution is denoted as γ. Italic notation (a, p, o, c) represents observation variables, ordinary notation used as a superscript (a, p, o, c) represents sensory-channels.

The robot needs to estimate the number of categories based on experience because the robot cannot have previous knowledge about categories. The proposed method can learn an appropriate number of categories, depending on the collected data, by using a nonparametric Bayesian approach. Specifically, this method uses the SBP, a method based on the Dirichlet process. Therefore, this method can consider theoretically infinite numbers K^{a}, K^{p}, K^{o}, and K^{c}. In this paper, we approximate the values of parameters representing the number of categories K^{a}, K^{p}, K^{o}, and K^{c} by assigning sufficiently large values, i.e., a weak-limit approximation (Fox et al., 2011).

3.3. Learning Algorithm

This model estimates parameters representing multiple categories, word distribution, the relationships between the word and the sensory-channel as input for the object features, positions, colors, robot actions, and the sentences spoken by a tutor. The model parameters and latent variables of the proposed method are estimated from the following joint posterior distribution by Gibbs sampling:

$$\Theta, Z \sim p(\Theta, Z \,|\, X, H), \tag{11}$$

where the set of model parameters is denoted as $\Theta = \{\{\pi\}, \{\phi\}, \theta\}$, the set of latent variables is denoted as $\mathrm{Z} = \{\{z\}, F\}$, the set of observation variables is denoted as $X = \{a, p, o, c, w, A\}$, and the set of hyperparameters of the model is denoted as $H = \{\{\alpha\}, \{\beta\}, \lambda, \gamma\}$.

The learning algorithm is obtained by repeatedly sampling the conditional posterior distributions for each parameter. The Dirichlet and GIW distributions are conjugate prior distributions for the categorical and Gaussian distributions, respectively (Murphy, 2012). Therefore, the conditional posterior distributions can be determined analytically. **Algorithm 1** shows the pseudo-code for the learning procedure. The initial values of the model parameters can be set arbitrarily in accordance with a condition. The following is the conditional posterior distribution of each element used for performing Gibbs sampling.

A parameter π^{o} of categorical distribution representing the mixture weight of an object category is sampled as follows:

$$\pi^{\mathrm{o}} \sim p(\pi^{\mathrm{o}}|z^{\mathrm{o}}, \alpha^{\mathrm{o}}) \propto \prod_{d=1}^{D} \prod_{m=1}^{M_d} \mathrm{Cat}(z^{\mathrm{o}}_{dm}|\pi^{\mathrm{o}})\mathrm{Dir}(\pi^{\mathrm{o}}|\alpha^{\mathrm{o}}) \propto \mathrm{Dir}(\pi^{\mathrm{o}}|z^{\mathrm{o}}, \alpha^{\mathrm{o}}), \tag{12}$$

where z^{o} denotes the set of all the latent variables of an object category. A parameter π^{c} of categorical distribution representing the mixture weight of the color category is sampled as follows:

$$\pi^{\mathrm{c}} \sim p(\pi^{\mathrm{c}}|z^{\mathrm{c}}, \alpha^{\mathrm{c}}) \propto \prod_{d=1}^{D} \prod_{m=1}^{M_d} \mathrm{Cat}(z^{\mathrm{c}}_{dm}|\pi^{\mathrm{c}})\mathrm{Dir}(\pi^{\mathrm{c}}|\alpha^{\mathrm{c}}) \propto \mathrm{Dir}(\pi^{\mathrm{c}}|z^{\mathrm{c}}, \alpha^{\mathrm{c}}), \tag{13}$$

where z^{c} denotes a set of all the latent variables of the color category. A parameter π^{p} of the categorical distribution representing the mixture weight of the position category is sampled as follows:

$$\pi^{p} \sim p(\pi^{\mathrm{p}}|z^{\mathrm{p}}, \alpha^{\mathrm{p}}) \propto \prod_{d=1}^{D} \prod_{m=1}^{M_d} \mathrm{Cat}(z^{\mathrm{p}}_{dm}|\pi^{\mathrm{p}})\mathrm{Dir}(\pi^{\mathrm{p}}|\alpha^{\mathrm{p}}) \propto \mathrm{Dir}(\pi^{\mathrm{p}}|z^{\mathrm{p}}, \alpha^{\mathrm{p}}), \tag{14}$$

where z^{p} denotes the set of all the latent variables of the position category. A parameter π^{a} of the categorical distribution representing the mixture weight of the action category is sampled as follows:

$$\pi^{\mathrm{a}} \sim p(\pi^{\mathrm{a}}|z^{\mathrm{a}}, \alpha^{\mathrm{a}}) \propto \prod_{d=1}^{D}\mathrm{Cat}(z^{\mathrm{a}}_{d}|\pi^{\mathrm{a}})\mathrm{Dir}(\pi^{\mathrm{a}}|\alpha^{\mathrm{a}}) \propto \mathrm{Dir}(\pi^{\mathrm{a}}|z^{\mathrm{a}}, \alpha^{\mathrm{a}}), \tag{15}$$

where z^{a} denotes a set of all the latent variables of the action category. A parameter ϕ^{o}_{k} of the Gaussian distribution of the object category is sampled for each $k \in \{1, 2, \ldots, K^{\mathrm{o}}\}$ as follows:

$$\phi^{\mathrm{o}}_{k} \sim p(\phi^{\mathrm{o}}_{k}|z^{\mathrm{o}}, o, \beta^{\mathrm{o}}) \propto \prod_{d=1}^{D} \prod_{m=1}^{M_d} \mathrm{Gauss}(o_{dm}|\phi^{\mathrm{o}}_{k})\mathrm{GIW}(\phi^{\mathrm{o}}_{k}|\beta^{\mathrm{o}}) \propto \mathrm{GIW}(\phi^{\mathrm{o}}_{k}|o_k, \beta^{\mathrm{o}}), \tag{16}$$

where o_k denotes a set of all the object features of the object category $z^{\mathrm{o}}_{dm} = k$ in $m \in \{1, 2, \ldots, M_d\}$ and $d \in \{1, 2, \ldots, D\}$.

Algorithm 1 | Learning algorithm based on Gibbs sampling.

```
procedure Gibbs_Sampling ($a, p, o, c, w, A$)
  Setting of hyperparameters $\{\alpha\}, \{\beta\}, \lambda, \gamma$
  Initialization of parameters and latent variables $\{\pi\}, \{\phi\}, \theta, \{z\}, F$
  for $j = 1$ to iteration_number do
    $\pi^{\mathrm{o}} \sim \mathrm{Dir}(\pi^{\mathrm{o}} \mid z^{\mathrm{o}}, \alpha^{\mathrm{o}})$   // equation (12)
    $\pi^{\mathrm{c}} \sim \mathrm{Dir}(\pi^{\mathrm{c}} \mid z^{\mathrm{c}}, \alpha^{\mathrm{c}})$   // equation (13)
    $\pi^{\mathrm{p}} \sim \mathrm{Dir}(\pi^{\mathrm{p}} \mid z^{\mathrm{p}}, \alpha^{\mathrm{p}})$   // equation (14)
    $\pi^{\mathrm{a}} \sim \mathrm{Dir}(\pi^{\mathrm{a}} \mid z^{\mathrm{a}}, \alpha^{\mathrm{a}})$   // equation (15)
    for $k = 1$ to $K^{\mathrm{o}}$ do
      $\phi^{\mathrm{o}}_{k} \sim \mathrm{GIW}(\phi^{\mathrm{o}}_{k}|o_k, \beta^{\mathrm{o}})$   // equation (16)
    end for
    for $k = 1$ to $K^{\mathrm{c}}$ do
      $\phi^{\mathrm{c}}_{k} \sim \mathrm{GIW}(\phi^{\mathrm{c}}_{k}|c_k, \beta^{\mathrm{c}})$   // equation (17)
    end for
    for $k = 1$ to $K^{\mathrm{p}}$ do
      $\phi^{\mathrm{p}}_{k} \sim \mathrm{GIW}(\phi^{\mathrm{p}}_{k}|p_k, \beta^{\mathrm{p}})$   // equation (18)
    end for
    for $k = 1$ to $K^{\mathrm{a}}$ do
      $\phi^{\mathrm{a}}_{k} \sim \mathrm{GIW}(\phi^{\mathrm{a}}_{k}|a'_k, \beta^{\mathrm{a}})$   // equation (19)
    end for
    for $l = \left(F_{dn}, z^{F_{dn}}_{dA_d}\right)$ in $\left\{\left(F_{dn}, z^{F_{dn}}_{dm}\right) \mid F_{dn} \in \{\mathrm{o}, \mathrm{c}, \mathrm{p}, \mathrm{a}\},\ z^{F_{dn}}_{dm} \in \{1, 2, \ldots, K^{F_{dn}}\}\right\}$ do
      $\theta_l \sim \mathrm{Dir}(\theta_l \mid w_l, \gamma)$   // equation (20)
    end for
    for $d = 1$ to $D$ do
      for $m = 1$ to $M_d$ do
        $z^{\mathrm{o}}_{dm} \sim \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn}|\theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Gauss}\left(o_{dm}|\phi^{\mathrm{o}}_{z^{\mathrm{o}}_{dm}}\right) \mathrm{Cat}(z^{\mathrm{o}}_{dm}|\pi^{\mathrm{o}})$   // equation (21)
        $z^{\mathrm{c}}_{dm} \sim \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn}|\theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Gauss}\left(c_{dm}|\phi^{\mathrm{c}}_{z^{\mathrm{c}}_{dm}}\right) \mathrm{Cat}(z^{\mathrm{c}}_{dm}|\pi^{\mathrm{c}})$   // equation (22)
        $z^{\mathrm{p}}_{dm} \sim \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn}|\theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Gauss}\left(p_{dm}|\phi^{\mathrm{p}}_{z^{\mathrm{p}}_{dm}}\right) \mathrm{Cat}(z^{\mathrm{p}}_{dm}|\pi^{\mathrm{p}})$   // equation (23)
      end for
      $z^{\mathrm{a}}_{d} \sim \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn}|\theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Gauss}\left(a_d|\phi^{\mathrm{a}'}_{z^{\mathrm{a}}_{d}}\right) \mathrm{Cat}(z^{\mathrm{a}}_{d}|\pi^{\mathrm{a}})$   // equation (24)
      $F_d \sim \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn}|\theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Unif}(F_d|\lambda)$   // equation (25)
    end for
  end for
  return $\{\pi\}, \{\phi\}, \theta, \{z\}, F$
end procedure
```

A parameter ϕ^{c}_{k} of the Gaussian distribution of the color category is sampled for each $k \in \{1, 2, \ldots, K^{\mathrm{c}}\}$ as follows:

$$\phi^{\mathrm{c}}_{k} \sim p(\phi^{\mathrm{c}}_{k}|z^{\mathrm{c}}, c, \beta^{\mathrm{c}}) \propto \prod_{d=1}^{D} \prod_{m=1}^{M_d} \mathrm{Gauss}(c_{dm}|\phi^{\mathrm{c}}_{k})\mathrm{GIW}(\phi^{\mathrm{c}}_{k}|\beta^{\mathrm{c}}) \propto \mathrm{GIW}(\phi^{\mathrm{c}}_{k}|c_k, \beta^{\mathrm{c}}), \tag{17}$$

where c_k denotes the set of all the color features of the color category $z^{\mathrm{c}}_{dm} = k$ in $m \in \{1, 2, \dots, M_d\}$ and $d \in \{1, 2, \dots, D\}$. A parameter ϕ^{p}_k of the Gaussian distribution of the position category is sampled for each $k \in \{1, 2, \dots, K^{\mathrm{p}}\}$ as follows:

$$\phi^{\mathrm{p}}_k \sim p(\phi^{\mathrm{p}}_k | z^{\mathrm{p}}, p, \beta^{\mathrm{p}}) \propto \prod_{d=1}^{D} \prod_{m=1}^{M_d} \mathrm{Gauss}(p_{dm}|\phi^{\mathrm{p}}_k)\mathrm{GIW}(\phi^{\mathrm{p}}_k|\beta^{\mathrm{p}}) \\ \propto \mathrm{GIW}(\phi^{\mathrm{p}}_k | p_k, \beta^{\mathrm{p}}), \tag{18}$$

where p_k denotes the set of all the position information of the position category $z^{\mathrm{p}}_{dm} = k$ in $m \in \{1, 2, \dots, M_d\}$ and $d \in \{1, 2, \dots, D\}$. A parameter ϕ^{a}_k of the Gaussian distribution of the action category is sampled for each $k \in \{1, 2, \dots, K^{\mathrm{a}}\}$ as follows:

$$\phi^{\mathrm{a}}_k \sim p(\phi^{\mathrm{a}}_k | z^{\mathrm{a}}, a, p, A, \beta^{\mathrm{a}}) \propto \prod_{d=1}^{D} \mathrm{Gauss}(a'_d|\phi^{\mathrm{a}}_k)\mathrm{GIW}(\phi^{\mathrm{a}}_k|\beta^{\mathrm{a}}) \\ \propto \mathrm{GIW}(\phi^{\mathrm{a}}_k | a'_k, \beta^{\mathrm{a}}), \tag{19}$$

where a denotes the set of all the action information, p denotes the set of all the position information, and A denotes the set of all the attention information. The element representing the relative coordinates of the hand of ${a'}_d$ is calculated by the element representing the absolute coordinates of the hand of a, the object positions p, and the attention information A. The set of all the action information of the action category $z^{\mathrm{a}}_d = k$ in $d \in \{1, 2, \dots, D\}$ is denoted as a'_k. A parameter θ_l of the word probability distribution is sampled for each $l \in \{(F_{dn}, z^{F_{dn}}_{dm}) | F_{dn} \in \{\mathrm{o}, \mathrm{c}, \mathrm{p}, \mathrm{a}\}, z^{F_{dn}}_{dm} \in \{1, 2, \dots, K^{F_{dn}}\}\}$ as follows:

$$\theta_l \sim p(\theta_l | w, z^{\mathrm{o}}, z^{\mathrm{c}}, z^{\mathrm{p}}, z^{\mathrm{a}}, F, A, \gamma) \\ \propto \prod_{d=1}^{D} \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn} | \theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Dir}(\theta_l|\gamma) \\ \propto \mathrm{Dir}(\theta_l | w_l, \gamma) \tag{20}$$

where w denotes the set of all the words, F denotes the set of frames of all the sentences, and w_l denotes the set of all the words of the word category $l = (F_{dn}, z^{F_{dn}}_{dA_d})$ in $n \in \{1, 2, \dots, N_d\}$ and $d \in \{1, 2, \dots, D\}$. A latent variable z^{o}_{dm} of the object category is sampled for each $m \in \{1, 2, \dots, M_d\}$ and $d \in \{1, 2, \dots, D\}$ as follows:

$$z^{\mathrm{o}}_{dm} \sim p(z^{\mathrm{o}}_{dm} | w_d, z^{\mathrm{c}}_d, z^{\mathrm{p}}_d, z^{\mathrm{a}}_d, z^{\mathrm{o}}_{-dm}, \theta, F_d, A_d, o_{dm}, \phi^{\mathrm{o}}, \pi^{\mathrm{o}}) \\ \propto \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn} | \theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Gauss}(o_{dm}|\phi^{\mathrm{o}}_{z^{\mathrm{o}}_{dm}})\mathrm{Cat}(z^{\mathrm{o}}_{dm}|\pi^{\mathrm{o}}), \tag{21}$$

where w_d is a sequence of words in the d-th trial and z^{o}_{-dm} is the set of indicates of the object categories without z^{o}_{dm} in the d-th trial. A latent variable z^{c}_{dm} of the color category is sampled for each $m \in \{1, 2, \dots, M_d\}$ and $d \in \{1, 2, \dots, D\}$ as follows:

$$z^{\mathrm{c}}_{dm} \sim p(z^{\mathrm{c}}_{dm} | w_d, z^{\mathrm{o}}_d, z^{\mathrm{p}}_d, z^{\mathrm{a}}_d, z^{\mathrm{c}}_{-dm}, \theta, F_d, A_d, c_{dm}, \phi^{\mathrm{c}}, \pi^{\mathrm{c}}) \\ \propto \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn} | \theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Gauss}\left(c_{dm}|\phi^{\mathrm{c}}_{z^{\mathrm{c}}_{dm}}\right) \mathrm{Cat}(z^{\mathrm{c}}_{dm}|\pi^{\mathrm{c}}), \tag{22}$$

where z^{c}_{-dm} is the set of indicates of the object categories without z^{c}_{dm} in the d-th trial. A latent variable z^{p}_{dm} of the position category is sampled for each $m \in \{1, 2, \dots, M_d\}$ and $d \in \{1, 2, \dots, D\}$ as follows:

$$z^{\mathrm{p}}_{dm} \sim p(z^{\mathrm{p}}_{dm} | w_d, z^{\mathrm{o}}_d, z^{\mathrm{c}}_d, z^{\mathrm{a}}_d, z^{\mathrm{p}}_{-dm}, \theta, F_d, A_d, p_{dm}, \phi^{\mathrm{p}}, \pi^{\mathrm{p}}) \\ \propto \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn} | \theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Gauss}\left(p_{dm}|\phi^{\mathrm{p}}_{z^{\mathrm{p}}_{dm}}\right) \mathrm{Cat}(z^{\mathrm{p}}_{dm}|\pi^{\mathrm{p}}), \tag{23}$$

where z^{p}_{-dm} is the set of indicates of the object categories without z^{p}_{dm} in the d-th trial. A latent variable z^{a}_d of the action category is sampled for each $d \in \{1, 2, \dots, D\}$ as follows:

$$z^{\mathrm{a}}_d \sim p(z^{\mathrm{a}}_d | w_d, z^{\mathrm{o}}_d, z^{\mathrm{c}}_d, z^{\mathrm{p}}_d, \theta, F_d, A_d, a_d, p_d, \phi^{\mathrm{a}}, \pi^{\mathrm{a}}) \\ \propto \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn} | \theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Gauss}\left(a_d|\phi^{\mathrm{a}'}_{z^{\mathrm{a}}_d}\right) \mathrm{Cat}(z^{\mathrm{a}}_d|\pi^{\mathrm{a}}), \tag{24}$$

where p_d is the set of position data in the d-th trial. A latent variable F_d representing the sensory-channels of words in a sentence is sampled for each $d \in \{1, 2, \dots, D\}$ as follows:

$$F_d \sim p(F_d | w, z^{\mathrm{o}}, z^{\mathrm{c}}, z^{\mathrm{p}}, z^{\mathrm{a}}, \theta, A, \lambda) \\ \propto \prod_{n=1}^{N_d} \mathrm{Cat}\left(w_{dn} | \theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)}\right) \mathrm{Unif}(F_d|\lambda). \tag{25}$$

3.4. Action Generation and Attention Selection

In this section, we describe the approach that selects an action and an object of attention from the human spoken sentence. A robot capable of learning word meanings accurately is considered to be able to understand human instruction more accurately. In an action generation task, the robot performs an action a_d based on word meanings and multiple categories Θ from observed information w_d, o_d, c_d, and p_d. In this case, the robot can use the set of model parameters Θ learned by using Gibbs sampling in the CSL task. In the action generation task, we maximize the following equation:

$$\underset{a_d}{\mathrm{argmax}}\, p(a_d | w_d, o_d, c_d, p_d, \theta, \{\phi\}, \{\pi\}, \lambda) \\ = \underset{a_d}{\mathrm{argmax}} \sum_{A_d} \sum_{z^{\mathrm{a}}_d} p(a_d | \phi^{\mathrm{a}}, z^{\mathrm{a}}_d, p_d, A_d) \\ \times p(A_d, z^{\mathrm{a}}_d | w_d, o_d, c_d, p_d, \theta, \{\phi\}, \{\pi\}, \lambda). \tag{26}$$

In practice, this maximization problem is separated into two approximation processes, because it is difficult to maximize equation (26) directly.

(1) The first process is the maximization of the attention A_d and the index of the action category z^{a}_d

$$A^*_d, z^{\mathrm{a}*}_d = \underset{A_d, z^{\mathrm{a}}_d}{\mathrm{argmax}}\, p(A_d, z^{\mathrm{a}}_d | w_d, o_d, c_d, p_d, \theta, \{\phi\}, \{\pi\}, \lambda). \tag{27}$$

The probability distribution of equation (27) is represented by the following equation:

$$\begin{aligned}
&p(A_d, z^{\mathrm{a}}_d | w_d, o_d, c_d, p_d, \theta, \{\phi\}, \{\pi\}, \lambda) \\
&\propto p(A_d = m) p(z^{\mathrm{a}}_d | \pi^{\mathrm{a}}) \prod_{M_d} \sum_{z^{\mathrm{o}}_{dm}} \sum_{z^{\mathrm{c}}_{dm}} \sum_{z^{\mathrm{p}}_{dm}} \\
&\mathrm{Gauss}\left(o_{dm} | \phi^{\mathrm{o}}_{z^{\mathrm{o}}_{dm}} \right) \mathrm{Cat}(z^{\mathrm{o}}_{dm} | \pi^{\mathrm{o}}) \\
&\mathrm{Gauss}\left(c_{dm} | \phi^{\mathrm{c}}_{z^{\mathrm{c}}_{dm}} \right) \mathrm{Cat}(z^{\mathrm{c}}_{dm} | \pi^{\mathrm{c}}) \\
&\mathrm{Gauss}\left(p_{dm} | \phi^{\mathrm{p}}_{z^{\mathrm{p}}_{dm}} \right) \mathrm{Cat}(z^{\mathrm{p}}_{dm} | \pi^{\mathrm{p}}) \\
&\left[\sum_{F_d} \mathrm{Unif}(F_d | \lambda) \prod_{N_d} \mathrm{Cat}\left(w_{dn} | \theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)} \right) \right]. \qquad (28)
\end{aligned}$$

Then, we assumed $p(A_d = m) = 1/M_d$ as equal probability for the number of objects.

(2) The second process is the maximization of the action a_d using A^*_d and $z^{\mathrm{a}*}_d$

$$\begin{aligned}
a^*_d &= \operatorname*{argmax}_{a_d} p(a_d | \phi^{\mathrm{a}}, z^{\mathrm{a}*}_d, p_d, A^*_d) \\
&= \operatorname*{argmax}_{a_d} \mathrm{Gauss}\left(a_d | \phi^{\mathrm{a}'}_{z^{\mathrm{a}*}_d} \right) \\
&= \mu^{\mathrm{a}'}_{z^{\mathrm{a}*}_d}, \qquad (29)
\end{aligned}$$

where the mean vector of the Gaussian distribution of the action category $z^{\mathrm{a}*}_d$ is denoted as $\mu^{\mathrm{a}'}_{z^{\mathrm{a}*}_d}$.

3.5. Description of the Current Situation and Self-Action by the Robot

In this section, we describe the approach followed by the description task representing the current situation and the self-action of the robot. We consider a robot capable of learning word meanings accurately to be able to describe the current situation and self-action more accurately. In the action description task, the robot utters a sentence w_d regarding a self-action a_d and observed information o_{d}, c_d, and p_d based on word meanings and multiple categories Θ. In this case, the robot can use the set of model parameters Θ learned by using Gibbs sampling in the CSL task. In the action description task, we maximize the following equation:

$$\begin{aligned}
&\operatorname*{argmax}_{w_d} p(w_d | a_d, o_d, c_d, p_d, \theta, \{\phi\}, \{\pi\}, F_d, A_d) \\
&\propto \mathrm{argmax}_{w_d} \sum_{z^{\mathrm{a}}_d} \sum_{z^{\mathrm{o}}_{dA_d}} \sum_{z^{\mathrm{c}}_{dA_d}} \sum_{z^{\mathrm{p}}_{dA_d}} \\
&\mathrm{Gauss}\left(a_d | \phi^{\mathrm{a}'}_{z^{\mathrm{a}}_d} \right) \mathrm{Cat}(z^{\mathrm{a}}_d | \pi^{\mathrm{a}}), \\
&\mathrm{Gauss}\left(o_{dA_d} | \phi^{\mathrm{o}}_{z^{\mathrm{o}}_{dA_d}} \right) \mathrm{Cat}(z^{\mathrm{o}}_{dA_d} | \pi^{\mathrm{o}}) \\
&\mathrm{Gauss}\left(c_{dA_d} | \phi^{\mathrm{c}}_{z^{\mathrm{c}}_{dA_d}} \right) \mathrm{Cat}(z^{\mathrm{c}}_{dA_d} | \pi^{\mathrm{c}}) \\
&\mathrm{Gauss}\left(p_{dA_d} | \phi^{\mathrm{p}}_{z^{\mathrm{p}}_{dA_d}} \right) \mathrm{Cat}(z^{\mathrm{p}}_{dA_d} | \pi^{\mathrm{p}}) \\
&\prod_{N_d} \mathrm{Cat}\left(w_{dn} | \theta_{l=\left(F_{dn}, z^{F_{dn}}_{dA_d}\right)} \right). \qquad (30)
\end{aligned}$$

If the frame of the sentence is decided, e.g., $F_d = (\mathrm{a}, \mathrm{p}, \mathrm{c}, \mathrm{o})$, equation (30) is represented as the following:

$$\begin{aligned}
\text{Equation (30)} = \prod_{N_d} \operatorname*{argmax}_{w_{dn}} \sum_{z^{F_{dn}}_{dA_d}} &\mathrm{Gauss}\left(x^{F_{dn}}_{dA_d} | \phi^{F_{dn}}_{z^{F_{dn}}_{dA_d}} \right) \\
&\times \mathrm{Cat}\left(z^{F_{dn}}_{dA_d} | \pi^{F_{dn}} \right) \mathrm{Cat}\left(w_{dn} | \theta_{l=(F_{dn}, z^{F_{dn}}_{dA_d})} \right), \qquad (31)
\end{aligned}$$

where $x^{F_{dn}}_{dA_d}$ denotes data of the sensory-channel F_{dn} in the object number A_d, i,e., a_d, p_{dA_d}, c_{dA_d}, or o_{dA_d}. Therefore, equation (30) can be divided into the equations of finding a maximum value for each word.

4. EXPERIMENT I: SIMULATION ENVIRONMENT

We performed the experiments described in this section using the iCub simulator (Tikhanoff et al., 2008). In Section 4.1, we describe the difference in the conditions of the methods that are used for comparison purposes. In Section 4.2, we describe the CSL experiment. In Section 4.3, we describe the experiment involving the action generation task. In Section 4.4, we describe the experiment relating to the action description task.

4.1. Comparison Methods

We evaluated our proposed method by comparing its performance with that of two other methods.

(A) The proposed method.

This method has a mutual exclusivity constraint between the word and the sensory-channel (MEC-I and II), determining that each sensory-channel occurs only once in each sentence. For example, if the number of words in a sentence is $N_d = 4$, F_d can become a sequence such as (a, c, p, o), (a, p, c, o), or (p, c, o, a). Possible values of F_d are constrained by λ as a permutation of four sensory-channels. The number of permutations is ${}_4\mathrm{P}_{N_d} = 4!/(4 - N_d)!$.

(B) The proposed method without the mutual exclusivity constraint (w/o MEC-II).

This method does not have the mutual exclusivity constraint (MEC-II). This means that several words in a sentence may relate to the same sensory-channel. For example, if the number of words in a sentence is $N_d = 4$, F_d can become a sequence such as (a, o, c, o), (a, p, p, o), or (o, o, o, o) in addition to the above example of (A). Possible values of F_d are constrained by λ as a repeated permutation of four sensory-channels. The number of repeated permutations is ${}_4\Pi_{N_d} = 4^{N_d}$. In this case, the robot needs to consider additional pairs of relationships between the sensory-channel and word compared to method (A).

(C) The multilayered multimodal latent Dirichlet allocation (mMLDA) (Attamimi et al., 2016).

This method is based on mMLDA. In this research, this method was modified from the actual mMLDA to apply to our task and the proposed method. In particular, the emission probability for each sensory-channel is changed from

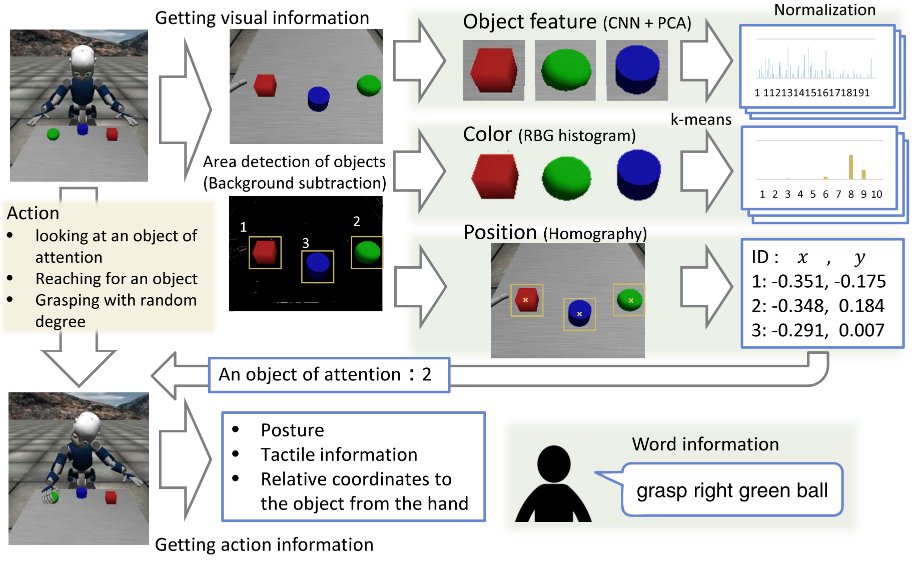


FIGURE 3 | Procedure for obtaining and processing data.

a categorical distribution to a Gaussian distribution. This means the multimodal categorization methods are based on a Gaussian distribution for each sensory-channel and a categorical distribution for word information. This method relates all observed words in a situation to all observed sensory-channel information in the situation. This method neither has the mutual exclusivity constraint (MEC-I and II) nor does it select the sensory-channel by words, i.e., F_d is not estimated.

4.2. Cross-Situational Learning

4.2.1. Experimental Procedure and Conditions

We conducted an experiment to learn the categories for each sensory-channel and the words associated with each category. **Figure 3** shows the procedure for obtaining and processing data. We describe the experimental procedure for CSL as follows:

1. The robot takes the initial position and posture. Some objects are placed on the table.
2. The robot acquires a visual image of the table. Subsequently, the robot detects object areas by using background subtraction. The detected object areas are cut out as object images of 64 × 64 pixels. The robot obtains the number of objects on the table.
3. The robot extracts object features, color features, and object positions. We used the deep learning framework Caffe (Jia et al., 2014) for convolutional neural networks (CNNs) (Krizhevsky et al., 2012) as an object feature extractor. We used a pre-trained CNN, i.e., CaffeNet trained by using ImageNet Large Scale Visual Recognition Challenge 2012 as the dataset. The object features are obtained from the fully connected FC6 layer (4096-dimensions) in CaffeNet. After that, the object features are reduced by principal component analysis (PCA). In terms of color features, the RGB histogram is vector quantized by k-means and normalized. The position data are converted into the world coordinate by homography. The position data are two-dimensional to represent the plane of the table.
4. The robot performs an action including a little randomness to an object of attention. The difference and uncertainty in the robot's action are represented by this randomness. First, the robot moves its eye-gaze to an object of attention. The object is selected randomly. Next, the robot moves its right hand to the coordinates of the target object by using inverse kinematics. A little random noise is added to a target position of the end-effector of the right hand. In many cases, the robot moves its right hand after looking at the object. The robot rarely refrains from moving its hands, looks at the object, which means the action of "look-at." When the hand approaches the position of the target object, the robot bends its fingers. The rate at which it bends its fingers is selected randomly. The five fingers move in synchronization. After the action is completed, the robot acquires the data relating to this action, including data relating to the posture, tactile data, and the relative coordinates of the object from its right hand. The action data are 38-dimensional and include the position of the right hand relative to the object (3-dim.), the rate at which the finger bends (1-dim.), the joint angles of the head, right_arm, and torso (6, 16, 3-dim.), and tactile information of the right hand (9-dim.). The action data is normalized to [0,1] for each dimension.
5. When the robot completes an action, the human tutor speaks a sentence about the object of attention and the action of the robot. The sentence contains the word related to each sensory-channel once, e.g., "touch left red box." In this task, the number of words in the sentence is indicated by a number ranging from zero to four. Zero means that the tutor did not speak a sentence.

The above process is carried out many times in different situations. The robot learns multiple categories and word meanings by using multimodal data observed in many trials.

The number of trials was $D = 20$ and 40 for CSL. The number of objects M_d on the table for each trial was a number from one to three. The number of words N_d in the sentence was a number from zero to four. We assume that a word related to each sensory-channel is spoken only once in each sentence. The word order in the sentences was changed. This experiment used 14 kinds of words: "reach," "touch," "grasp," "look-at," "front," "left," "right," "far," "green," "red," "blue," "box," "cup," and "ball." The upper limit number of the categories for each sensory-channel was $\mathrm{K} = 10$, i.e., the number of word distributions was $L = 40$. The number of iterative cycles used for Gibbs sampling was 200. The hyperparameters were $\alpha = 1.0$, $\gamma = 0.1$, $m_0 = \mathbf{O}_{x_{dim}}$, $\kappa_0 = 0.001$, $V_0 = \mathrm{diag}(0.01,\ 0.01)$, and $\nu 0 = x_{dim} + 2$, where the number of dimensions for each sensory-channel x is denoted as x_{dim} and the zero vector in x_{dim} dimensions is denoted as $\mathbf{O}_{x_{dim}}$. PCA is used to reduce the object features to 30 dimensions. The color features are quantized to 10 dimensions by k-means.

We describe the criteria of words uttered for action category as follows: "reach" corresponds to the robot extending its right hand toward an object and the robot's finger does not make contact with an object; "touch" corresponds to the robot touching an object and its finger is relatively opened; "grasp" corresponds to the robot's hand holding firmly an object; "look-at" corresponds to the robot not moving its right hand and it focuses on an object of attention only. Based on these criteria, the tutor determines an action word. In particular, "reach" and "touch" are similar; the only difference is whether the hand touches the object or not.

We evaluate the estimation accuracy of the learning results by using uncertain teaching sentences. Each sentence contains four words or fewer in different order. We compare the accuracy of three methods by reducing the word information. In addition, the number of learning trials is changed. We compared the accuracy by changing the number of trials. We evaluated the methods according to the following metrics.

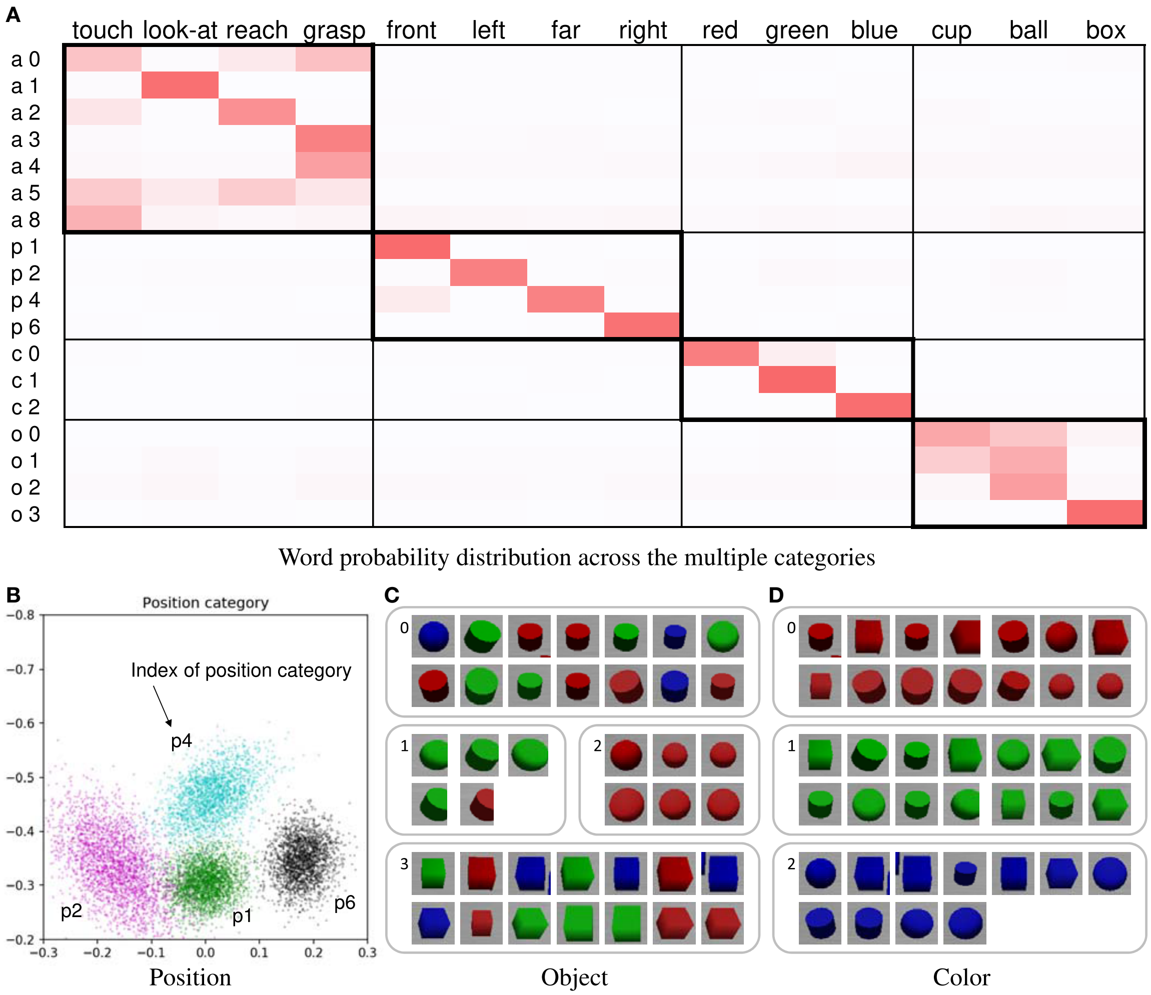
- Adjusted Rand index (ARI)
  We compare the matching rate between the estimated latent variables z for each sensory-channel and the true categorization results. The evaluation of this experiment uses the ARI (Hubert and Arabie, 1985), which is a measure of the degree of similarity between two clustering results.
- Estimation accuracy rate of F_d (EAR)
  The evaluation of the estimation results of the sensory-channels corresponding to the words are determined as follows:

$$\mathrm{EAR} = 1 - \frac{\text{The number of estimation errors}}{\text{The number of all of uttered words}}. \qquad (32)$$

4.2.2. Learning Results and Evaluation

The learning results obtained by using the proposed method are presented here. Forty trials were used. In this case, the number of words was four in all utterance sentences. **Figure 4A** shows the word probability distributions θ. Higher probability values are represented by darker shades. If the relationship between the word and sensory-channel can be estimated correctly, the ranges within thick-bordered boxes show higher probabilities. For example, the action categories show higher probabilities for words of action ("touch," "look-at," "reach," and "grasp"). The categories of the other sensory-channels are also the same. In the position and color categories, the estimated number of categories was equal to the number of types of words representing the sensory-channel. In the action category, the words "touch," "reach," and "grasp" were associated across several categories. In addition, these words were confused with each other. We considered actions representing these words to be ambiguous and similar. On the other hand, we considered the reason why these actions were divided into several categories to be a change in posture information depending on the position of the target object. **Figure 4B** shows the learning result for the position category ϕ^p. For example, the position category p1 is associated with the word "front" (see **Figures 4A,B**). **Figures 4C,D** show examples of the categorization results for the object and color categories. The object categorization result was not perfect. We considered the robot to find it difficult to clearly distinguish objects of different shapes because the 3D-models of the objects had simple shapes. The color categorization result was perfect. In this case, F_d was correctly estimated in all of the trials. The results demonstrate that the proposed method was able to accurately associate each word with its respective sensory-channel.

We performed the learning scenarios 10 times for each method. **Tables 1A,B** show the evaluation values of the experimental results for 20 and 40 trials. The rate of omitted words (ROW), which is expressed as a percentage, represents the uncertainty of teaching sentences. For example, the total number of words is 80 when ROW is 0%, 64 words for 20%, 48 words for 40%, and 32 words for 60% in 20 trials. Also, the total number of words is 160 for a ROW value of 0% and 96 words for 40% in 40 trials. ARI_a, ARI_p, ARI_o, and ARI_c are the ARI values of the action, position, object, and color category, respectively. The EAR values of mMLDA were not calculated because this method does not have F_d. If the ROW value is 100 (no word), the three methods will be equivalent as ALL, i.e., the GMM for each sensory-channel. We described the ARI values of ALL as reference values because ALL is not CSL. The EAR value obtained for the proposed method was higher than that obtained for the other methods. When the ROW decreased, i.e., the word information increased, the evaluation values tended to increase. Particularly, the result for the position category was favorably affected by the increase in word information for categorization. In addition, when the number of trials increased, the evaluation values tended to increase. This result suggests that the robot is able to learn the word meanings more effectively by accumulating more experience even in more complicated and uncertain situations. When the number of words was small (i.e., the ROW value is 40 or 60%), the difference between the EAR values of methods (A) and (B) was small (approximately 0.02) in 20 trials. However, when the number of words was large, the difference between the EAR values of methods (A) and (B) increased, and the EAR value of the method (A) was larger than that of (B). As a result, when the number of words was small, e.g., sentences including one or two words, there was almost no influence of the presence or absence of the MEC-II because the number of possible values of F_d of the methods (A) and (B) were close. On the other hand, when the number of words was large, e.g., sentences including four words, the MEC-II worked well because the number of possible values of F_d of the method (A) was narrowed properly down.



Word probability distribution across the multiple categories

Position

Object

Color

FIGURE 4 | **(A)** Word probability distribution across the multiple categories; darker shades represent higher probability values. The pair consisting of a letter and a number on the left of the table is the index of the word distribution, which represents the sensory-channel related to the word distribution and the index of the category. Note that category indices are not shown; they are merged and not used because the number of the categories is automatically estimated by the nonparametric Bayesian method. **(B)** Learning result of the position category; for example, the index of position category p1 corresponds to the word "front." The point group of each color represents each Gaussian distribution of the position category. The crosses in the different colors represent the object positions of the learning data. Each color represents a position category. **(C)** Example of categorization results of object category; **(D)** example of categorization results of color category.

4.3. Action Generation Task

4.3.1. Experimental Procedure and Conditions

In this experiment, the robot generates the action regarding the sentence spoken by the human tutor. The robot uses the learning results of the CSL task in Section 4.2. The robot selects the object of attention from among the objects on the table. In addition, the robot performs the action on the object of attention. In this task, the robot cannot use joint attention. Therefore, the robot needs to overcome both the problems of CSL-I and II. We describe the process of action generation as follows:

1. The robot takes the initial position and posture. Some objects are placed on the table.
2. The tutor speaks a sentence about an action that should be performed by the robot. The robot recognizes the tutor's spoken sentence.
3. The robot detects the objects on the table. The robot obtains object, color features, and position data by the same process as in Section 4.2.1 (step 3).
4. The robot selects the action category and the object of the attention by using equation (26). The robot calculates the target position by using equation (29).
5. The robot directs its eye-gaze to the object of attention, and the robot performs an action on the object of attention.

The above process is carried out many times on different sentences.

We compare the three methods by quantitative evaluation on the action generation task. We evaluate the accuracy of the selection of the object of attention. In addition, we evaluate the accuracy of an action of the robot based on questionnaire evaluation by participants. The robot generates an action from the tutor's spoken sentence in a situation. Participants check videos of the action generated by the robot and select a word representing the robot's action. We calculate the word accuracy rate (WAR) of the words selected by participants and the true words spoken by the tutor. In addition, we calculate the object accuracy rate (OAR) representing the rate at which the robot correctly selected the object instructed by the tutor.

We performed action generation tasks for a total of 12 different test-sentences. The test-sentences included four words representing the four sensory-channels. This placement of objects on the table was not used during the learning trials. In addition, the word order of sentences uttered during the action generation task is different from the word order of sentences uttered during the CSL task. The eight participants checked 36 videos of the robot's actions.

4.3.2. Results

Figure 5 shows three examples of the action generation results of the proposed method. **Figure 5A** shows the result of action generation by the robot in response to the sentence "reach front blue cup." **Figure 5B** shows the result of action generation by the robot in response to the sentence "grasp right green ball." **Figure 5C** shows the result of action generation by the robot in response to the sentence "touch left red box." **Table 2** shows the results of the quantitative evaluation of the action generation task. The proposed method enabled the robot to accurately select the object. As a result of the proposed method, the object indicated and the object selected by the robot coincided in all sentences. In addition, the proposed method showed the highest values for both WAR and OAR. Therefore, the robot could select an appropriate object and could perform an action even in situations and for sentences not used for CSL.

4.4. Action Description Task

4.4.1. Experimental Procedure and Conditions

In HRI, the ability of the robot to use the acquired word meanings for a description of the current situation is important. In this experiment, the robot performs an action and speaks the sentence

TABLE 1 | Experimental results of the CSL task for 20 and 40 trials.

Method	ROW	ARI_a	ARI_p	ARI_o	ARI_c	EAR_F_d
(A) 20 trials						
Proposed	0	0.300	0.606	0.408	0.782	**0.970**
w/o MEC-II	0	0.317	0.648	0.338	0.805	**0.759**
mMLDA	0	0.316	0.428	0.277	0.756	–
Proposed	20	0.290	0.564	0.332	0.762	0.727
w/o MEC-II	20	0.342	0.486	0.436	0.755	0.598
mMLDA	20	0.267	0.494	0.369	0.776	–
Proposed	40	0.324	0.493	0.354	0.780	0.556
w/o MEC-II	40	0.318	0.486	0.347	0.812	0.529
mMLDA	40	0.356	0.479	0.312	0.771	–
Proposed	60	0.282	0.460	0.295	0.783	0.381
w/o MEC-II	60	0.311	0.454	0.326	0.750	0.406
mMLDA	60	0.294	0.487	0.403	0.724	–
ALL	100 (no word)	0.325	0.431	0.346	0.751	–
(B) 40 trials						
Proposed	0	0.375	0.540	0.366	0.870	**0.989**
w/o MEC-II	0	0.383	0.524	0.333	0.805	0.834
mMLDA	0	0.388	0.594	0.377	0.822	–
Proposed	40	0.368	0.543	0.313	0.835	**0.867**
w/o MEC-II	40	0.417	0.577	0.320	0.842	0.780
mMLDA	40	0.340	0.600	0.377	0.856	–

Bold and underscore indicate the highest evaluation values, and bold indicates the second highest evaluation values.

TABLE 2 | Results of evaluation values for the action generation using the results of the CSL for 40 trials (ROW is 0%).

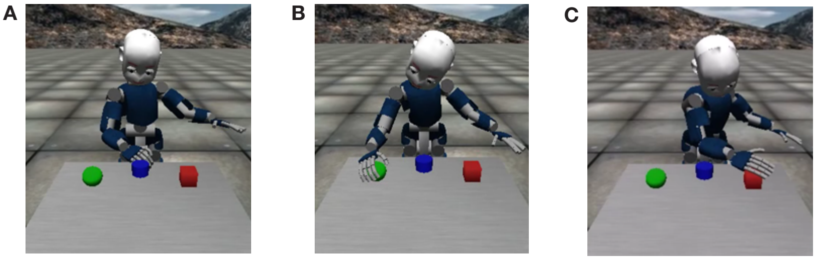
Method	WAR	OAR
Proposed	**0.604**	**1.000**
w/o MEC-II	**0.510**	**0.917**
mMLDA	0.260	0.667

Bold and underscore indicate the highest evaluation values, and bold indicates the second highest evaluation values.

TABLE 3 | Experimental results of action description task for 20 and 40 trials.

Method	Trials	ROW	F1	ACC
Proposed	20	0	**0.586**	**0.660**
w/o MEC-II	20	0	**0.534**	**0.613**
mMLDA	20	0	0.401	0.469
Proposed	20	40	**0.388**	**0.425**
w/o MEC-II	20	40	**0.343**	**0.369**
mMLDA	20	40	0.319	0.352
Proposed	40	0	**0.663**	**0.692**
w/o MEC-II	40	0	**0.642**	**0.671**
mMLDA	40	0	0.474	0.560
Proposed	40	40	**0.588**	**0.623**
w/o MEC-II	40	40	**0.548**	**0.606**
mMLDA	40	40	0.479	0.569

Bold and underscore indicate the highest evaluation values, and bold indicates the second highest evaluation values.



"reach front blue cup." "grasp right green ball." "touch left red box."

FIGURE 5 | Example of results of the action generation task in the iCub simulator. **(A)** Reach front blue cup. **(B)** Grasp right green ball. **(C)** Touch left red box.

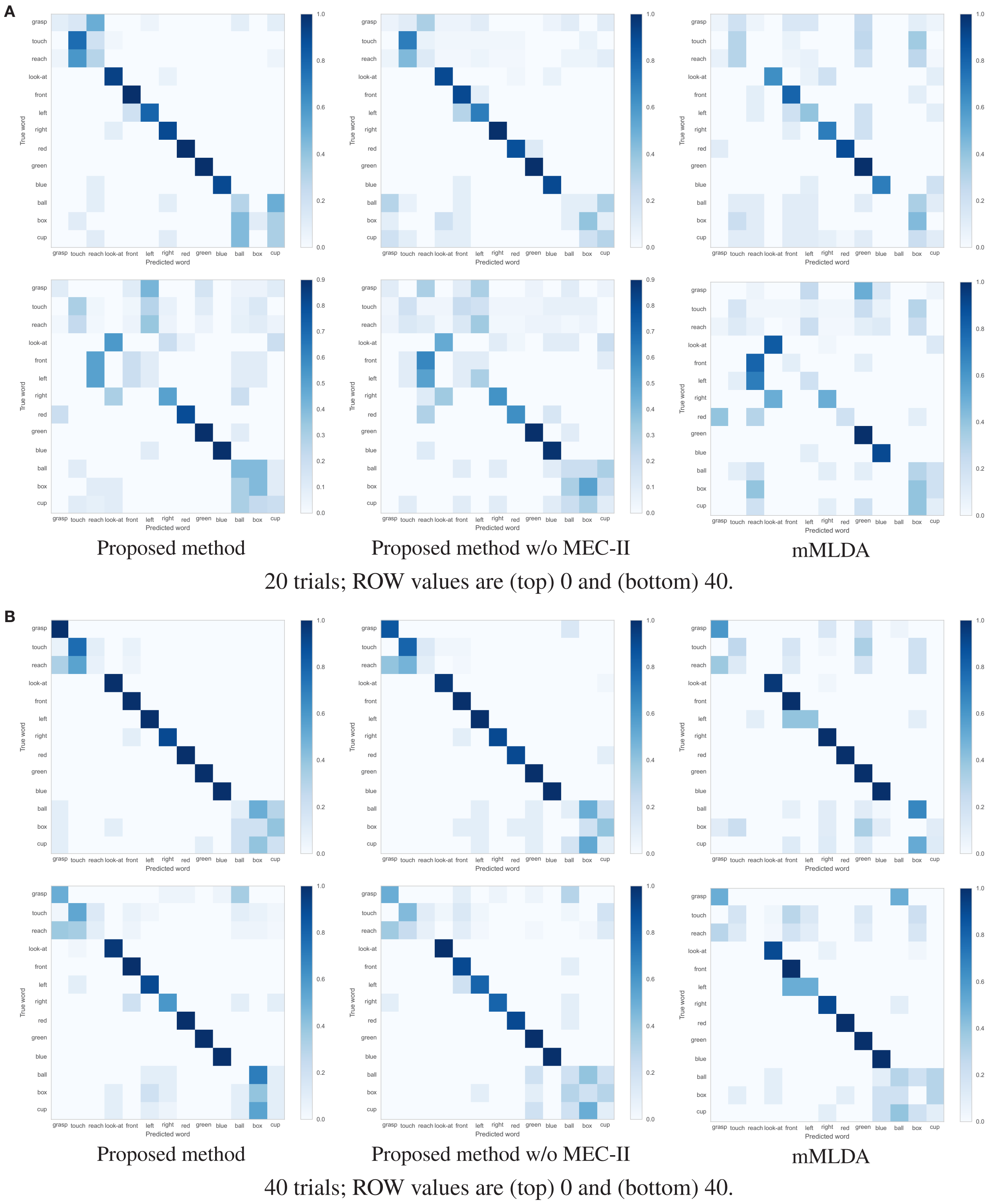


FIGURE 6 | Confusion matrix of results of the action description task using the learning result for 20 and 40 trials. **(A)** 20 trials; ROW values are (top) 0 and (bottom) 40. **(B)** 40 trials; ROW values are (top) 0 and (bottom) 40.

corresponding to this action. In other words, the robot explains self-action by using a sentence. The robot uses the learning results of the CSL task in Section 4.2. We describe the process of action description as follows:

1. The robot takes the initial position and posture. Some objects are placed on the table.
2. The robot detects the objects on the table. The robot obtains the object, color features, and position data by the same process as in Section 4.2.1 (step 3).
3. The robot selects the object of attention randomly. The robot directs its eye-gaze to the object of attention, and the robot performs an action on the object of attention by using the same process as in Section 4.2.1 (step 4).
4. When the robot completes an action, it utters a sentence about this action.

The above process is carried out many times on different actions. We performed action description tasks for a total of 12 actions. This placement of objects on the table was not used during the learning trials. The robot generates a sentence consisting of four words that include the four sensory-channels. The word order in the sentence is fixed as $F_d = (\mathrm{a, p, c, o})$.

We compare the three methods by quantitative evaluation of the action description task. We evaluate the F1-measure and the accuracy (ACC) between the sentence generated by the robot and the correct sentence decided by the tutor. The evaluation values are calculated by generating the confusion matrix between the predicted words and true words.

4.4.2. Results

Table 3 show the F1-measure and ACC values of the action description task using the learning results under the different conditions. The proposed method showed the highest evaluation values. **Figures 6A,B** shows the confusion matrices of the results of the action description task using the learning result for 20 and 40 training trials. Overall, the robot confused the words "reach" and "touch" similar to the learning result in **Figure 4A**. The robot had difficulties in distinguishing between "reach" and "touch." In other words, this result suggests that these words were learned as synonyms. When the ROW increased, the evaluation values decreased. For the ROW value of 40% obtained for 20 trials, the robot confused words related to the action and position categories. This could be explained by considering that the robot misunderstood the correspondence between the word and the sensory-channel because the word information was insufficient and uncertain during CSL with the ROW value of 40% and 20 trials. On the other hand, an increase in the number of learning trials resulted in an increase in the evaluation values. Even if the robot is exposed to uncertain utterances, the robot can explain self-action more accurately by gaining more experience. As a result, the robot could acquire the ability to explain self-action by CSL based on the proposed method.

5. EXPERIMENT II: REAL iCub ENVIRONMENT

In this section, we describe the experiment that was conducted by using the real iCub robot. The real-world environment involves more complexity than the simulation environment. We demonstrate that results similar to those of the simulator experiment can be obtained even in a more complicated real environment. We compare three methods, as in Section 4.1. In Section 5.1, we describe the experiment to assess cross-situational learning. In Section 5.2, we describe the experiment relating to the action generation task. In Section 5.3, we describe the experiment relating to the action description task.

5.1. Cross-Situational Learning

5.1.1. Conditions

The experimental procedure is the same as in Section 4.2.1. We use ARI and EAR as evaluation values. **Figure 7** shows all of the objects that were used in the real environment. We used 14 different objects including four types (car, cup, ball, and star) and four colors (red, green, blue, and yellow). In the simulation environment, the same type objects had the same shapes. In the real environment, objects of the same type include different shapes. In particular, all the car objects have different shapes, the cup objects have different sizes, and the star objects include one different shape. This experiment used 16 kinds of words: "reach," "touch," "grasp," "look-at," "front," "left," "right," "far," "green," "red," "blue," "yellow," "car," "cup," "ball," and "star." The number of trials was $D = 25$ and 40 for CSL. The number of objects M_d on the table for each trial was a number ranging from one to three. The number of words N_d in the sentence was a number ranging from zero to four. We assume that a word related to each sensory-channel is spoken only once in each sentence. The word order in the sentences was changed. Object features are reduced to 65 dimensions by PCA. Color features are quantized to 10 dimensions by k-means. The upper limit number of the categories for each sensory-channel was $K = 10$, i.e., the upper limit for the number of word distributions was $L = 40$. The hyperparameters were $\alpha = 1.0$, $\gamma = 0.1$, $m_0 = \mathbf{O}_{x_{dim}}$, $\kappa_0 = 0.001$, $V_0 = \mathrm{diag}(0.01, 0.01)$, and $\nu_0 = x_{dim} + 2$. The number of iterative cycles used for Gibbs sampling was 200.

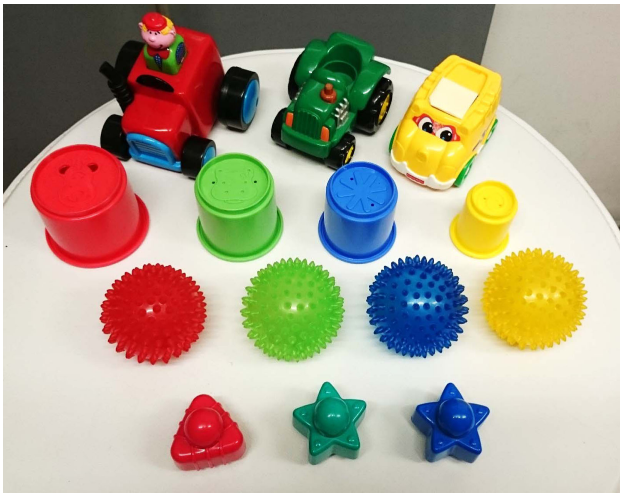


FIGURE 7 | All of the objects used in the real experiments (14 objects including four types and four colors).

5.1.2. Learning Results and Evaluation

The example we describe is the learning result of 25 trials and for a ROW value of 9%. In this case, the number of categories was set to $K = 5$. **Figure 8A** shows the word distributions θ. In the action category, the robot confused the words "reach" and "touch" as is the case with the simulator experiment. **Figure 8B** shows the learning result of the position category on the table. **Figure 8C** shows categorization results of objects. Although the object categorization contained a few mistakes, the results were mostly correct. **Figure 8D** shows the categorization results obtained for the

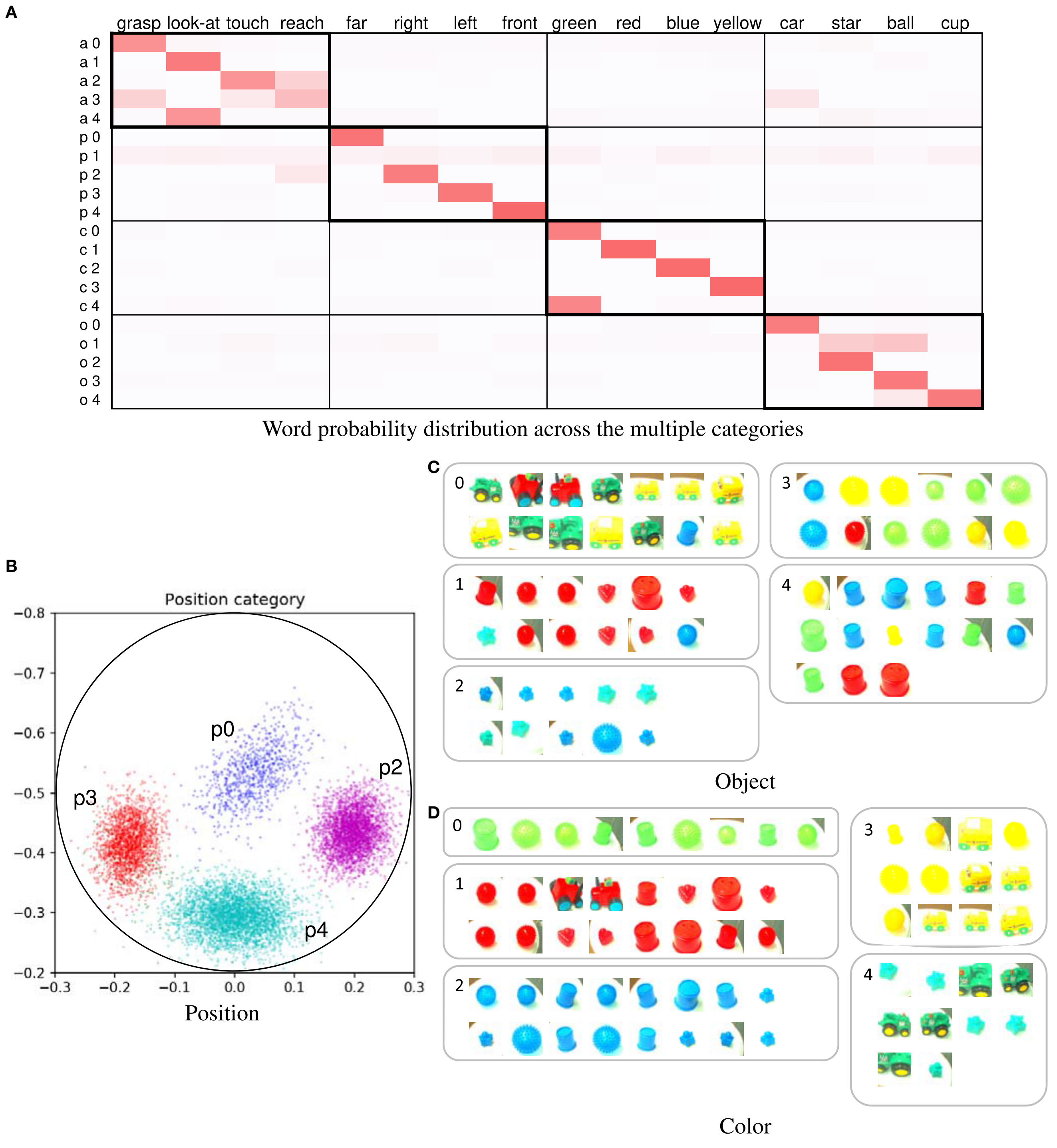


FIGURE 8 | **(A)** Word probability distribution across the multiple categories; **(B)** learning result of position category; each color of the point group represents each of the Gaussian distributions of the position category. The crosses of each color represent the object positions of the learning data. Each color represents a position category. The circle represents the area of the white circular table. **(C)** Example of categorization results of object category; **(D)** example of categorization results of color category.

color categorization, which was successful. Interestingly, two categories corresponding to the word "green" were created because the robot distinguished between bright green and dark green. In addition, the robot was able to learn that both of these categories related to the word "green."

Table 4 shows the evaluation values of the experimental results for 25 and 40 trials. There was not much difference in ARI values between the methods and between different conditions of ROW values. The EAR values of the proposed method were higher than those of the other methods. An increase in the number of trials led to an increase in the evaluation values, similar to the simulation results.

5.2. Action Generation Task

5.2.1. Conditions

In this experiment, the robot generates the action corresponding to the sentence spoken by the human tutor. The robot uses the learning results of the CSL task in Section 5.1. The experimental procedure is the same as in Section 4.3.1. We evaluated accuracy of object selection (the OAR values) using the CSL results for 25 trials. We performed the action generation task for a total of 12 different test-sentences, each of which comprised four words representing the four sensory-channels. The placement of objects on the table was different in each trial and differed from the placements that were used during the learning trials.

TABLE 4 | Experimental results of the CSL task for 25 and 40 trials.

Method	ROW	ARI_a	ARI_p	ARI_o	ARI_c	EAR_F_d
(A) 25 trials						
Proposed	0	0.239	0.932	0.201	0.720	**0.866**
w/o MEC-II	0	0.299	0.971	0.207	0.717	0.723
mMLDA	0	0.255	0.959	0.226	0.703	–
Proposed	30	0.297	0.879	0.227	0.702	**0.751**
w/o MEC-II	30	0.242	0.980	0.218	0.683	0.601
mMLDA	30	0.296	0.893	0.256	0.730	–
Proposed	50	0.240	0.905	0.257	0.681	0.604
w/o MEC-II	50	0.224	0.895	0.211	0.694	0.482
mMLDA	50	0.221	0.981	0.303	0.688	–
(B) 40 trials						
Proposed	0	0.304	0.960	0.240	0.691	**0.988**
w/o MEC-II	0	0.282	0.986	0.190	0.729	0.763
mMLDA	0	0.290	0.959	0.224	0.736	–
Proposed	30	0.303	0.978	0.193	0.698	**0.829**
w/o MEC-II	30	0.349	0.917	0.219	0.726	0.718
mMLDA	30	0.307	0.956	0.159	0.717	–
Proposed	50	0.316	0.922	0.199	0.718	0.668
w/o MEC-II	50	0.258	0.937	0.210	0.726	0.639
mMLDA	50	0.297	0.989	0.123	0.687	–

Bold and underscore indicate the highest evaluation values, and bold indicates the second highest evaluation values.

5.2.2. Results

Figure 9 shows an example of the results of the action generation task. **Figure 9A** shows the result of action generation by the robot for the sentence "grasp front red ball." **Figure 9B** shows the result of action generation by the robot for the sentence "reach right red cup." **Figure 9C** shows the result of action generation by the robot for the sentence "look-at left yellow cup." The resulting OAR values of the proposed method and its w/o MEC-II were 1.000, and the OAR value of mMLDA was 0.833. As a result, the robot could select an appropriate object even in situations and sentences not used at the CSL.

5.3. Action Description Task

5.3.1. Conditions

In this experiment, the robot performs the action and speaks the sentence regarding this action. The robot uses the learning results of the CSL task in Section 5.1. The experimental procedure is the same as in Section 4.4.1. We use the F1-measure and ACC as evaluation values. We performed the action description task for a total of 10 actions. The placement of objects on the table was different for each trial and differed from those used during the learning trials. The robot generates a sentence of four words representing the four sensory-channels. The word order in the sentence is fixed as $F_d = (\text{a, p, c, o})$.

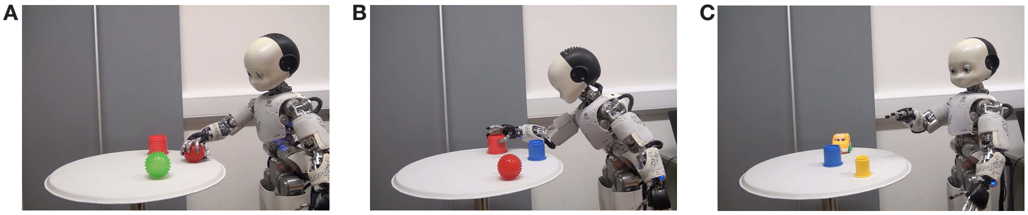
5.3.2. Results

Table 5 shows F1-measure and ACC values of action description task using the learning results under the different conditions. The

TABLE 5 | Experimental results in 25 and 40 trials.

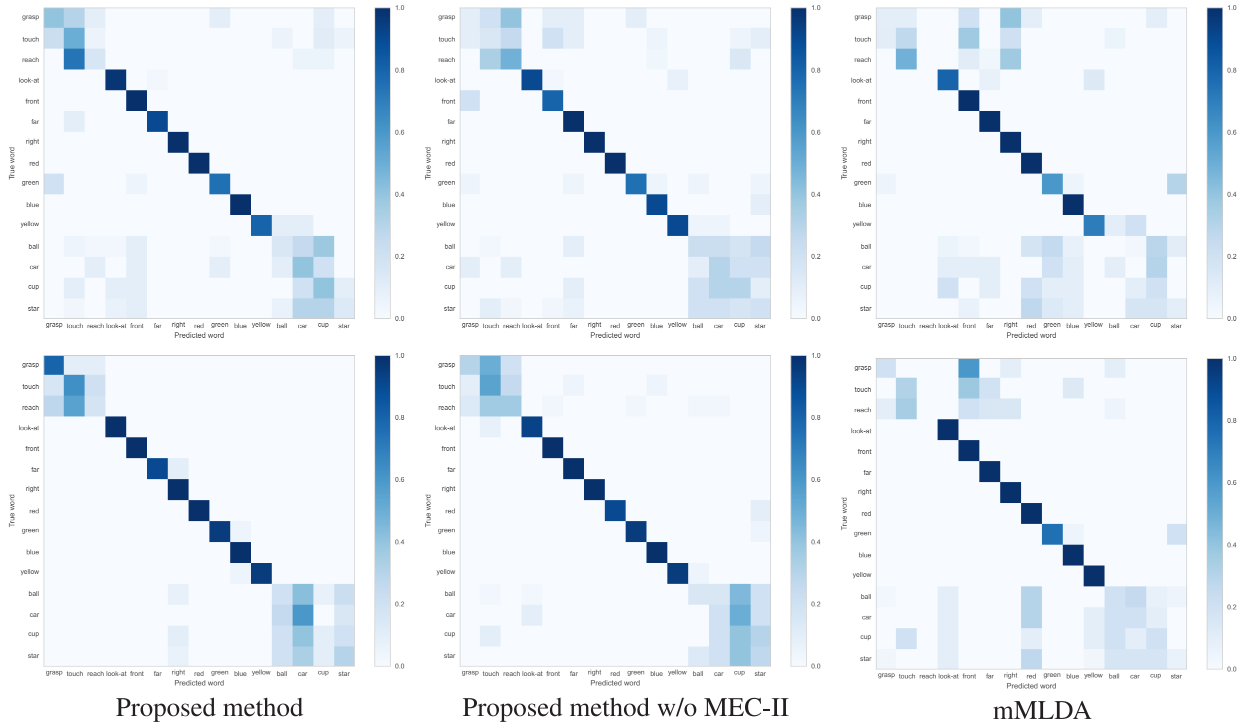
Method	Trials	ROW	F1	ACC
Proposed	25	0	**0.575**	**0.650**
w/o MEC-II	25	0	**0.558**	**0.640**
mMLDA	25	0	0.406	0.558
Proposed	40	0	**0.618**	**0.720**
w/o MEC-II	40	0	**0.654**	**0.698**
mMLDA	40	0	0.509	0.645

Bold and underscore indicate the highest evaluation values, and bold indicates the second highest evaluation values.



"grasp front red ball." "reach right red cup." "look-at left yellow cup."

FIGURE 9 | Examples of results of action generation task with real iCub. **(A)** Grasp front red ball. **(B)** Reach right red cup. **(C)** Look-at left yellow cup.



Proposed method

Proposed method w/o MEC-II

mMLDA

FIGURE 10 | Confusion matrix of results on action description task using the learning result by (top) 20 and (bottom) 40 trials under the ROW is 0%.

proposed method showed the higher evaluation values than other methods. **Figure 10** shows confusion matrices between predicted words and true words using the learning results by 20 and 40 trials. In the action category, there was a tendency to confuse words "reach" and "touch" similar to simulation. The major difference in the result for each method was found in the words of action and object categories. Even if the accuracy of categorization is low as in action and object categories and the categories include uncertainty, the robot could describe the action more correctly if the correspondence between the word and the sensory-channel was performed more properly.

6. CONCLUSION

In this paper, we have proposed a Bayesian generative model that can estimate multiple categories and the relationships between words and multiple sensory-channels. We performed experiments of cross-situational learning using the simulator and real iCub robot in complex situations. The experimental results showed that it is possible for a robot to learn the combination between a sensory-channel and a word from their co-occurrence in complex situations. The proposed method could learn word meanings from uncertain sentences, i.e., the sentence including four words or less with a changing order. In comparative experiments, we showed that the mutual exclusivity constraint is effective in the lexical acquisition by CSL. In addition, we performed experiments of action generation task and action description task by the robot that learned word meanings. The action generation task confirmed that the robot could also select an object successfully and generate an action even for situations other than those it encountered during the learning scenario. The action description task confirmed that the robot was able to use the learned word meanings to explain the current situation.

The accuracy of the categorization of objects and actions tended to be lower than those of the color and position categories. In this paper, we used GMM for the categorization of each sensory-channel. MLDA achieved highly accurate object categorization by integrating multimodal information (Nakamura et al., 2011a). The accuracy of object categorization can be improved by using MLDA instead of GMM, i.e., by increasing the number of sensory-channels for the object categories. In the action categorization, the robot confused "reach" and "touch," because these are similar actions. However, the robot is able to classify diverse actions more accurately. In addition, we used static features as action information. The accuracy can be improved by segmenting the time-series data of the actions by using a method based on the hidden Markov model (HMM) (Sugiura et al., 2011; Nakamura et al., 2016).

In this study, we performed the action generation task by a sentence including four words corresponding to the four sensory-channels. However, action instruction also presented cases in which an uttered sentence contains uncertainty. In future, we plan to investigate what kind of action the robot performs based on uncertain utterances, such as when the number of words is fewer than four, when the same objects exist on the table, and when the sentence contains the wrong word. If the robot can learn the word meanings more accurately, the robot would be able to perform an action successfully even from an utterance including uncertainty. Detailed and quantitative evaluation of such advanced action generation tasks is a subject for future work.

Other factors we aim to address in future studies are grammatical information, which was not considered in the present study, and sentences containing five words or more. We showed that the robot could accurately learn word meanings without considering grammar in the scenario of this study. However, it is important to include even more complicated situations with more natural sentences such as "grasp the red box beside the green cup." More complicated sentences would require us to consider a method that takes the grammatical structure into account. We, therefore, aim to extend the proposed method to more complicated situations and natural sentences. Attamimi et al. (2016) used HMM for the estimation of transition probabilities between words based on concepts, as a post-processing step of mMLDA. However, they were unable to use grammatical information to learn the relationships between words and categories. Hinaut et al. (2014) proposed a method based on recurrent neural networks for learning grammatical constructions by interacting with humans, which is related to the study of an autobiographical memory reasoning system (Pointeau et al., 2014). Integrating such methods with the proposed method may be effective for action generation and action description using more complicated sentences.

In this paper, we focused on mutual exclusivity of words indicating categories in language acquisition. However, there are hierarchies of categories, e.g., ball and doll belong to the toy category. Griffiths et al. (2003) proposed a hierarchical LDA (hLDA), which is a hierarchical clustering method based on a Bayesian generative model, and it was applied to objects (Ando et al., 2013) and places (Hagiwara et al., 2016). We consider the possibility of applying hLDA to the proposed method for hierarchical categorization of sensory-channels.

For future work, we also plan to demonstrate the effectiveness of the proposed method by employing a long-term experiment that uses a larger number of objects. We believe that the robot can learn more categories and word meanings based on more experience. In addition, as a further extension of the proposed method, we intend increasing the types of sensory-channels, adding a positional relationship between objects, and identifying words that are not related to sensory-channels. For example, Aly et al. (2017) learned object categories and spatial prepositions by using a model similar to the proposed model. It would be possible to merge the proposed method with this model in the theoretical framework of the Bayesian generative model. This combined model is expected to enable the robot to learn many different word meanings from situations more complicated than the scenario in this study.

AUTHOR CONTRIBUTIONS

AT, TT, and AC conceived, designed the research, and wrote the paper. AT performed the experiment and analyzed the data.

FUNDING

This work was partially supported by JST CREST and JSPS KAKENHI Grant Number JP17J07842. This material is partially based upon work supported by the Air Force Office of Scientific Research grant number: A9550-15-1-0025, Air Force Materiel Command, USAF under Award No. FA9550-15-1-0025, and by the EU H2020 Marie Skodowska-Curie European Industrial Doctorate APRIL (674868).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at https://www.frontiersin.org/articles/10.3389/fnbot.2017.00066/full#supplementary-material.

VIDEO S1 | Cross-situational learning scenario using a real iCub.

VIDEO S2 | Action generation task and action description task using a real iCub.

REFERENCES

Aly, A., Taniguchi, A., and Taniguchi, T. (2017). "A generative framework for multimodal learning of spatial concepts and object categories: an unsupervised part-of-speech tagging and 3d visual perception based approach," in *Proceedings of the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*.

Ando, Y., Nakamura, T., Araki, T., and Nagai, T. (2013). "Formation of hierarchical object concept using hierarchical latent dirichlet allocation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Tokyo: IEEE), 2272–2279.

Attamimi, M., Ando, Y., Nakamura, T., Nagai, T., Mochihashi, D., Kobayashi, I., et al. (2016). Learning word meanings and grammar for verbalization of daily life activities using multilayered multimodal latent Dirichlet allocation and Bayesian hidden Markov models. *Adv. Robot.* 30, 806–824. doi:10.1080/01691864.2016.1172507

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: parameter estimation. *Comput. Linguist.* 19, 263–311.

Cangelosi, A., and Schlesinger, M. (2015). *Developmental Robotics: From Babies to Robots. Intelligent Robotics and Autonomous Agents Series.* MIT Press.

Celikkanat, H., Orhan, G., Pugeault, N., Guerin, F., Sahin, E., and Kalkan, S. (2014). "Learning and using context on a humanoid robot using latent Dirichlet allocation," in *Proceedings of the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Genoa.

Chen, Y., Bordes, J.-B., and Filliat, D. (2016). "An experimental comparison between nmf and lda for active cross-situational object-word learning," in *Proceedings of the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Paris.

Fontanari, J. F., Tikhanoff, V., Cangelosi, A., Ilin, R., and Perlovsky, L. I. (2009). Cross-situational learning of object–word mapping using neural modeling fields. *Neural Netw.* 22, 579–585. doi:10.1016/j.neunet.2009.06.010

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2011). A sticky HDP-HMM with application to speaker diarization. *Ann. Appl. Stat.* 5, 1020–1056. doi:10.1214/10-AOAS395

Frank, M. C., Goodman, N. D., and Tenenbaum, J. B. (2007). "A bayesian framework for cross-situational word-learning," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Vancouver, 457–464.

Frank, M. C., Goodman, N. D., and Tenenbaum, J. B. (2009). Using speakers' referential intentions to model early cross-situational word learning. *Psychol. Sci.* 20, 578–585. doi:10.1111/j.1467-9280.2009.02335.x

Goldwater, S., Griffiths, T. L., and Johnson, M. (2009). A Bayesian framework for word segmentation: exploring the effects of context. *Cognition* 112, 21–54. doi:10.1016/j.cognition.2009.03.008

Griffiths, T. L., Jordan, M. I., Tenenbaum, J. B., and Blei, D. M. (2003). "Hierarchical topic models and the nested Chinese restaurant process," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Vancouver, 17–24.

Hagiwara, Y., Masakazu, I., and Taniguchi, T. (2016). "Place concept learning by hMLDA based on position and vision information," in *Proceedings of the 13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (IFAC HMS)*, Kyoto.

Harnad, S. (1990). The symbol grounding problem. *Phys. D Nonlinear Phenom.* 42, 335–346. doi:10.1016/0167-2789(90)90087-6

Heath, S., Ball, D., and Wiles, J. (2016). Lingodroids: cross-situational learning for episodic elements. *IEEE Trans. Cogn. Dev. Syst.* 8, 3–14. doi:10.1109/TAMD.2015.2442619

Heymann, J., Walter, O., Haeb-Umbach, R., and Raj, B. (2014). "Iterative Bayesian word segmentation for unsupervised vocabulary discovery from phoneme lattices," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence.

Hinaut, X., Petit, M., Pointeau, G., and Dominey, P. F. (2014). Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks. *Front. Neurorobot.* 8, 1–17. doi:10.3389/fnbot.2014.00016

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735

Hörnstein, J., Gustavsson, L., Santos-Victor, J., and Lacerda, F. (2010). "Multimodal language acquisition based on motor learning and interaction," in *From Motor Learning to Interaction Learning in Robots*, eds O. Sigaud and J. Peters (Berlin, Heidelberg: Springer), 467–489. doi:10.1007/978-3-642-05181-4_20

Hubert, L., and Arabie, P. (1985). Comparing partitions. *J. Classif.* 2, 193–218. doi:10.1007/BF01908075

Imai, M., and Mazuka, R. (2007). Language-relative construal of individuation constrained by universal ontology: revisiting language universals and linguistic relativity. *Cogn. Sci.* 31, 385–413. doi:10.1080/15326900701326436

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). Caffe: convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093.*

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Nevada, 1097–1105.

Markman, E. M., and Hutchinson, J. E. (1984). Children's sensitivity to constraints on word meaning: taxonomic versus thematic relations. *Cogn. Psychol.* 16, 1–27. doi:10.1016/0010-0285(84)90002-1

Markman, E. M., and Wachtel, G. F. (1988). Children's use of mutual exclusivity to constrain the meanings of words. *Cogn. Psychol.* 20, 121–157. doi:10.1016/0010-0285(88)90017-5

Matuszek, C., Fitzgerald, N., Zettlemoyer, L., Bo, L., and Fox, D. (2012). "A joint model of language and perception for grounded attribute learning," in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, Edinburgh, 1671–1678.

Morse, A. F., Belpaeme, T., Cangelosi, A., and Smith, L. B. (2010). "Thinking with your body: modelling spatial biases in categorization using a real humanoid robot," in *Proceedings of the Annual Meeting of the Cognitive Science Society* (Portland, USA), 1362–1368.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press.

Nakamura, T., Araki, T., Nagai, T., and Iwahashi, N. (2011a). Grounding of word meanings in latent Dirichlet allocation-based multimodal concepts. *Adv. Robot.* 25, 2189–2206. doi:10.1163/016918611X595035

Nakamura, T., Nagai, T., and Iwahashi, N. (2011b). "Multimodal categorization by hierarchical Dirichlet process," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (San Francisco, CA: IEEE), 1520–1525.

Nakamura, T., Iwata, K., Nagai, T., Mochihashi, D., Kobayashi, I., Asoh, H., et al. (2016). "Continuous motion segmentation based on reference point dependent GP-HSMM," in *Proceedings of the IROS Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics*, Daejeon.

Pointeau, G., Petit, M., and Dominey, P. F. (2014). Successive developmental levels of autobiographical memory for learning through social interaction. *IEEE Trans. Auton. Ment. Dev.* 6, 200–212. doi:10.1109/TAMD.2014.2307342

Qu, S., and Chai, J. Y. (2008). "Incorporating temporal and semantic information with eye gaze for automatic word acquisition in multimodal conversational systems," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Honolulu, 244–253.

Qu, S., and Chai, J. Y. (2010). Context-based word acquisition for situated dialogue in a virtual world. *J. Artif. Intell. Res.* 37, 247–278. doi:10.1613/jair.2912

Roy, D., and Pentland, A. (2002). Learning words from sights and sounds: a computational model. *Cogn. Sci.* 26, 113–146. doi:10.1207/s15516709cog2601_4

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Stat. Sin.* 4, 639–650.

Smith, K., Smith, A. D., and Blythe, R. A. (2011). Cross-situational learning: an experimental study of word-learning mechanisms. *Cogn. Sci.* 35, 480–498. doi:10.1111/j.1551-6709.2010.01158.x

Smith, L. B., and Samuelson, L. (2010). "Objects in space and mind: from reaching to words," in *Spatial Foundations of Cognition and Language: Thinking Through Space*, eds K. S. Mix, L. B. Smith, and M. Gasser (Oxford: Oxford University Press), 188–207. doi:10.1093/acprof:oso/9780199553242.001.0001

Spranger, M. (2015). "Incremental grounded language learning in robot-robot interactions-examples from spatial language," in *Proceedings of the Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* (Providence, RI: IEEE), 196–201.

Spranger, M., and Steels, L. (2015). "Co-acquisition of syntax and semantics: an investigation in spatial language," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, Buenos Aires, 1909–1915.

Steels, L., and Hild, M. (2012). *Language Grounding in Robots*. New York: Springer Science & Business Media.

Stramandinoli, F., Marocco, D., and Cangelosi, A. (2017). Making sense of words: a robotic model for language abstraction. *Auton. Robots*. 41, 367–383. doi:10.1007/s10514-016-9587-8

Sugiura, K., Iwahashi, N., Kashioka, H., and Nakamura, S. (2011). Learning, generation and recognition of motions by reference-point-dependent probabilistic models. *Adv. Robot.* 25, 825–848. doi:10.1163/016918611X563328

Taniguchi, A., Taniguchi, T., and Inamura, T. (2016a). "Simultaneous estimation of self-position and word from noisy utterances and sensory information," in *Proceedings of the 13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems (IFAC HMS)*, Kyoto.

Taniguchi, A., Taniguchi, T., and Inamura, T. (2016b). Spatial concept acquisition for a mobile robot that integrates self-localization and unsupervised word discovery from spoken sentences. *IEEE Trans. Cogn. Dev. Syst.* 8, 285–297. doi:10.1109/TCDS.2016.2565542

Taniguchi, T., Nagai, T., Nakamura, T., Iwahashi, N., Ogata, T., and Asoh, H. (2016c). Symbol emergence in robotics: a survey. *Adv. Robot.* 30, 706–728. doi:10.1080/01691864.2016.1164622

Taniguchi, T., Nakashima, R., Liu, H., and Nagasaka, S. (2016d). Double articulation analyzer with deep sparse autoencoder for unsupervised word discovery from speech signals. *Adv. Robot.* 30, 770–783. doi:10.1080/01691864.2016.1159981

Tikhanoff, V., Cangelosi, A., Fitzpatrick, P., Metta, G., Natale, L., and Nori, F. (2008). "An open-source simulator for cognitive robotics research: the prototype of the icub humanoid robot simulator," in *Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems* (Gaithersburg: ACM), 57–61.

Tomasello, M., and Farrar, M. J. (1986). Joint attention and early language. *Child Dev.* 57, 1454–1463. doi:10.2307/1130423

Twomey, K. E., Morse, A. F., Cangelosi, A., and Horst, J. S. (2016). Children's referent selection and word learning: insights from a developmental robotic system. *Interact. Stud.* 17, 93–119. doi:10.1075/is.17.1.05two

Yamada, T., Murata, S., Arie, H., and Ogata, T. (2015). "Attractor representations of language-behavior structure in a recurrent neural network for human-robot interaction," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg: IEEE), 4179–4184.

Yamada, T., Murata, S., Arie, H., and Ogata, T. (2016). Dynamical integration of language and behavior in a recurrent neural network for human-robot interaction. *Front. Neurorobot.* 10:5. doi:10.3389/fnbot.2016.00005

Zhong, J., Cangelosi, A., and Ogata, T. (2017). "Toward abstraction from multimodal data: empirical studies on multiple time-scale recurrent models," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 21 December 2017
doi: 10.3389/fnbot.2017.00067

Segmenting Continuous Motions with Hidden Semi-markov Models and Gaussian Processes

Tomoaki Nakamura [1]*, *Takayuki Nagai* [1], *Daichi Mochihashi* [2], *Ichiro Kobayashi* [3], *Hideki Asoh* [4] *and Masahide Kaneko* [1]

[1] *Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, Chofu-shi, Japan,* [2] *Department of Mathematical Analysis and Statistical Inference, Institute of Statistical Mathematics, Tachikawa, Japan,* [3] *Department of Information Sciences, Faculty of Sciences, Ochanomizu University, Bunkyo-ku, Japan,* [4] *Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan*

Humans divide perceived continuous information into segments to facilitate recognition. For example, humans can segment speech waves into recognizable morphemes. Analogously, continuous motions are segmented into recognizable unit actions. People can divide continuous information into segments without using explicit segment points. This capacity for unsupervised segmentation is also useful for robots, because it enables them to flexibly learn languages, gestures, and actions. In this paper, we propose a Gaussian process-hidden semi-Markov model (GP-HSMM) that can divide continuous time series data into segments in an unsupervised manner. Our proposed method consists of a generative model based on the hidden semi-Markov model (HSMM), the emission distributions of which are Gaussian processes (GPs). Continuous time series data is generated by connecting segments generated by the GP. Segmentation can be achieved by using forward filtering-backward sampling to estimate the model's parameters, including the lengths and classes of the segments. In an experiment using the CMU motion capture dataset, we tested GP-HSMM with motion capture data containing simple exercise motions; the results of this experiment showed that the proposed GP-HSMM was comparable with other methods. We also conducted an experiment using karate motion capture data, which is more complex than exercise motion capture data; in this experiment, the segmentation accuracy of GP-HSMM was 0.92, which outperformed other methods.

Keywords: motion segmentation, Gaussian process, hidden semi-Markov model, motion capture data

Edited by:
Ganesh R. Naik,
Western Sydney University, Australia

Reviewed by:
Douglas Scott Blank,
Bryn Mawr College, United States
Suparerk Janjarasjitt,
Ubon Ratchathani University, Thailand
Marc De Kamps,
University of Leeds, United Kingdom

Correspondence:
Tomoaki Nakamura
tnakmaura@uec.ac.jp

Received: *22 May 2017*
Accepted: *29 November 2017*
Published: *21 December 2017*

Citation:
Nakamura T, Nagai T, Mochihashi D, Kobayashi I, Asoh H and Kaneko M (2017) Segmenting Continuous Motions with Hidden Semi-markov Models and Gaussian Processes. Front. Neurorobot. 11:67. doi: 10.3389/fnbot.2017.00067

1. INTRODUCTION

Human beings typically divide perceived continuous information into segments to enable recognition. For example, humans can segment speech waves into recognizable morphemes. Similarly, continuous motions are segmented into recognizable unit actions. In particular, motions are divided into smaller components called motion primitives, which are used for imitation learning and motion generation (Argall et al., 2009; Lin et al., 2016). It is possible for us to divide continuous information into segments without using explicit segment points. This capacity for unsupervised segmentation is also useful for robots, because it enables them to flexibly learn languages, gestures, and actions.

However, segmentation of time series data is a difficult task. When time series data is segmented, the data points in the sequence must be classified, and each segment's start and end points must be determined. Moreover, each segment affects other segments because of the nature of time series data. Hence, segmentation of time series data requires the exploration of all possible segment lengths and classes. However, this exploration process is difficult; in many studies, the lengths are not estimated explicitly or heuristics are used to reduce computational complexity. Furthermore, in the case of motions, the sequences vary because of dynamic characteristics, even though the same movements are performed. For segmentation of actual human motions, we must address such variations.

In this paper, we propose GP-HSMM (Gaussian process–hidden semi-Markov model), a novel method to divide time series motion data into unit actions by using a stochastic model to estimate their lengths and classes. The proposed method involves a hidden semi-Markov model (HSMM) with a Gaussian process (GP) emission distribution, where each state represents a unit action. **Figure 1** shows an overview of the proposed GP-HSMM. The observed time series data is generated by connecting segments generated by each class. The segment points and segment classes are estimated by learning the parameters of the model in an unsupervised manner. Forward filtering-backward sampling (Uchiumi et al., 2015) is used for the learning process; the segment lengths and segment classes are determined by sampling them simultaneously.

2. RELATED WORK

Various studies have focused on learning motion primitives from manually segmented motions (Gräve and Behnke, 2012; Manschitz et al., 2015). Manschitz et al. proposed a method to generate sequential skills by using motion primitives that are learned in a supervised manner. Gräve et al. proposed segmenting motions using motion primitives that are learned by a supervised hidden Markov model. In these studies, the motions are segmented and labeled in advance. However, we consider that it is difficult to segment and label all possible motion primitives.

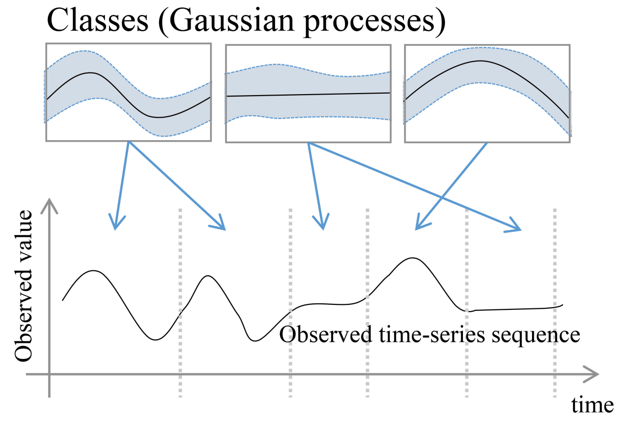


FIGURE 1 | Overview of the proposed GP-HSMM.

Additionally, some studies have proposed unsupervised motion segmentation. However, these studies rely on heuristics. For instance, Wächter et al. have proposed a method to segment human manipulation motions based on contact relations between the end-effectors and objects in a scene (Wachter and Asfour, 2015); in their method, the points at which the end-effectors make contact with an object are determined as boundaries of motions. We believe this method works well in limited scenes; however, there are many motions, such as gestures and dances, in which objects are not manipulated. Lioutikov et al. proposed unsupervised segmentation; however, to reduce computational costs, this technique requires the possible boundary candidates between motion primitives to be specified in advance (Lioutikov et al., 2015). Therefore, the segmentation depends on those candidates, and motions cannot be segmented correctly if the correct candidates are not selected. In contrast, our proposed method does not require such candidates; all possible cutting points are considered by use of forward filtering-backward sampling, which uses the principles of dynamic programming. In some methods (Fod et al., 2002; Shiratori et al., 2004; Lin and Kulić, 2012), motion features (such as the zero velocity of joint angles) are used for motion segmentation. However, these features cannot be applied to all motions. Takano et al. use the error between actual movements and predicted movements as the criteria for specifying boundaries (Takano and Nakamura, 2016). However, the threshold must be manually tuned according to the motions to be segmented. Moreover, they used an HMM that is a stochastic model. We consider such an assumption to be unnatural from the viewpoint of stochastic models, and boundaries should be determined based on a stochastic model. In our proposed method, we do not use such heuristics and assumptions, and instead formulate the segmentation based on a stochastic model.

Fox et al. have proposed unsupervised segmentation for the discovery of a set of latent, shared dynamical behaviors in multiple time series data (Fox et al., 2011). They introduce a beta process, which represents a share of motion primitives in multiple motions, into autoregressive HMM. They formulate the segmentation using a stochastic model, and no heuristics are used in their proposed model. However, in their proposed method, continuous data points that are classified into the same states are extracted as segments, and the lengths of the segments are not estimated. The states can be changed in the short term, and therefore shorter segments are estimated. They reported that some true segments were split into two or more categories, and that those shorter segments were bridged in their experiment. On the other hand, our proposed method classifies data points into states, and uses HSMM to estimate segment lengths. Hence, our proposed method can prevent states from being changed in the short term.

Matsubara et al. proposed an unsupervised segmentation method called AutoPlait (Matsubara et al., 2014). This method uses multiple HMMs, each of which represents a fixed pattern; moreover, transitions between the HMMs are allowed. Therefore, time series data is segmented at points at which the state is changed to another HMM's state. However, we believe that HMMs are too simple to represent complicated sequences such as motions. **Figure 2** illustrates an example of representation of time series data by HMM. The graph on the right in **Figure 2** represents the mean and standard deviation learned by HMM from data points shown in the graph on the left. HMM represents time series data using only the mean and standard deviation; therefore, details of time series data can be lost. Therefore, we use Gaussian processes, which are non-parametric methods that can represent complex time series data.

The field of natural language processing has also produced literature related to sequence data segmentation. For example, unsupervised morphological analysis has been proposed for segmenting sequence data (Goldwater, 2006; Mochihashi et al., 2009; Uchiumi et al., 2015). Goldwater et al. proposed a method to divide sentences into words by estimating the parameters of a 2-gram language model based on a hierarchical Dirichlet process. The parameters are estimated in an unsupervised manner by Gibbs sampling (Goldwater, 2006). Mochihashi et al. proposed a nested Pitman-Yor language model (NPYLM) (Mochihashi et al., 2009). In this method, parameters of an n-gram language model based on the hierarchical Pitman-Yor process are estimated via the forward filtering-backward sampling algorithm. NPYLM can thus divide sentences into words more quickly and accurately than the method proposed in (Goldwater, 2006). Moreover, Uchiumi et al. extended the NPYLM to a Pitman-Yor hidden semi-Markov model (PY-HSMM) (Uchiumi et al., 2015) that can divide sentences into words and estimate the parts of speech (POS) of the words by sampling not only words, but also POS in the sampling phase of the forward filtering-backward sampling algorithm. However, these relevant studies aimed to divide symbolized sequences (such as sentences) into segments, and did not consider analogous divisions in continuous sequence data, such as that obtained by analyzing human motion.

Taniguchi et al. proposed a method to divide continuous sequences into segments by utilizing NPYLM (Taniguchi and Nagasaka, 2011). In their method, continuous sequences are discretized and converted into discrete-valued sequences using the infinite hidden Markov model (Fox et al., 2007). The discrete-valued sequences are then divided into segments by using NPYLM. In this method, motions can be recognized by the learned model, but cannot be generated naively because they are discretized. Moreover, segmentation based on NPYLM does not work well if errors occur in the discretization step.

Therefore, we propose a method to divide a continuous sequence into segments without using discretization. This method divides continuous motions into unit actions. Our proposed method is based on HSMM, the emission distribution of which is GP, which represents continuous unit actions. To learn the model parameters, we use forward filtering-backward sampling, and segment points and classes are sampled simultaneously. However, our proposed method also has limitations. One limitation is that the method requires the number of motion classes to be specified in advance. It is estimated automatically in methods such as (Fox et al., 2011) and (Matsubara et al., 2014). Another limitation is that computational costs are very high, owing to the numerous recursive calculations. We discuss these limitations in the experiments.

3. GAUSSIAN PROCESS-HIDDEN SEMI-MARKOV MODEL

Figure 3 shows a graphical representation of the proposed GP-HSMM. In this figure, $c_j (j = 1, 2, \cdots, J)$ denotes classes

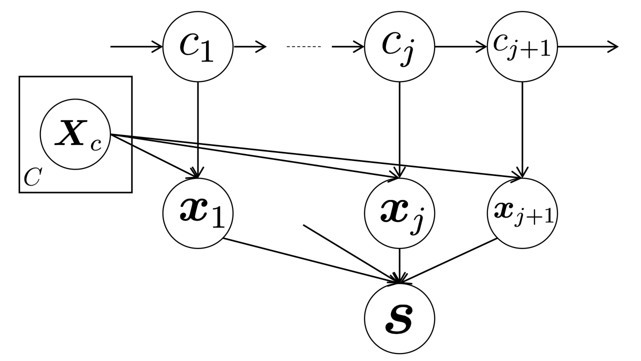


FIGURE 3 | Graphical representation of the proposed GP-HSMM.

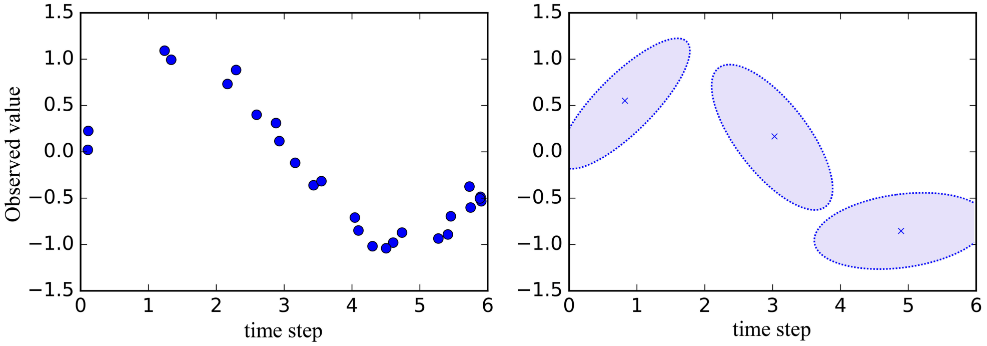


FIGURE 2 | Example of representation of time series data by HMM. **Left:** Data points for learning HMM. **Right:** Mean and standard deviation learned by HMM.

of segments, and each segment $\boldsymbol{x}_j$ is generated by a Gaussian process, with parameters denoted by $\boldsymbol{X}_c$ and given by the following generative process:

$$c_j \sim P(c|c_{j-1}), \tag{1}$$

$$\boldsymbol{x}_j \sim \mathcal{GP}(\boldsymbol{x}|\boldsymbol{X}_{c_j}), \tag{2}$$

where $\boldsymbol{X}_c$ represents a set of segments classified into class c. Segments are generated by this generative process, and the observed time-series data $\boldsymbol{s}$ is generated by connecting the segments.

3.1. Gaussian Process

In this study, we utilize Gaussian process regression, which learns emission x_i of time step i in a segment. This makes it possible to represent each unit action as part of a continuous trajectory. If we obtain pairs $(\boldsymbol{i}, \boldsymbol{X}_c)$ of emissions x_i of time step i of segments belonging to the same class c, a predictive distribution whereby the emission of time step i becomes x follows a Gaussian distribution.

$$p(x|i, \boldsymbol{X}_c, \boldsymbol{i}) \propto \mathcal{N}(\boldsymbol{k}^T \boldsymbol{C}^{-1} \boldsymbol{i}, c - \boldsymbol{k}^T \boldsymbol{C}^{-1} \boldsymbol{k}), \tag{3}$$

where $k(\cdot, \cdot)$ represents the kernel function and $\boldsymbol{C}$ is a matrix whose elements are

$$C(i_p, i_q) = k(i_p, i_q) + \beta^{-1} \delta_{pq}. \tag{4}$$

β is a hyperparameter that represents noise in the observation. In Equation (3), $\boldsymbol{k}$ is a vector containing the elements $k(i_p, i)$, and c is a scalar value $k(i, i)$. Using the kernel function, GP can learn a time-series sequence that contains complex changes. We use the following Gaussian kernel, which is generally used for Gaussian process regression:

$$k(i_p, i_q) = \theta_0 \exp(-\frac{1}{2}||i_p - i_q||^2 + \theta_2 + \theta_3 i_p i_q), \tag{5}$$

where θ_* represents parameters of the kernel. **Figure 4** shows examples of Gaussian processes. The left graph in each pair of graphs represents learning data points $(\boldsymbol{i}, \boldsymbol{X}_c)$, and the right graph shows the learned probabilistic distribution $p(x|i, \boldsymbol{X}_c, \boldsymbol{i})$. One can see that the standard deviation decreases with an increase in the number of learning data points. If the emission of time step i is multidimensional vector $\boldsymbol{x} = (x_0, x_1, \cdots)$, we assume that each dimension is generated independently, and a predictive distribution $\mathcal{GP}(\boldsymbol{x}|\boldsymbol{X}_c)$ is computed as follows:

$$\begin{aligned} \mathcal{GP}(\boldsymbol{x}|\boldsymbol{X}_c) = \; & p(x_0|i, \boldsymbol{X}_{c,0}, \boldsymbol{i}_c) \\ & \times p(x_1|i, \boldsymbol{X}_{c,1}, \boldsymbol{i}_c) \\ & \times p(x_2|i, \boldsymbol{X}_{c,2}, \boldsymbol{i}_c) \cdots . \end{aligned} \tag{6}$$

Based on this probability, similar segments can be classified into the same class.

3.2. Learning of GP-HSMM

3.2.1. Blocked Gibbs Sampler

Segments and classes of segments in the observed sequences are estimated based on dynamic programming and sampling. For efficient sampling, we use the blocked Gibbs sampler, which samples segments and their classes in an observed sequence. In the initialization phase, all observed sequences are first randomly divided into segments. Segments $\boldsymbol{x}_{nj} (j = 1, 2, \cdots, J_n)$ in observed sequence $\boldsymbol{s}_n$ are then removed from the learning data, and parameter $\boldsymbol{X}_c$ of the Gaussian process and transition probability $P(c|c')$ of HSMM are updated. Segments $\boldsymbol{x}_{nj} (j = 1, 2, \cdots, J_n)$ and their classes $c_{nj} (j = 1, 2, \cdots, J_n)$ are then estimated as follows:

$$(\boldsymbol{x}_{n1}, \cdots, \boldsymbol{x}_{nJ_n}), (c_{n1}, \cdots, c_{nJ_n}) \sim P(\boldsymbol{X}, \boldsymbol{c}|\boldsymbol{s}_n), \tag{7}$$

where $\boldsymbol{X}$ is a set of segments into which $\boldsymbol{s}_n$ is divided, and $\boldsymbol{c}$ denotes classes of the segments. To carry out this sampling efficiently, the probability of all possible segments $\boldsymbol{X}$ and

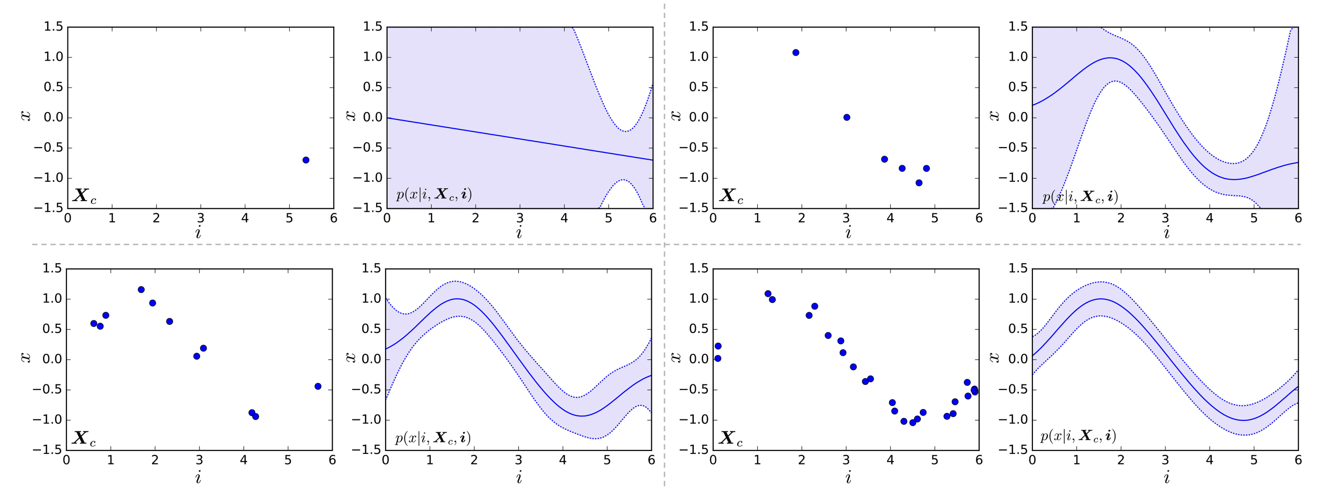


FIGURE 4 | Examples of Gaussian processes. Left graph in each pair of graphs represents learning data points $(\boldsymbol{i}, \boldsymbol{X}_C)$. Right graph shows the learned probabilistic distribution $p(x|i, \boldsymbol{X}_C, \boldsymbol{i})$; the solid line represents the mean, and the blue region represents the range of standard deviation.

Algorithm 1 Blocked Gibbs Sampler

```
// Iterate the following procedure until convergence
for $n = 1$ to $N$ do
    for $j = 1$ to $J_n$ do
        $N_{c_{nj}} - = 1$
        $N_{c_{nj},c_{n,j+1}} - = 1$
        if $j \neq 0$ then
            Delete segments $\boldsymbol{x}_{nj}$ from $\boldsymbol{X}_{c_{nj}}$
        end if
    end for

    // Sample segments and their classes
    $(\boldsymbol{x}_{n1}, \cdots, \boldsymbol{x}_{nJ_n}), (c_{n1}, \cdots, c_{nJ_n}) \sim P(\boldsymbol{X}, \boldsymbol{c}|\boldsymbol{s}_n)$

    for $j = 1$ to $J_n$ do
        $N_{c_{nj}} + +$
        $N_{c_{nj},c_{n,j+1}} + +$
        if $j \neq$ then
            Add segments $\boldsymbol{x}_{nj}$ into $\boldsymbol{X}_{c_{nj}}$
        end if
    end for
end for
```

Algorithm 2 Forward filtering-backward sampling

```
// Forward filtering
for $t = 1$ to $T$ do
    for $k = 1$ to $K$ do
        for $c = 1$ to $C$ do
            Compute $\alpha[t][k][c]$
        end for
    end for
end for

// Backward sampling
$t = T, j = 1$
while $t > 0$ do
    $k, c \sim \alpha[t][k][c]$
    $\boldsymbol{x}_j = \boldsymbol{s}_{t-k:t}$
    $c_j = c$
    $t = t - k$
    $j = j + 1$
end while
return $(\boldsymbol{x}_{J_n}, \boldsymbol{x}_{J_n-1}, \cdots, \boldsymbol{x}_1), (c_{J_n}, c_{J_n-1}, \cdots, c_1)$
```

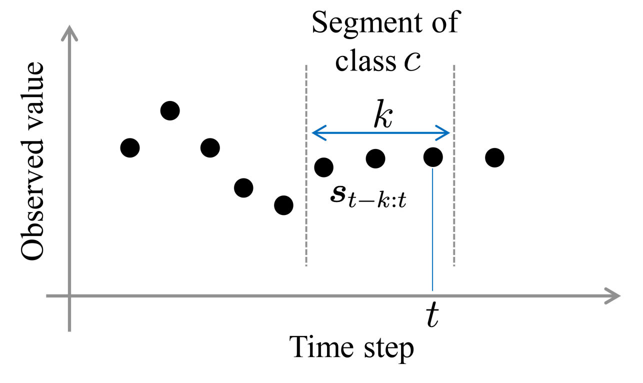


FIGURE 5 | A segment whose probability is computed during forward filtering.

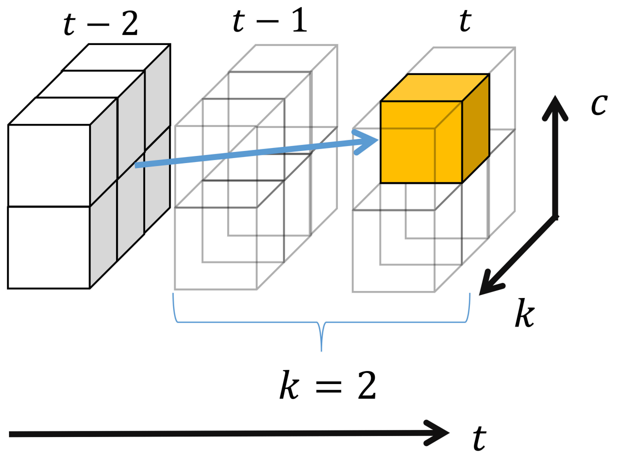


FIGURE 6 | Recursive computation in forward filtering.

classes $\boldsymbol{c}$ must be computed; however, these probabilities are difficult to compute simply because the number of potential combinations is very large. Thus, we utilize forward filtering-backward sampling, which we presently explain. After sampling $\boldsymbol{x}_{nj}$ and c_{nj}, parameter $\boldsymbol{X}_c$ of the Gaussian process and transition probability $P(c|c')$ of HSMM are updated by adding them to the learning data. The segments and parameters of Gaussian processes are optimized alternately by iteratively performing the above procedure. Algorithm 1 shows the pseudocode of the blocked Gibbs sampler. $N_{c_{nj}}$ and $N_{c_{nj},\, c_{n,\, j+1}}$ represent parameters for computing the transition probability in Equation (10).

3.2.2. Forward Filtering-Backward Sampling

In this study, we regard segments and their classes as latent variables that are sampled by forward filtering-backward sampling (Algorithm 2). In forward filtering, as shown in Figure 5, the probability that k samples $\boldsymbol{s}_{t-k:t}$ prior to time step t in observed sequence $\boldsymbol{s}$ form a segment, and that the resulting segment belongs to class c, is computed as follows:

$$\alpha[t][k][c] = P(\boldsymbol{s}_{t-k:t}|\boldsymbol{X}_c) \times \sum_{k'=1}^{K}\sum_{c'=1}^{C} p(c|c')\alpha[t-k][k'][c'], \quad (8)$$

where C and K denote the number of classes and the maximum length of segments, respectively. $P(\boldsymbol{s}_{t-k:t}|\boldsymbol{X}_c)$ represents the probability that $\boldsymbol{s}_{t-k:t}$ is generated from a class c; this is computed as follows:

$$P(\boldsymbol{s}_{t-k:t}|\boldsymbol{X}_c) = \mathcal{GP}(\boldsymbol{s}_{t-k:t}|\boldsymbol{X}_c)P_{len}(k|\lambda). \quad (9)$$

where $P_{len}(k|\lambda)$ represents a Poisson distribution with a mean parameter λ; this corresponds to the distribution of the segment lengths. $p(c|c')$ in Equation (8) represents a transition probability computed as follows:

$$p(c|c') = \frac{N_{c'c} + \alpha}{N_{c'} + C\alpha}, \quad (10)$$

where $N_{c'}$ and $N_{c'c}$ denote the number of segments whose classes are c' and the number of transitions from c' to c, respectively, and k' and c' respectively denote the length and class of the segment preceding $\boldsymbol{s}_{t-k:t}$; these are marginalized out in Equation (8). Moreover, $\alpha[t][k][*] = 0$ if $t - k < 0$, and $\alpha[0][0][*] = 1.0$. All elements of $\alpha[*][*][*]$ in Equation (8) can be recursively computed from $\alpha[1][1][*]$ by dynamic programming. **Figure 6** depicts the computation of a three-dimensional array $\alpha[t][k][c]$. In this example, the probability that two samples before time step t become a segment is computed; the resulting segment would be assigned to class two. Hence, samples at $t - 1$ and t become a segment, and all the segments whose end point is $t - 2$ can potentially transit to this segment. $\alpha[t][2][2]$ can be computed by marginalizing out these possibilities.

Finally, segment $\boldsymbol{x}_j$ and its class are determined by backward sampling length k and class c of the segment, based on forward probabilities in α. From $t = T$, length k_1 and class c_1 are determined according to $k_1, c_1 \sim \alpha[T][k][c]$, and $\boldsymbol{s}_{T-k_1:T}$ becomes a segment whose class is c_1. Then, length k_2 and class c_2 of the next segment are determined according to $k_2, c_2 \sim \alpha[T - k_1][k][c]$. By iterating this procedure until $t = 0$, the observed sequence can be divided into segments and their classes can be determined.

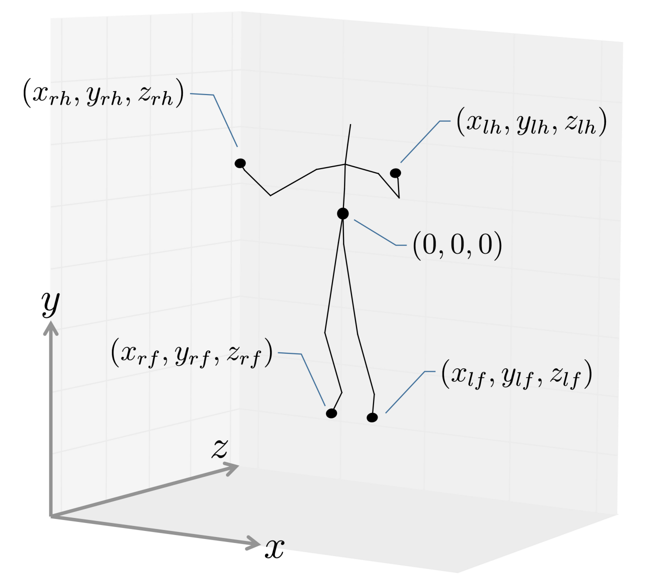

(x_{rh}, y_{rh}, z_{rh})
(x_{lh}, y_{lh}, z_{lh})
$(0, 0, 0)$
y
(x_{rf}, y_{rf}, z_{rf})
(x_{lf}, y_{lf}, z_{lf})
z
x

FIGURE 7 | Coordinate system used in the experiments.

4. EXPERIMENTS

We conducted experiments to confirm the validity of the proposed method. We used two types of motion capture data: (1) data from the CMU motion capture dataset (CMU, 2009), and (2) data containing karate motions.

4.1. Segmentation of Exercise Motions

We first applied our proposed method to CMU motion capture data containing several exercise routines. The CMU motion capture data was captured using a Vicon motion capture system, and positions and angles of 31 body parts are available. The dataset contains 2605 trials in six categories and 23 subcategories, and motions in each subcategory were performed by one or a few subjects. In this experiment, three sequences from subject 14 in the general exercise and stretching category were used, and include running, jumping, squats, knee raises, reach out stretches, side stretches, body twists, up and down movements, and toe touches. To reduce computational cost, we downsampled from 120 frames per second to 4 frames per second. **Figure 7** shows the coordinate system of motion capture data used in this experiment; two-dimensional frontal views of the left hand (x_{lh}, y_{lh}), right hand (x_{rh}, y_{rh}), left foot(x_{lf}, y_{lf}), and right foot (x_{rf}, y_{rf}) were used. Therefore, each frame was represented by eight dimensional vectors:

TABLE 1 | Segmentation accuracy of CMU motion capture data.

Hamming distance	Precision	Recall	F-measure
0.33	0.81	0.81	0.81

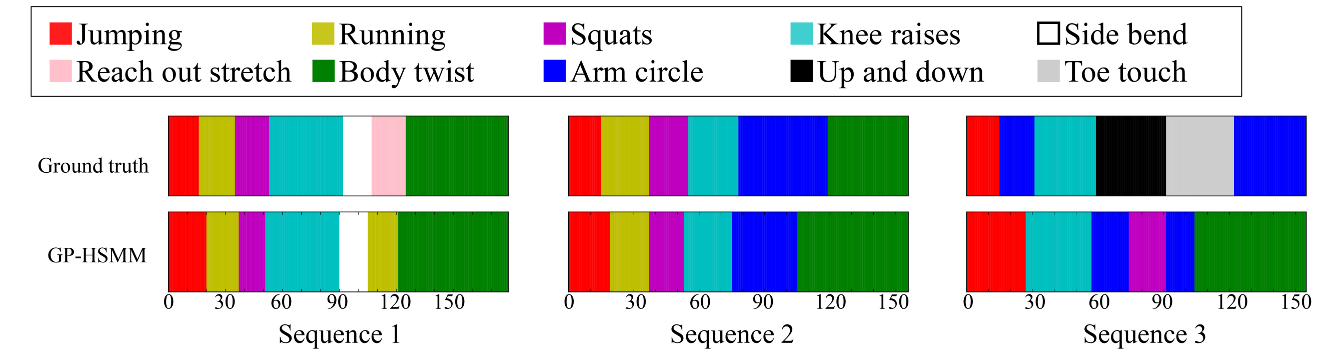
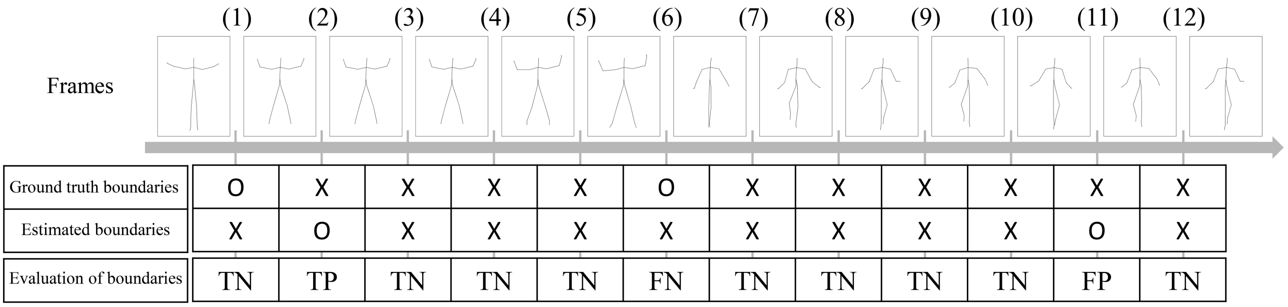


FIGURE 8 | Segmentation results of CMU motion capture data.



	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)
Ground truth boundaries	O	X	X	X	X	O	X	X	X	X	X	X
Estimated boundaries	X	O	X	X	X	X	X	X	X	X	O	X
Evaluation of boundaries	TN	TP	TN	TN	TN	FN	TN	TN	TN	TN	FP	TN

FIGURE 9 | Example of segmentation evaluation. Estimated boundaries are evaluated as true positive (TP), true negative (TN), false positive (FP), or false negative (FN).

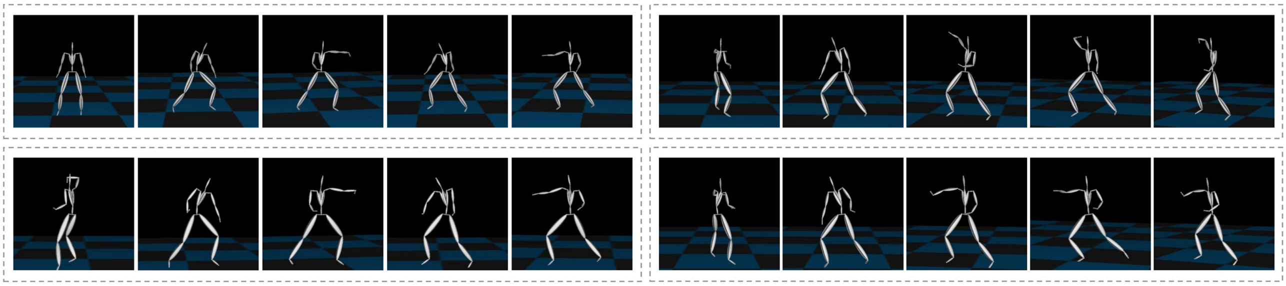


FIGURE 10 | Motion capture data of karate motions.

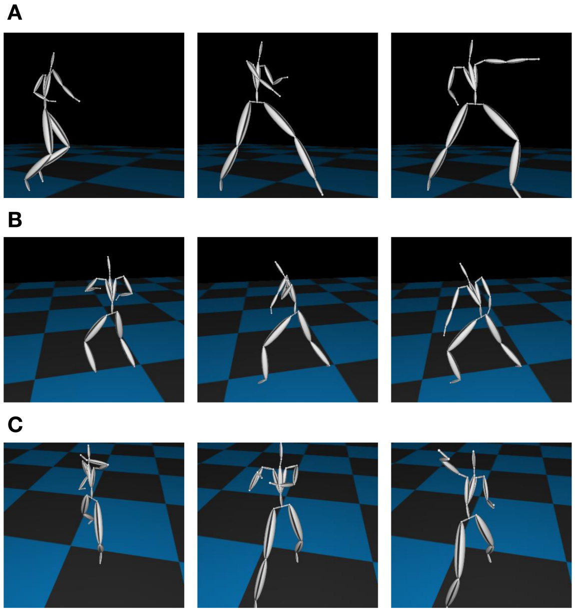


FIGURE 11 | Basic motions in Kata: **(A)** Left punch. **(B)** Left lower guard. **(C)** Right upper guard.

$(x_{lh}, y_{lh}, x_{rh}, y_{rh}, x_{lf}, y_{lf}, x_{rf}, y_{rf})$. Because GP-HSMM requires the number of classes to be specified in advance, we set it to eight.

Figure 8 shows the results of the segmentation. The horizontal axis represents the frame number, and the colors represent motion classes into which each segment was classified. The segments were classified into seven classes out of eight. **Table 1** shows the accuracy of the segmentation. We computed the following normalized Hamming distance between the unsupervised segmentation and the ground truth:

$$ND(\boldsymbol{c}, \bar{\boldsymbol{c}}) = \frac{D(\boldsymbol{c}, \bar{\boldsymbol{c}})}{|\bar{\boldsymbol{c}}|}, \tag{11}$$

where $\boldsymbol{c}$ and $\bar{\boldsymbol{c}}$ represent sequences of estimated motion classes and true motion classes, $D(\boldsymbol{c}, \bar{\boldsymbol{c}})$ is the Hamming distance between two sequences, and $|\bar{\boldsymbol{c}}|$ represents the length of the sequence. Therefore, the normalized Hamming distance ranges from 0 to 1; lower Hamming distances indicate more accurate segmentation. In this experiment, the Hamming distance was 0.33, which is comparable with the BP-HMM reported in (Fox et al., 2011). However, they also reported that some segments were split into two or more categories, and that those shorter segments were bridged. In contrast, we performed no such modifications, and **Figure 8** shows that there are no shorter segments. We also computed the precision, recall, and F-measure of the segmentation. To compute them, estimated boundaries of segments are evaluated as true positive (TP), true negative (TN), false positive (FP), or false negative (FN). **Figure 9** shows an example of segmentation evaluation. We considered the estimated boundary to be TP if it was within true boundary ± four frames, as shown in **Figure 9**(2). If

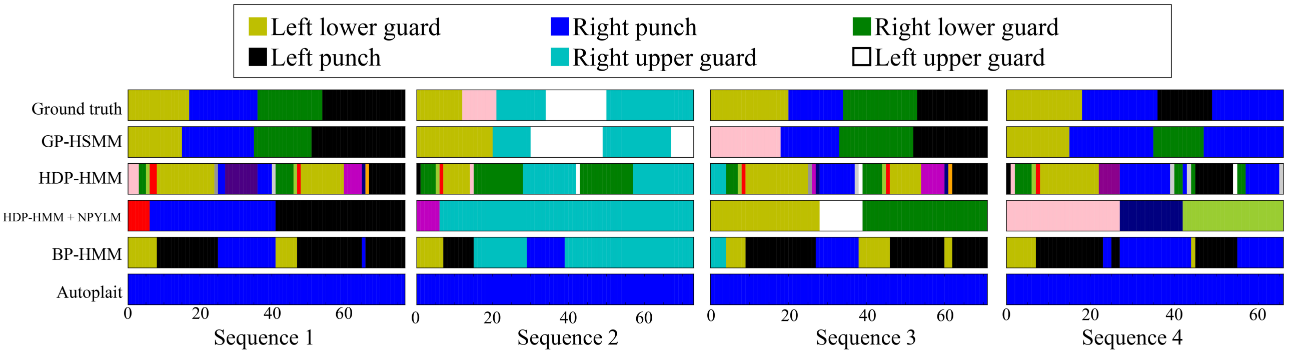


FIGURE 12 | Results of segmentation and classification for each method.

TABLE 2 | Segmentation accuracy of karate motions.

	Hamming distance	Precision	Recall	F-measure
GP-HSMM	0.21	0.92	0.92	0.92
HDP-HMM	0.47	0.12	0.54	0.19
HDP-HMM + NPYLM	0.61	0.00	0.00	0.00
BP-HMM	0.49	0.13	0.23	0.16
AutoPlait	0.76	0.00	0.00	0.00

the ground truth boundary has no corresponding estimated boundary as shown in **Figure 9**(6), it was considered as FN. Conversely, if the estimated boundary has no corresponding ground truth boundary as shown in **Figure 9**(11), it was considered as FP. From these evaluations, the precision, recall, and F-measure of the segmentation are computed as follows:

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \tag{12}$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \tag{13}$$

$$F = \frac{2PR}{P + R}, \tag{14}$$

where N_{TP}, N_{FP}, and N_{TN} represent the number of points assessed as TP, FP, and FN. The F-measure of the segmentation was 0.81, and this fact indicates that GP-HSMM can estimate boundaries reasonably. This is because GP-HSMM estimates the length of segments as well as the classes of segments.

Moreover, **Figure 8** shows that most false segmentations are in sequence 3. This is because "up and down" and "toe touch" motions are included only in sequence 3, and GP-HSMM was not able to extract patterns that occur infrequently. However, this problem is not limited to GP-HSMM, and it is generally difficult for any learning method to extract infrequent patterns. The Hamming distance, which was computed only from sequence 1 and sequence 2, was 0.15. This result shows that GP-HSMM can accurately estimate segments that appear multiple times in a sequence.

4.2. Segmentation of Karate Motion

We then applied our proposed method to more complex motion capture data, which consisted of the basic motions of karate (called kata in Japanese)[1] as shown in **Figure 10** from the motion capture library Mocapdata.com[2]. There are fixed motion patterns (punches or guards) in kata, and it is easy to form a ground truth for the segmentation. However, there might be shorter motion patterns, and GP-HSMM might be able to find those motion patterns if the number of classes is set to a larger number. Moreover, it is possible for GP-HSMM to discover patterns that cannot be labeled by humans, and GP-HSMM has the potential to analyze unlabeled time series data. However, in this experiment, we must evaluate the proposed method quantitatively, and fixed motion patterns (punches or guards) labeled by a human expert are used as ground truth. The type of kata we used was called heian 1, which is the most basic form of kata consisting of punches, lower guard, and upper guard (Tsuki, Gedanbarai, and Joudanuke in Japanese). **Figure 11** shows the basic movements used in heian 1. We divided this motion sequence into four parts, for use as four motion sequences to apply the blocked Gibbs sampler. Each motion sequence consisted of the following actions:

1. Left lower guard, right punch, right lower guard, and left punch.
2. Left lower guard, right upper guard, left upper guard, and right upper guard.
3. Left lower guard, right punch, right lower guard, and left punch.
4. Left lower guard, right punch, left punch, and right punch

By way of its preprocessing, as shown in **Figure 7**, the motion capture data was converted into motions with the body facing forward with a center of (0,0,0). To reduce computational cost, we downsampled the motion capture data from 30 frames per

[1] https://mocapdata.blob.core.windows.net/freemotions/karate.zip
[2] http://www.mocapdata.com/

second to 15 frames per second, and used two-dimensional left-hand positions (x_{lh}, y_{lh}) and right-hand positions (x_{rh}, y_{rh}) in the frontal view, as shown in **Figure 7**. To compare our method with others, we used segmentation based on HDP-HMM (Beal et al., 2001) and segmentation based on NPYLM and HDP-HMM (Taniguchi and Nagasaka, 2011), where NPYLM (Mochihashi et al., 2009) divides sequences discretized by HDP-HMM. In addition, we compared our method with BP-HMM (Fox et al., 2011) and AutoPlait (Matsubara et al., 2014).

Figure 12 shows the segmentation results. The horizontal axis represents the frame number, and the colors represent motion classes into which each segment was classified. The figure shows that HDP-HMM estimated shorter segments than the ground truth. This occurred because the emission distribution of HDP-HMM is a Gaussian distribution, which cannot represent continuous trajectories. Moreover, the result produced by segmentation, in which NPYLM divided sequences discretized by HDP-HMM, yielded longer segments. Moreover, NPYLM cannot extract fixed patterns of sequences. This is because the sequences discretized by HDP-HMM included noise and, therefore, NPYLM was unable to find a pattern in them. It was also difficult for BP-HMM to estimate correct segments, and some shorter segments were present. Further, AutoPlait could not find any segments in the karate motion sequences. We believe this occurred because HMMs are too simple to model complex motions. On the contrary, we use Gaussian processes that make it possible to model complex sequences. **Table 2** shows the segmentation accuracy of each method. We considered the estimated boundary to be correct if it was within true boundary ± five frames. The F-measure of the proposed method was 0.92, which indicates that GP-HSMM can estimate boundaries

TABLE 3 | Computational time of each method.

	Time (s)
GP-HSMM	248
HDP-HMM	1.99
HDP-HMM + NPYLM	18.2
BP-HMM	3.37
AutoPlait	0.31

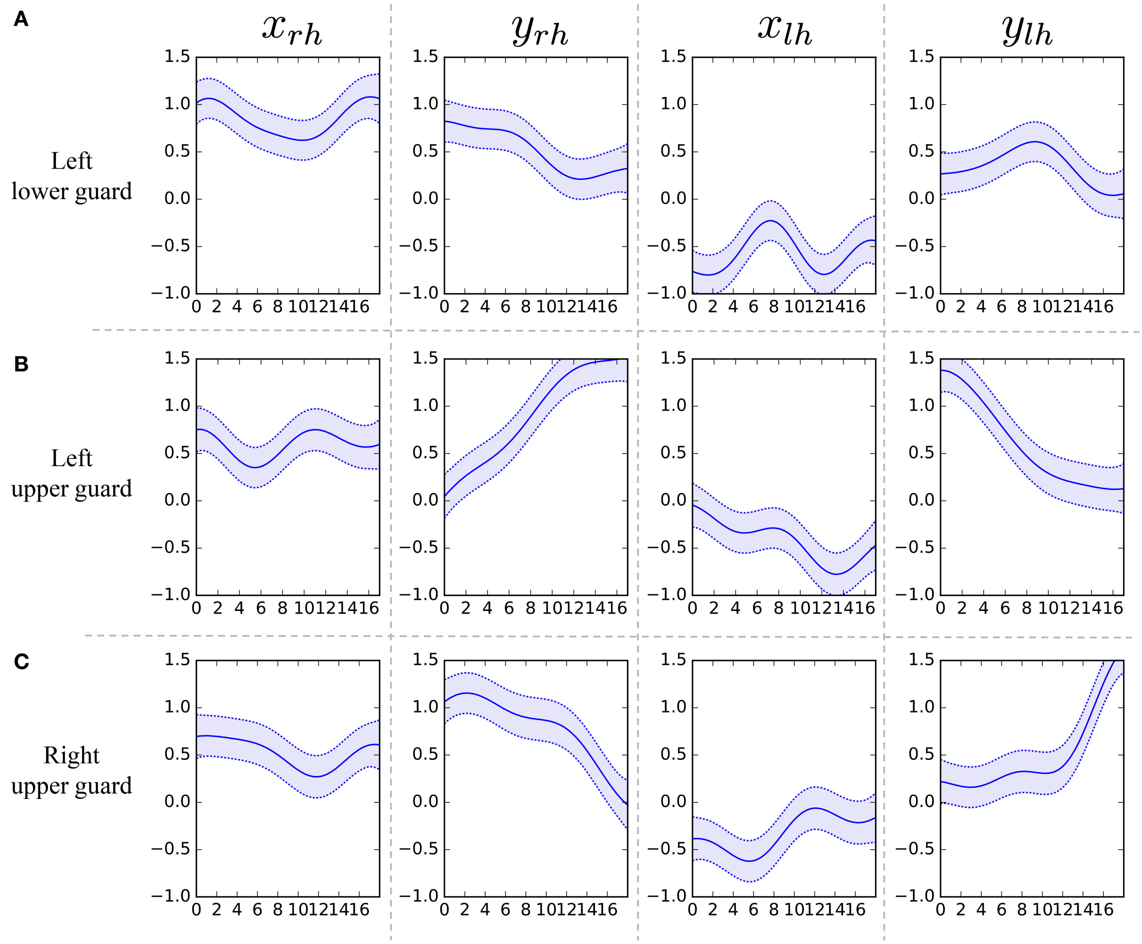


FIGURE 13 | Learned Gaussian processes for left lower guard, left upper guard, and right upper guard.

accurately. The results show that GP-HSMM outperforms the other methods. **Figure 13** shows the learned Gaussian process. y_{rh} in **Figure 13A**, which represents the height of the left hand, is decreased, which indicates the motion where the left hand is dropped for the lower guard. In contrast, y_{rh} in **Figure 13B** is increased, which indicates the motion where the left hand is raised for the upper guard. Conversely, y_{lh} in **Figure 13C** is increased for the right upper guard. From this result, we can see that characteristics of motions can be learned by Gaussian processes.

Moreover, the motions were classified into seven classes, although we set the number of classes to eight. This result indicates that the number of classes can be estimated to a certain extent, if a number closer to the correct number is given. However, a smaller number leads to under-segmentation and misclassification, and a much larger number leads to over-segmentation. This is a limitation of the current GP-HSMM, and we believe it can be solved by introducing a non-parametric Bayesian model.

Computational cost is another limitation of GP-HSMM. **Table 3** shows the computational time required to segment karate motion. HMM-based methods such as HDP-HMM, BP-HMM, and AutoPlait are relatively faster. In particular, AutoPlait is the fastest because it uses a single scan algorithm proposed in (Matsubara et al., 2014) to find boundaries, and it has been demonstrated that AutoPlait can detect meaningful patterns from large datasets. In contrast, our proposed GP-HSMM is much slower than other methods, and cannot process such large datasets. This is another limitation of the proposed method.

5. CONCLUSION

In this paper, we proposed a method for motion segmentation based on a hidden semi-Markov model (HSMM) with a Gaussian process (GP) emission distribution. By employing HSMM, segment classes and their lengths can be estimated. Moreover, a forward filtering-backward sampling algorithm is used to estimate the parameters of GP-HSMM; this makes it possible to efficiently search for all possible segment lengths and classes. The experimental results showed that the proposed method can accurately segment motion capture data. Although motions that occurred in the sequences a single time were difficult to segment correctly, motions that occurred a few times could be segmented with higher accuracy.

However, some issues remain in the current GP-HSMM. The most significant problem is that GP-HSMM requires the number of classes to be specified in advance. We believe this value can be estimated by utilizing a non-parametric Bayesian model. We are planning to introduce a stick-breaking process as a prior distribution of the transition matrix, and beam sampling for parameter estimation; these techniques are utilized in Beal et al. (2001). Another problem is computational cost. The computational cost to learn a Gaussian process is $O(n^3)$, where n denotes the number of data points classified in the GP. To overcome this problem, efficient computation methods have been proposed (Nguyen-Tuong et al., 2009; Okadome et al., 2014), and we will consider introducing these methods into GP-HSMM.

AUTHOR CONTRIBUTIONS

ToN, TaN, DM, IK, and HA conceived of the presented idea. ToN, TaN, and DM developed the theory and performed the computations. IK and HA verified the theory and the analytical methods. ToN wrote the manuscript with support from TaN and MK. IK and HA supervised the project. All authors discussed the results and contributed to the final manuscript.

ACKNOWLEDGMENTS

This work was supported by JST CREST Grant Number JPMJCR15E3 and JSPS KAKENHI Grant Number JP17K12758.

REFERENCES

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robot. Auton. Sys.* 57, 469–483. doi: 10.1016/j.robot.2008.10.024

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. (2001). "The infinite hidden markov model," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 577–584.

CMU (2009). *CMU Graphics Lab Motion Capture Database.* Available online at: http://mocap.cs.cmu.edu/

Fod, A., Matarić, M. J., and Jenkins, O. C. (2002). Automated derivation of primitives for movement classification. *Auton. Rob.* 12, 39–54. doi: 10.1023/A:1013254724861

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2007). *The Sticky hdp-hmm: Bayesian Nonparametric Hidden Markov Models with Persistent States.* Technical Report, MIT Laboratory for Information and Decision Systems.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2011). Joint modeling of multiple related time series via the beta process. *arXiv preprint arXiv:1111.4226.*

Goldwater, S. (2006). *Nonparametric Bayesian Models of Lexical Acquisition*. Ph.D. thesis: Brown University, Providence, RI.

Gräve, K., and Behnke, S. (2012). "Incremental action recognition and generalizing motion generation based on goal-directed features," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura), 751–757.

Lin, J. F.-S., Karg, M., and Kulić, D. (2016). Movement primitive segmentation for human motion modeling: A framework for analysis. *IEEE Trans. Hum. Mach. Sys.* 46, 325–339. doi: 10.1109/THMS.2015.2493536

Lin, J. F.-S., and Kulić, D. (2012). "Segmenting human motion for automated rehabilitation exercise analysis," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (San Diego, CA), 2881–2884.

Lioutikov, R., Neumann, G., Maeda, G., and Peters, J. (2015). "Probabilistic segmentation applied to an assembly task," in *IEEE-RAS International Conference on Humanoid Robots* (Seoul), 533–540.

Manschitz, S., Kober, J., Gienger, M., and Peters, J. (2015). Learning movement primitive attractor goals and sequential skills from kinesthetic demonstrations. *Robot. Auton. Sys.* 74, 97–107. doi: 10.1016/j.robot.2015.07.005

Matsubara, Y., Sakurai, Y., and Faloutsos, C. (2014). "Autoplait: utomatic mining of co-evolving time sequences," in *ACM SIGMOD International Conference on Management of Data* (Snowbird, UT), 193–204.

Mochihashi, D., Yamada, T., and Ueda, N. (2009). "Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling," in *Joint*

Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing Vol. 1 (Singapore), 100–108.

Nguyen-Tuong, D., Peters, J. R., and Seeger, M. (2009). "Local gaussian process regression for real time online model learning," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 1193–1200.

Okadome, Y., Urai, K., Nakamura, Y., Yomo, T., and Ishiguro, H. (2014). Adaptive lsh based on the particle swarm method with the attractor selection model for fast approximation of gaussian process regression. *Art. Life Robot.* 19, 220–226. doi: 10.1007/s10015-014-0161-1

Shiratori, T., Nakazawa, A., and Ikeuchi, K. (2004). "Detecting dance motion structure through music analysis," in *IEEE International Conference on Automatic Face and Gesture Recognition* (Seoul), 857–862.

Takano, W., and Nakamura, Y. (2016). Real-time unsupervised segmentation of human whole-body motion and its application to humanoid robot acquisition of motion symbols. *Robot. Auton. Sys.* 75, 260–272. doi: 10.1016/j.robot.2015.09.021

Taniguchi, T. and Nagasaka, S. (2011). "Double articulation analyzer for unsegmented human motion using pitman-yor language model and infinite hidden markov model," in *IEEE/SICE International Symposium on System Integration* (Kyoto), 250–255.

Uchiumi, K., Hiroshi, T., and Mochihashi, D. (2015). "Inducing Word and Part-of-Speech with Pitman-Yor Hidden Semi-Markov Models," in *Joint Conference of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing* (Beijing), 1774–1782.

Wachter, M., and Asfour, T. (2015). "Hierarchical segmentation of manipulation actions based on object relations and motion characteristics," in *International Conference on Advanced Robotics* (Istanbul), 549–556.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 22 December 2017
doi: 10.3389/fnbot.2017.00070

Representation Learning of Logic Words by an RNN: From Word Sequences to Robot Actions

*Tatsuro Yamada[1], Shingo Murata[2], Hiroaki Arie[2] and Tetsuya Ogata[1]**

[1] Department of Intermedia Art and Science, Waseda University, Tokyo, Japan, [2] Department of Modern Mechanical Engineering, Waseda University, Tokyo, Japan

An important characteristic of human language is compositionality. We can efficiently express a wide variety of real-world situations, events, and behaviors by compositionally constructing the meaning of a complex expression from a finite number of elements. Previous studies have analyzed how machine-learning models, particularly neural networks, can learn from experience to represent compositional relationships between language and robot actions with the aim of understanding the symbol grounding structure and achieving intelligent communicative agents. Such studies have mainly dealt with the words (nouns, adjectives, and verbs) that directly refer to real-world matters. In addition to these words, the current study deals with logic words, such as "not," "and," and "or" simultaneously. These words are not directly referring to the real world, but are logical operators that contribute to the construction of meaning in sentences. In human–robot communication, these words may be used often. The current study builds a recurrent neural network model with long short-term memory units and trains it to learn to translate sentences including logic words into robot actions. We investigate what kind of compositional representations, which mediate sentences and robot actions, emerge as the network's internal states via the learning process. Analysis after learning shows that referential words are merged with visual information and the robot's own current state, and the logical words are represented by the model in accordance with their functions as logical operators. Words such as "true," "false," and "not" work as non-linear transformations to encode orthogonal phrases into the same area in a memory cell state space. The word "and," which required a robot to lift up both its hands, worked as if it was a universal quantifier. The word "or," which required action generation that looked apparently random, was represented as an unstable space of the network's dynamical system.

Keywords: symbol grounding, neural network, human–robot interaction, logic words, language understanding, sequence-to-sequence learning

Edited by:
Alex Pitti,
Université de Cergy-Pontoise, France

Reviewed by:
Xavier Hinaut,
Inria Bordeaux—Sud-Ouest Research Centre, France
Cornelius Weber,
University of Hamburg, Germany

***Correspondence:**
Tetsuya Ogata
ogata@waseda.jp

Received: *18 August 2017*
Accepted: *14 December 2017*
Published: *22 December 2017*

Citation:
Yamada T, Murata S, Arie H and Ogata T (2017) Representation Learning of Logic Words by an RNN: From Word Sequences to Robot Actions. Front. Neurorobot. 11:70. doi: 10.3389/fnbot.2017.00070

1. INTRODUCTION

In recent years, the development of robots that work collaboratively in our living environment has attracted great attention. In many scenarios, these robots will be required to behave appropriately by understanding linguistic instruction from humans. Here, the meanings of instructions may change depending on the environment. Thus, robots must be able to flexibly adapt their behavior in accordance with the current situation or context. In the real world, no two events are identical; thus, a model that can generalize in order to translate an instruction to appropriate behavior even in novel situations is required. Specifying rules to define relations between language and behavior for various possible contexts becomes difficult and costs much more as task complexity increases. Therefore, to build a learning model that enables a robot to acquire generalizable relations from experience is especially desirable. To flexibly link language, which operates on discrete elements, to behavior, which operates within a continuous world, requires a solution to the symbol grounding problem (Harnad, 1990; Taniguchi et al., 2016).

One important characteristic of human language that enables us to describe even previously unseen situations is compositionality. In the field of formal semantics, the principle of compositionality (also referred to as Frege's principle) models a language system as follows: the meaning of a phrase or a sentence is given as a function of the meanings of its parts (e.g., words) (Partee, 2004). This principle means that the meaning of a complex expression is built from the meaning of its constituents and rules for combining them. Thanks to the compositionality of language and our cognitive ability to deal with it, humans can efficiently describe a wide variety of situations and dynamic events in the real world by compositionally constructing a complex expression from a finite number of elements. Investigating the compositional aspects of language deeply is important for understanding how human languages work in practice and for building intelligent communicative agents. Using the principle of compositionality as a base, formal semanticists attempt to build theoretical frameworks to explain the compositionality of natural language in a top-down manner.

In contrast with the top-down approach, there is a bottom-up approach that attempts to work from observation and analyze what kind of symbolic or compositional expressions emerge spontaneously through communicative tasks among humans, robots, and other intelligent agents (Steels and Kaplan, 1998; Steels and McIntyre, 1998; Steels, 2001; Kirby, 2002; Sasahara et al., 2007; Bleys et al., 2009; Schueller and Oudeyer, 2015; Spranger, 2015; Sukhbaatar et al., 2016; Wang et al., 2016; Havrylov and Titov, 2017; Lazaridou et al., 2017; Mordatch and Abbeel, 2017). In particular, in recent years, there have been many studies of multi-agent interaction, in which agents implemented with a deep learning model are developed in a mutually interactive manner and a compositional communication protocol emerges through the interaction. In Mordatch and Abbeel (2017), multiple agents situated within simulated 2D environments were given collaborative tasks in which agents had to symbolically communicate with each other to tell other agents their own goals. Before learning, symbols were meaningless. Being trained by reinforcement learning, the agents spontaneously gave the symbols shared meanings, which were sometimes interpretable by humans (e.g., "GO-TO," "LOOK-AT"), and they became able to communicate by combining the symbols, each of which was a token representing a subject, verb, or objective. In Havrylov and Titov (2017), two long short-term memory (LSTM) networks developed their own communication protocol to express the content of images. The sender network encoded the image information as a sentence expression, and the receiver network decoded the sentence and inferred which image among alternatives was described by the sentence. The analysis showed that a natural language-like coding such as hierarchy of categories or the importance of word order could be developed.

In the bottom-up approach, there has also been much research that trained neural network models by supervised learning (Sugita and Tani, 2005; Ogata et al., 2007; Sugita and Tani, 2008; Arie et al., 2010; Tuci et al., 2011; Chuang et al., 2012; Stramandinoli et al., 2012; Ogata and Okuno, 2013; Heinrich and Wermter, 2014; Heinrich et al., 2015; Hinaut et al., 2014; Yamada et al., 2015, 2016; Zhong et al., 2017). In these studies, the example sets of language and corresponding behavior were designed and prepared by humans in advance. These sets were used as ground truth during training, and after that, compositional representations intermediating between language and behavior were self-organized in their models. For example, Sugita and Tani (2005) and Arie et al. (2010) trained recurrent neural network (RNN) models (Elman, 1990) to learn relations between 2- or 3-word sentences and corresponding robot behavior. After training, representations corresponding to verbs and nouns were topologically self-organized as different components in the feature space binding language with robot behavior. These were construed as plausible materialization of linguistic compositionality by a dynamical system approach. Tuci et al. (2011) also conducted robot experiments using a feed-forward neural network and claimed that the compositional aspects that potentially exist in the behavior space are required for embedding robot behavior into compositional semantics via language. Heinrich et al. (2015) trained an RNN model to translate a robot's visual input into a corresponding sentence at the phoneme level. After training, the activated internal states of the RNN were more correlated with the type of word (color, shape, or position) than the phonemes. Hinoshita et al. (2011) visualized a similar kind of abstract encoding by a hierarchical RNN that was activated in accordance with the categories of words, even though they trained the RNN with linguistic sequences only. Investigating such representations organized in machine learning models is valuable, not only for understanding the compositionality of language but also for building interpretable intelligent systems.

The current study follows the supervised learning approach to the integration of language and behavior. In most previous studies of this type, mainly words that are directly grounded in real-world matters have been considered. For example, nouns (e.g., ball, box) or adjectives (e.g., red, tall) correspond to characteristics of objects. Verbs (e.g., hit, push) or adverbs (e.g., quickly, slowly) correspond to characteristics of motion. However, in our language, there are more abstract words (e.g., society, justice) that are not grounded in concrete physical objects or actions. To tackle the grounding of such words, Cangelosi et al. have conducted a series of language-robot experiments from the point of view of cognitive developmental robotics (Cangelosi et al., 2010; Chuang et al., 2012; Stramandinoli et al., 2012; Zhong et al., 2014; Stramandinoli et al., 2017). In these works, a robot implemented with a neural network develops its linguistic skill step by step, beginning by acquiring relations between simple basic motions and words (e.g., "push," "pull") directly grounded in them and moving on to achieving relationships between more abstract actions and words (e.g., "give," "reject") only indirectly grounded in them through connections to basic words.

However, the current study deals with another kind of abstraction. Language expressions in this study include grounded, in other words, referential[1] words and logic words, such as "not," "and," and "or". These logical words are not directly referring to the real world but act as logical operators in the construction of the meaning of the sentence. For example, just after you have closed a door, the commands "open the door!" and "do not close the door!" can express the same behavior OPEN-DOOR[2]. In another case, the appropriate behaviors in response to "bring A or B" include BRING-A and BRING-B. These logic words have not been addressed in conventional studies of integrative learning of language and behavior. In accordance with the formulation of formal semantics, even such non-referential words working as logical operators can be handled in a unified way. In fact, in cases of actual human–robot communication, it is highly likely that these words will be used.

The current study investigates what kind of structure representing compositional relations between language and robot actions is self-organized in the space of internal states of an RNN model trained through supervised learning. Here, our designed tasks include referential words and non-referential logic words. The meanings of sentences are constructed from both word types. We analyze how logic words are processed and how their functions are represented by the RNN dynamics along with the referential words. More precisely, we apply the sequence-to-sequence learning method that has recently attracted great attention in the field of natural language processing (Sutskever et al., 2014; Bahdanau et al., 2015; Vinyals and Le, 2015; Wu et al., 2016) to the translation from sentences to robot actions and analyze representations by visualizing internal states during interactions that occur after training.

This paper is organized as follows. In section 2, we introduce the learning model. In section 3, we give the results of the learning experiment for the first task and analyze the representations acquired by the learning model in detail. In section 4, we report the results of the second task. In section 5, we discuss the results and then conclude this study.

2. LEARNING MODEL

2.1. Problem Formulation

The aim of the current study is to investigate how the compositional relations between language and robot actions are developed and represented internally by the model from direct experiences of interaction. Therefore, we define the interactive instruction–action task as a simple problem, learning to predict a robot's joint angles appropriate to the current situation. At each discrete time step t a neural network model receives a word $\boldsymbol{w}_t$, visual information $\boldsymbol{v}_t$, and the robot's current joint angle configuration $\boldsymbol{j}_t$. An instruction sentence is given as a concatenation of some words, thus it takes some time steps. At each time step the model generates its prediction $\boldsymbol{j}_{t+1}$ based on the input history $\boldsymbol{w}_{0:t}$, $\boldsymbol{v}_{0:t}$, and $\boldsymbol{j}_{0:t}$. During the instruction phase the appropriate prediction would be just keeping the current posture $\boldsymbol{j}_t$. After an instruction is given, an appropriate prediction should be the generation of angles different from the current ones. An action corresponding to the instruction must also be generated as a sequence of joint angle configurations over several time steps. In our tasks, the appropriate action sequence after an instruction is determined by the combination of the instruction sequence, the visual information given simultaneously with the sentence, and the robot's current posture.

2.2. Model Architecture and Forward Dynamics

In this study, as a model that learns the aforementioned problem, we use an RNN with an LSTM layer (Hochreiter and Schmidhuber, 1997). The model is a three-layer neural network whose middle layer is the LSTM layer, as shown in **Figure 1**. All the LSTM units have a peephole connection (Gers and Schmidhuber, 2000). At each time step, the model receives $\boldsymbol{w}_t$, $\boldsymbol{v}_t$, and $\boldsymbol{j}_t$. The LSTM layer calculates the current output $\boldsymbol{h}_t$ from these external inputs, the memory cell state in the previous step $\boldsymbol{c}_{t-1}$, and its own output in the previous step $\boldsymbol{h}_{t-1}$:

$$\boldsymbol{h}_t = \mathrm{LSTM}(\boldsymbol{w}_t, \boldsymbol{v}_t, \boldsymbol{j}_t, \boldsymbol{h}_{t-1}, \boldsymbol{c}_{t-1}; \boldsymbol{\theta}), \tag{1}$$

where $\boldsymbol{\theta}$ denotes the parameters of the LSTM layer. In this process, $\boldsymbol{c}_{t-1}$ is also updated to $\boldsymbol{c}_t$. The output layer is a fully connected layer. It receives the output of the LSTM layer and predicts the appropriate joint angles for the next time step, denoting these $\hat{\boldsymbol{j}}_{t+1}$. We denote the model prediction by $\boldsymbol{j}_{t+1}$:

$$\boldsymbol{j}_{t+1} = \tanh(\boldsymbol{W}\boldsymbol{h}_t + \boldsymbol{b}), \tag{2}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are a learnable weight matrix and a bias vector, respectively. The model prediction is also used as the joint angle input at the next time step. In this process, receiving an instruction and generating an action are completely conducted in the forward-propagation algorithm. An instruction sentence, visual information, and the robot's current posture are encoded as the states of memory cells in the LSTM layer. After receiving the instruction, a corresponding action sequence is generated by decoding the integrated information.

The working after training seems to be similar to the normal sequence-to-sequence models that have recently been used in the field of natural language processing for tasks such as question answering and translation. However, the current model is different in that it has only one LSTM layer; in other words, it does not separate the decoder from the encoder.

[1] In this paper, we use the term "referential" instead of "grounded" for the following reason. We conduct two robot experiments in the following sections, but the first task is numerically simulated on a computer. Even though the second task uses a real robot, the visual input is still highly preprocessed. Strictly speaking, we do not deal with the symbol grounding problem in accordance with the definition by Harnad (1990). To prevent misunderstanding, we use the term "referential," and sometimes "linking" to express that a word has a referent or a corresponding feature in other sensorimotor modalities.

[2] In this paper, we denote specific actions or behaviors executed by agents with capital letters.

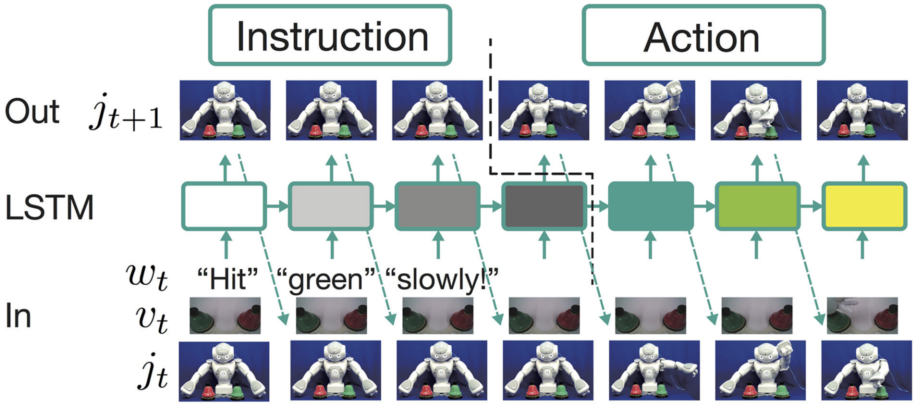


FIGURE 1 | The framework employed to learn the current tasks. The learning model is a three-layer neural network whose middle layer is an LSTM layer. At each time step, the model receives word $\boldsymbol{w}_t$, visual information $\boldsymbol{v}_t$, and the current robot joint angles $\boldsymbol{j}_t$. The LSTM layer calculates the current output $\boldsymbol{h}_t$ from these external inputs, the memory cell state $\boldsymbol{c}_t$, and its own output in the previous step $\boldsymbol{h}_{t-1}$. The output layer is a fully connected layer. It receives the output of the LSTM layer and predicts the appropriate joint angles for the next time step. In this process, receiving an instruction and generating an action are completely conducted in the forward propagation algorithm. An instruction sentence, visual information, and the robot's current posture are encoded as the states of memory cells in the LSTM layer. After receiving the instruction, a corresponding action sequence is generated by decoding the integrated information.

Moreover, the algorithm does not explicitly switch between the instruction and action phases. As visually illustrated in **Figure 3** in the next section, the relations between instructions and corresponding actions are experienced entirely in the sequential data that represent human robot interaction, which consists of repeated iteration of instructions and actions. With such data, as mentioned above, the model learns to predict only the robot's joint angles appropriate for the next time step in the current situation. Because both phases are only implicitly included in the sequential data, the model has to learn to switch phases without a priori knowledge. In more precise terms, the contrasting functions of encoding and decoding (i.e., instruction receiving and action generation) emerge as an apparent phenomenon as a result of learning alone. The model continues to predict the joint angles even during receipt of an instruction, while the target is keeping the current posture. In contrast, zero-filled vectors are continuously received as language inputs even when the robot is generating an action sequence. Although no external algorithms or explicit signals on the network I/O for phase switching exist, the trained model behaves as though it flexibly switched phases. For more discussion from the point of view of dynamical systems, refer to Yamada et al. (2016).

2.3. Training

To train the model, supervised learning is conducted by minimizing the squared error between the model's output $\boldsymbol{j}_{t+1}$ and the correct joint angles at the next time step $\hat{\boldsymbol{j}}_{t+1}$: that is, the model is trained to minimize

$$\boldsymbol{E} = \sum_s \sum_t (\boldsymbol{j}_{t+1} - \hat{\boldsymbol{j}}_{t+1})^2, \tag{3}$$

where s is the index of a sequence. The error at each time step is back-propagated to the initial time step without truncation by using the back propagation through time algorithm (Rumelhart et al., 1986). In our tasks, sometimes there are multiple correct actions. For example, if the instruction is "hit red or blue," both HIT-RED and HIT-BLUE can be correct. In such cases, one action is chosen randomly each time and given as the correct response.

In the following sections, we describe learning experiments conducted using the model described in this section. We designed two tasks, the "flag task" and the "bell task," in which a robot is required to generate an action in response to linguistic instructions that sometimes include logic words. Although the former task is numerically simulated on a computer from data preparation to evaluation, it is interpretable as a task for a robot. In contrast, the latter task collects motion data by using a real robot; it is, therefore, a more complicated task.

3. EXPERIMENT 1: FLAG TASK

3.1. Task Overview

In this section, we first report the learning results of the first task, the "flag task". Although this task is completely performed in a computer simulation, we describe the task as if it was undertaken by a robot so that it is easy to imagine intuitively. First, a human makes the robot grasp flags colored red, green, or blue, one in the left hand and another in the right, at random. After that, the human gives the robot a linguistic instruction. The sentence consists of a combination of an objective ("red," "green," "blue"), a verb ["up" (i.e., lift), "down" (i.e., lower)], and a truth value ("true," "false"). Note that the words are given in this order because this game was designed by modifying a popular children's game in Japan. Japanese is a subject-object-verb language (cf., English, which is a subject-verb-object language), therefore a verb follows an objective word, and a truth value, which is one of the auxiliary verbs, follows a verb. Here, the objective color word indicates the arm that is grasping a flag of the stated color. The verb determines whether the flag should be raised or lowered. Finally, if the truth value is "true," the robot must behave as indicated by the preceding verb. In contrast, if "false," it must generate the opposite action. For example, if the robot receives an instruction "red up false" when it is grasping a red flag in the right arm, the correct action is to lower the right arm (R-DOWN). In other words, "true" and "false" roughly represent "do" and "do not," respectively.

In the objective part, two color words can be concatenated by "and" (referred to as AND-concatenated). For example, if the robot receives the instruction "red and blue up true" when it is grasping red and blue flags, the robot must lift up both arms. There are also cases in which two color words are concatenated by "or" (referred to as OR-concatenated). For example, if the robot receives the instruction "green or blue up false" when it is grasping the green and blue flags, the correct action is to lower either arm. However, if at least one arm is already in the DOWN posture, the robot must keep the current posture. The number of possible goal-oriented actions is six: L-UP, R-UP, B-UP, L-DOWN, R-DOWN, and B-DOWN, where L, R, and B mean left, right, and both, respectively. However, there are situations in which, even though the same goal-oriented action is required, the actual motion that should be generated by the robot varies according to the robot's current posture (shown as arrows in **Figure 2**). Note that there are even cases in which the robot should not move either of its arms. The number of possible situations, based on the combination of flag colors (6 patterns), instructions (24 patterns), and the robot's waiting posture (4 patterns), is 576. In this task, instructions inconsistent with the flag colors are never given. For example, if the colors of the flags held by the robot are red and blue, the instruction "green up true" is never given. Furthermore, cases in which both flags are the same color are not permitted.

The requirements imposed on the robot in this game are analyzed as follows. (1) First, the arm indicated by the color words depends on the arm with which the robot holds the flag. In other words, referring to an external situation is required. (2) The actual motion trajectory to be generated depends on the robot's current posture. For example, suppose the robot is required to generate L-UP action. If the robot's left arm is in the DOWN posture, the robot has to lift its left arm. However, if the robot's left arm is already in the UP posture, the robot has to maintain its posture. (3) Finally, the RNN has to deal not only with referential words (e.g., verb, objective) but also logic words such as "true," "false," "and," and "or," which we focus on in the current study. Due to this task setting, in extreme cases, sentences completely orthogonal to each other can indicate the same action (e.g., "red up true" with the red flag in the left arm and "blue down false" with the blue flag in the left arm). In contrast, some OR-concatenated sentences have an ambiguity that allows the robot multiple choices even in the same situation.

3.2. Data Representation

We represent the execution of the flag task as a sequence of 14-dimensional vectors. The state $\boldsymbol{S}_t$ at time step t is represented as follows:

$$\boldsymbol{j}_t = [j_l^{(t)}, j_r^{(t)}], \tag{4}$$

$$\boldsymbol{v}_t = [v_r^{(t)}, v_g^{(t)}, v_b^{(t)}], \tag{5}$$

$$\boldsymbol{w}_t = [w_0^{(t)}, w_1^{(t)}, w_2^{(t)}, w_3^{(t)}, w_4^{(t)}, w_5^{(t)}, w_6^{(t)}, w_7^{(t)}, w_8^{(t)}], \tag{6}$$

$$\boldsymbol{S}_t = [\boldsymbol{j}_t; \boldsymbol{v}_t; \boldsymbol{w}_t]. \tag{7}$$

Regarding the robot joints, only the left and right shoulder pitches $(j_l^{(t)}, j_r^{(t)})$ are used. The permissible range of each shoulder pitch is scaled in the interval $[-1.0, 1.0]$. The UP posture corresponds to a pitch of 0.8, and the DOWN posture corresponds to a pitch of -0.8. Posture changes from UP to DOWN or from DOWN to UP after receiving an instruction are completed over 6 time steps. Visual information is represented in 3 dimensions $(v_r^{(t)}, v_g^{(t)}, v_b^{(t)})$. The three components correspond to the R, G, and B channels, respectively. If the color is grasped by the left hand, the component is set to 0.8; if it is in the right hand, the component is -0.8; and if not grasped by either hand, the component is 0.0. Nine elements are assigned for language. Each element corresponds to one word, out of "red," "green," "blue," "up," "down," "true," "false," "and," and "or," and an instruction sentence is represented as a sequence of one-hot vectors, which have the value of 0.8 at one element and 0.0 at the other element. In this study, the data representing the flag task are completely generated on a computer without using a real robot. Example interaction data are shown in **Figure 3**. Note that we added a small amount of Gaussian noise (mean: 0.00; standard deviation: 0.02) to the values of joint angles. In the preliminary experiment, we first trained the model without noise and got poor results. We then added noise and the results improved. We discuss this effect in section 5.

3.3. Learning Setting and Evaluation Method

We made 2,048 sequential datasets for training, each of which includes 10 episodes. The term, "episode" denotes a chunk consisting of an instruction and an action response. The situations included in each sequence were randomly ordered. All 576 possible situations were included at least once. We built five models with 50, 70, 100, 150, and 300 LSTM units and trained them 10 times from randomly initialized learnable parameters. We also trained the 100-node model with data without noise applied to the joint angles. Adam, a version of the stochastic gradient descent algorithm made stable by computing individual adaptive learning rates for each parameter, is used as an optimizer (for details, refer to Kingma and Ba, 2015). The number of learning iterations is 10,000, and the learning rate is set to 0.001. We coded our model within Python using the Chainer (https://chainer.org) framework. The source code of our model is available at https://github.com/ogata-lab/RNN_FNR2017.

After learning, we made another dataset for the evaluation. This dataset includes all the possible situations 10 times each. Although the situations were randomly ordered, the order was different from the training dataset. When the errors between

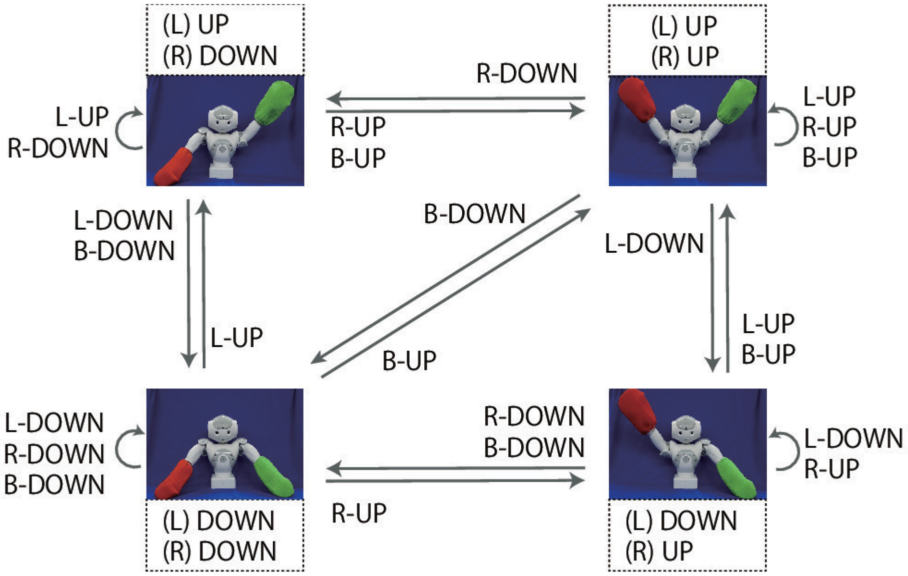


FIGURE 2 | Overview of the flag task. The experimenter makes the robot grasp two colored flags. Instructions are given as sentences in the form of an objective ("red," "green," "blue"), a verb ["up" (i.e., lift), "down" (i.e., lower)] and a truth value ("true," "false"). The robot must generate one of six goal-oriented actions (L-UP, L-DOWN, R-UP, R-DOWN, B-UP, B-DOWN) in accordance with the instruction. In the objective parts, two color words can be concatenated by "and". In this case, the robot must generate B-UP (B-DOWN) action. Two color words also can be concatenated by "or," in which case the robot must move either arm. The actual movements corresponding to these goal-oriented actions for each starting posture are indicated by the arrows in this figure.

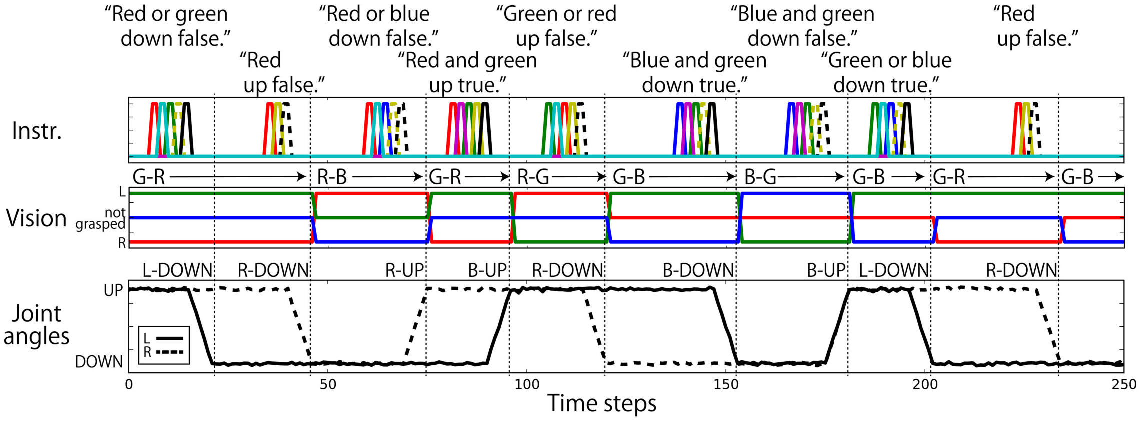


FIGURE 3 | An example sequence that represents the flag task. Each vertical broken line indicates the end of an episode. **(Top)** An instruction is given as a succession of words, which are each represented as a 1-hot vector. In the waiting and action-generation phases, zero-filled vectors are given. **(Middle)** Visual information is continuously given as a sequence of three-element (R, G, B) vectors. The flag colors can be changed randomly just after action generation. Because this task was numerically simulated on a computer, changes in flags were represented as instantaneous changes in values. Note that flags are sometimes not changed as in the case from the first episode to the second episode in this figure. **(Bottom)** Each action immediately follows an instruction.

the generated postures of both arms six steps after receiving an instruction and the correct ones are less than 0.04, we judge that the RNN has succeeded in generating an appropriate action. Here, there are cases in which the correct action cannot be determined uniquely. In such cases, if the RNN succeeds in generating any of the correct actions, we judge that as success. We regard the situation patterns in which the RNN succeeds in generating an appropriate action more than seven times out of 10 as "appropriately learned". Note that in the current task, the sequences are given to the robot as multiple repetitions of the instructions and corresponding actions. Therefore, even if situations that are defined by combination of an instruction, the vision, and the robot posture are the same, slightly different activations are gained every time because the contextual information of the previous episode remains in the memory cell states. Thus, the generated action is not identical among trials.

3.4. Task Performance after Training

We classify all possible situations into four types. (1) Situations in which the instruction includes only one objective word (192 situations). (2) Situations in which the instruction is AND-concatenated (192 situations). (3) Situations in which the instruction is OR-concatenated, but there is only one correct action. For example, when the instruction is "red or blue up true" and the both arms are already in the UP position, the only correct action is to maintain the UP-UP posture (144 situations). (4) Situations in which the instruction is OR-concatenated, and two correct actions exist (48 situations). We evaluate performance by counting how many situation patterns each model learns appropriately with respect to each of the four types. **Figure 4** shows the result. Most situations in types (1), (2), and (3), in which the correct action is uniquely determined, were appropriately learned by all the models. However, the 100-node model trained with data without noise applied to the joints could not learn sufficiently well. For type (4), in which the correct action cannot be determined uniquely, a clear difference exists between models: the number of appropriately learned situations increased in accordance with the number of LSTM nodes. The model without noise also performed worse than the 100-node model with noise. **Figure 5** shows an actual example of interaction achieved by the 300-node model. It can be seen that the RNN generates an appropriate action immediately after receiving an instruction in each episode.

Next, we checked which arm was actually moved in situations of type (4). If the model learned the type (4) situations just as a left-arm action or just as a right-arm action, the meaning of "or" cannot be regarded as being truly learned, although the aforementioned evaluation criteria is fulfilled. Here, we investigated the results for a model with 300 LSTM units. In 45.4% of the trials, the left hand was moved. In 52.5%, the right hand was moved. In 2.1%, neither movement could be generated successfully. Overall, the arms were quite evenly chosen in these situations. There are 48 situation patterns of type (4), and the test was conducted 10 times for each of them. In all cases, the RNN sometimes chose to move the left arm and other times chose to move the right arm. In other words, the RNN could learn the meaning of OR-concatenated instructions appropriately as "OR". Thus, the flag task was performed sufficiently well by the trained models.

3.5. Analyses of Internal Representations

In the previous subsection, we confirmed that the RNN could learn to execute the flag task. In this section, to analyze how the RNN internally represents the relations between instructions and sensorimotor information, we visualized the internal states

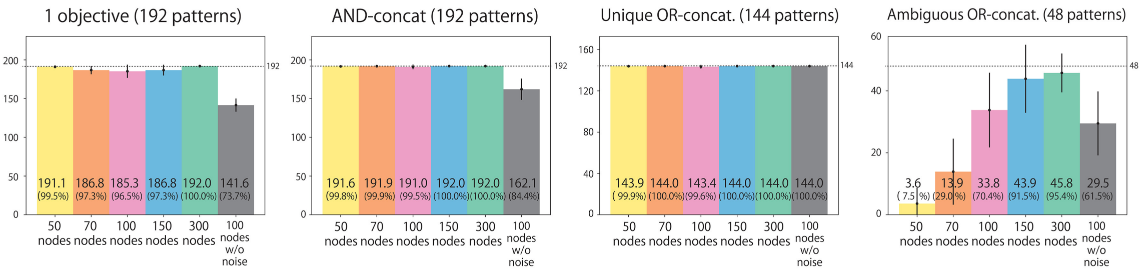


FIGURE 4 | Experiment 1 (flag task). Action generation performance. We evaluated performance by counting how many situation patterns each model learned appropriately with respect to each of the four situation types: (1) the instruction includes one objective; (2) the instruction is AND-concatenated; (3) the instruction is OR-concatenated, but there is only one correct action; and (4) the instruction is OR-concatenated, and two correct actions exist. Note that the written values are averages of 10 trials in which learning began with different seeds. Error bars represent standard deviations.

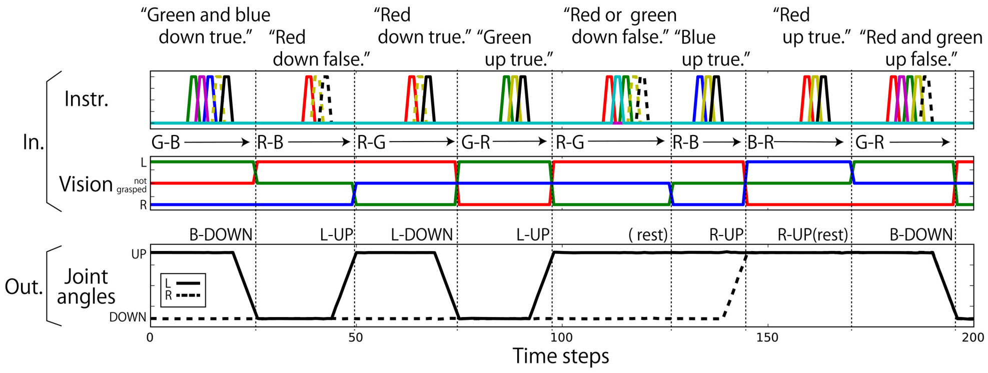


FIGURE 5 | An example of the resulting interaction in the flag task. The 300-node model could generate an appropriate action in almost all situations.

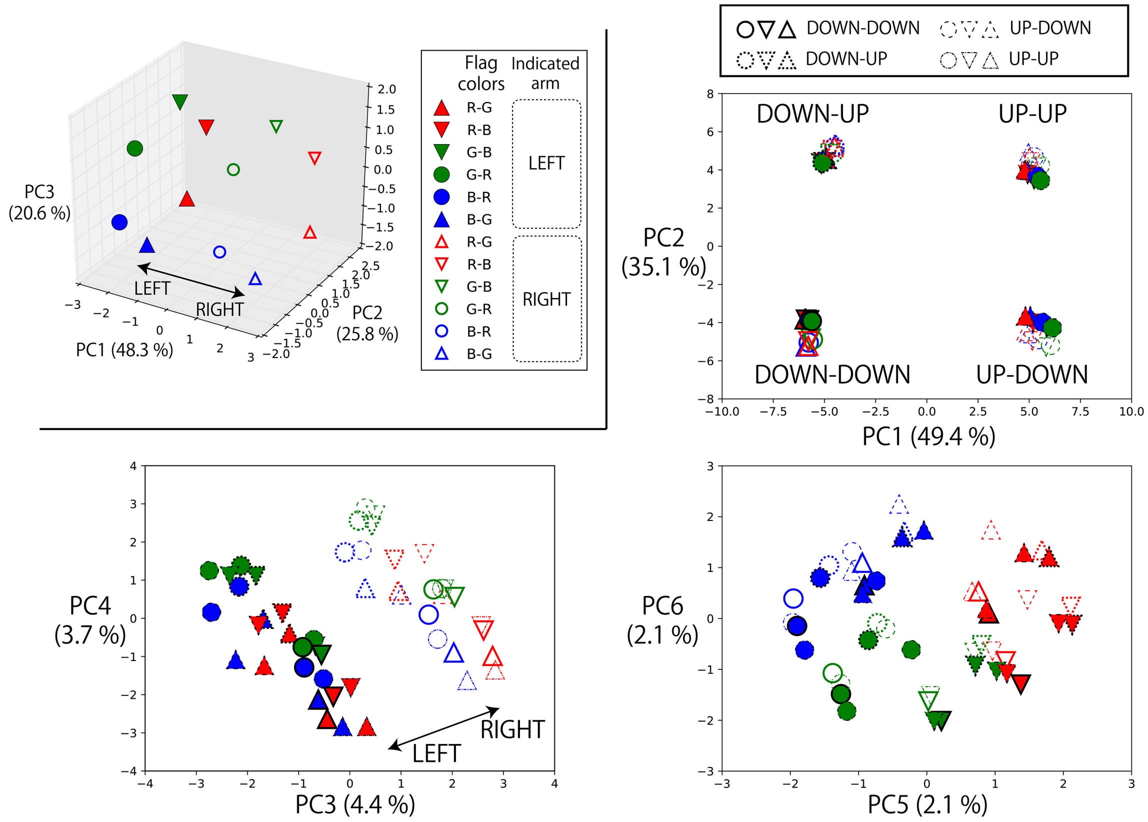


FIGURE 6 | Top left: The states of the memory cells after the instruction "(L-flag color word) up true" or "(R-flag color word) up true" is given to the robot projected onto the space spanned by PC1, 2, and 3. Here, the robot is always waiting in the DOWN-DOWN posture, but the situations are different with respect to the colors of the flags grasped in each hand. For example, the filled blue circle is the activation after receiving "blue up true" in the situation B-R in which a blue flag is in the left hand and a red flag is in the right. In this task, which arm should be moved cannot be determined from the given objective word alone. However, in the PC1 direction, which arm is indicated by the objective word is represented. The RNN learned to integrate the objective word information and the current visual information, and acquired a representation corresponding to the meaningful pair of "left–right". By using these activations, the robot could choose a correct arm for each trial. Others: We also plotted the internal states after giving these instructions to the robot that is waiting in the other postures, together with the internal states on the DOWN-DOWN condition. We projected them onto the PC1–2, PC3–4, and PC5–6 space. Note that we carried out PCA again by using the internal states on all of these conditions. Plot colors and shapes are as in the top left panel except that the frame lines differ according to the robot current posture. In this case, the current posture information is strongly reflected to the internal states, thus it is encoded in the PC1–2 plane. But the representation corresponding to "left" and "right" is still able to be seen easily in the PC3–4 plane. The visual information was encoded in the PC5–6 space although the hexagon shape was a little distorted.

during the execution of the task by principal component analysis (PCA)[3].

3.5.1. Representations of Referential Color Words

First, the top left panel of **Figure 6** shows the states of the memory cells after the instruction "(L-flag color word) up true" or "(R-flag color word) up true" is given to the robot. Here, the robot is always waiting in the DOWN-DOWN posture, but the situations are different with respect to the flag colors. Therefore, the RNN has to choose which arm should be raised by integrating the visual information and the input objective word. In the PC2–PC3 plane, the current visual input is directly embedded. However, in the PC1 direction, which arm has been indicated by an objective word is represented. In other words, in the experience of generating action sequences by receiving an instruction and visual input, the RNN acquired a representation corresponding to the meaningful pair of "left" and "right". We also plotted the internal states after giving these instructions to the robot that is waiting in the other postures, together with the internal states on the DOWN-DOWN condition. In the other three panels of **Figure 6**, we projected them onto the PC1–2, PC3–4, and PC5–6 space. In this case, the current posture information is strongly reflected to the internal states, thus it is encoded in the PC1–2 plane. But the representation corresponding to "left" and "right" is still able to be seen easily in the PC3–4 plane. Here, note that in the case of the UP-UP posture, the actual motions to be generated by receiving "(L-flag color word) up true" or "(R-flag color word) up true" are the same (keep the current posture), and, in fact, the network could keep the posture. This analysis shows that even in such situations in which the same action was generated, the model could internally represent these instructions

[3] Before applying PCA, parallel translation was applied to the internal state vectors to make the mean of them the zero vector (i.e., centering preprocessing was performed).

as different meanings, “left” or “right”. Incidentally, the visual information was also still encoded in a less principal component space (PC5–6) although the hexagon shape was a little distorted.

3.5.2. Representations of Logic Words: “True” and “False”

Next, we also analyzed the representations of logic words. We visualized memory cell activations after giving eight possible instructions with one objective word to a robot that was grasping R-B flags and waiting in the DOWN-DOWN posture (**Figure 7**). In the directions of PC1, PC2, and PC3, the activations directly corresponding to each part of speech (objective, verb, truth value) of the input sentence can be seen, that is, “red”/“blue”, “up”/“down” and “true”/“false” pairs are reflected in the PC2, PC1, and PC3 axes, respectively. Here, the problem is that the RNN has to solve an X-OR problem that consists of “up”/“down” and “true”/“false” (shown in the left panel of **Figure 7**), and to link its interpretation into UP or DOWN goal-oriented action. More precisely, if the sentence includes “up true” or “down false,” UP action must be chosen. In contrast, if the sentence includes “up false” or “down true,” DOWN action must be chosen.

Actually, by exploring the lower-rank component PC4, the activations that were located diagonally across the parallelogram in PC1–PC3 space were located in the same direction. “Up true” and “down false,” which are mutually orthogonal but have the same meaning UP, are represented in the bottom area of the right panel. In contrast, “up false” and “down true” are represented in the top area. Thanks to this non-linear embedding, the X-OR problem is solved in the PC4 direction. In summary, the RNN has extracted the XOR problem implicitly included in the sequential experiences and learned to link the orthogonal instructions in the same goal-oriented action by its non-linear dynamics, while retaining the information that the input sentences are very different from each other in the larger principal components.

3.5.3. Representations of Logic Words: “And” and “Or”

The left panel of **Figure 8** shows the memory cell states after giving a robot that is grasping R-B flags some instructions whose objective part is one word, AND-concatenated, or OR-concatenated. The verb and the truth value are “up” and “true,” respectively. AND-concatenated instructions that direct the robot to raise both arms are represented away from other instruction encodings in the PC1 direction. The pair of “red” and “blue” is represented in the PC2 direction. Here, the word “or” that directs the robot to raise either hand is embedded in the middle space between these two encodings. This suggests that “or” is represented as an unstable point of the network dynamics and that, thanks to this acquired dynamics, behavior which apparently looks like randomly choosing the left or right arm has emerged.

To verify this, we conducted the following additional simulation. To a robot that had 2,048 different contexts, we gave the instruction “green or blue up true.” Specifically, in all 2,048 contexts, a robot is currently waiting in a DOWN-DOWN posture with G-B flags. However, in each context, the order of preceding episodes is randomly different from in the other contexts. As mentioned in section 3.3, even when the situation, defined by the combination of an instruction, the vision, and the robot current posture (in this simulation, “green or blue up true,” the green flag in the left hand, the blue flag in the right hand, and DOWN-DOWN posture, respectively) is the same, different activations occur every time because the contextual information of the previous episodes still remains in the memory cell states. Therefore, we see 2,048 different activations corresponding to 2,048 contexts. As shown in the top left panel of the right side of **Figure 8**, the memory cell states after the instruction “green or blue up true” were then arranged in an arch-shaped space. Each point corresponds to one specific context. When the activation was on the left side of the arch, the robot generated L-UP action.

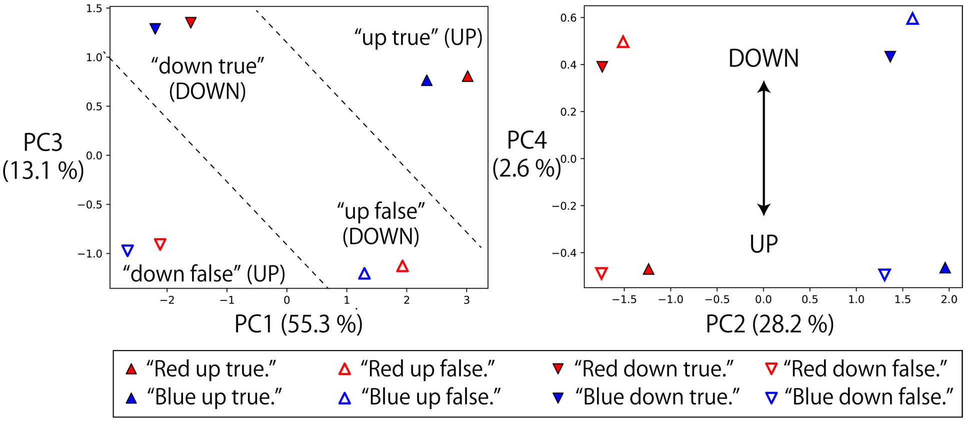


FIGURE 7 | Memory cell states after giving eight possible instructions with one objective word to the RNN in the situation that the grasped flags are R-B and the waiting posture is DOWN-DOWN. The left panel projects them onto PC1–3 space, and the right panel projects them onto PC2–4 space. In the directions of PC1, PC2, and PC3, the activations directly corresponding to each part of speech (objective, verb, truth value) can be seen: that is, “red”/“blue”, “up”/“down” and “true”/“false” pairs are reflected in the PC2, PC1, and PC3 axes, respectively. However, by exploring lower rank components, it can be seen that the X-OR problem consisting of “up”/“down” and “true”/“false” pairs is solved in the PC4 direction by non-linearly embedding the input sentences.

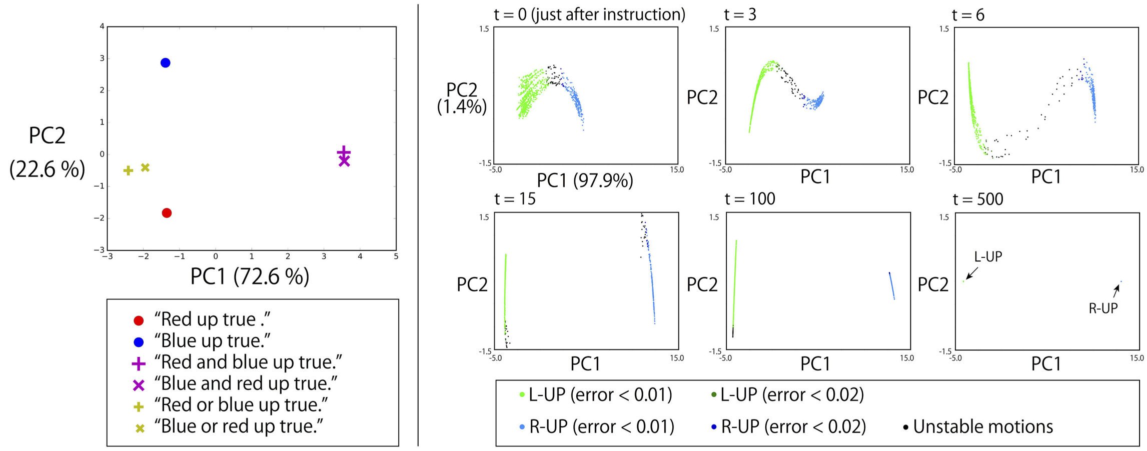


FIGURE 8 | Left: The memory cell states after giving a robot that is grasping red and blue flags some instructions whose objective part is one word, AND-concatenated, or OR-concatenated. The verb and the truth value are "up" and "true," respectively. The AND-concatenated instructions are represented away from other instruction encodings in the PC1 direction. The pair of "red" and "blue" is represented in the PC2 direction. The "or" that directs the robot to raise either hand is embedded in the middle space between these two encodings. **Right**: To a robot waiting in the DOWN-DOWN posture with G-B flags after 2,048 different contexts, we gave the instruction "green or blue up true." The memory cell states after the instruction ($t = 0$) were arranged on an arch-shaped space **(left top)**. Each point corresponds to one specific context. When the activation was on the left side of the arch, the robot generated L-UP action and the internal states converged to the fixed-point corresponding to the UP-DOWN posture. In contrast, on the right side, the robot generated R-UP action, and the internal states converged to the fixed-point corresponding to the DOWN-UP posture. When the activation was on the topmost area of the arch, a little unstable action was generated. However, even in such cases, the internal states eventually converged to one of fixed-points, as shown in the right bottom panel.

In contrast, for right-side activation, the robot generated R-UP action. When the activation was in the topmost area of the arch, some unstable motion was generated. However, in all cases, the internal states eventually converged into one of the fixed-point attractors that corresponded to the DOWN-UP posture or the UP-DOWN posture, as shown in the bottom rightmost panel of **Figure 8**. This means that to respond to OR instructions that require the robot to behave in a random exclusive-OR-like way, the internal representation was the convergence from an unstable space to either one of two stable points.

In this analysis, PC1 was strongly dominant (the contribution ratio is 97.9%). Therefore, due to this important contribution ratio, one could assume that only one neuron would be enough to generate this unstable dynamics. However, the activation in the PC1 direction was actually composed of the activations of multiple units. Specifically, no single unit has cosine similarity of more than 0.4 (or less than −0.4) to PC1. Instead, seven units have cosine similarity of in the range between 0.2 and 0.4 (or between −0.4 and −0.2) to PC1. In other words, this unstable dynamics was realized in a distributed way.

3.5.4. Dynamical Representations of the Task Execution

Finally, we visualized the internal dynamics during the execution of the task. **Figure 9** shows the state transition of memory cells while the robot experienced four episodes and its posture is moved in the order from DOWN-DOWN, through UP-DOWN, UP-UP, DOWN-UP, to DOWN-DOWN. Here, the PC1-2 space seems to roughly correspond to the robot's posture. Moreover, the transitions among different postures are represented as transitions among different fixed-point attractors (shown as circles), each of which corresponds to a posture. By receiving an instruction, the internal state is activated in the PC3 direction and reaches the unstable point indicated by a + mark. This activation is gained as a result of the integration of the visual

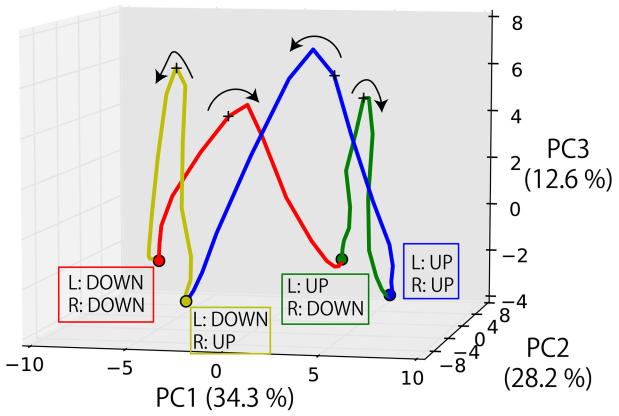


FIGURE 9 | The state transition of memory cells while the robot experienced four episodes and its posture moved in the order from DOWN-DOWN, through UP-DOWN, UP-UP, DOWN-UP, to DOWN-DOWN. The transitions among different postures are represented as transitions among different fixed-point attractors (circle marks), each of which corresponds to a posture. By receiving an instruction, the internal state is activated in the PC3 direction and reaches the unstable point indicated by a + mark. By converging into one of the fixed-points again after the activation, the corresponding goal-oriented action is generated. The robot then waits for a subsequent instruction at that point.

information and processing logic words, as mentioned above, although it is difficult to visualize them simultaneously in this figure. By converging to one of the fixed-points again after the activation, the corresponding goal-oriented action is generated. The robot then waits for a subsequent instruction at that point. This is the case even when the correct action is to maintain the current posture. While the apparent motion of joint angles is remaining stationary, it was internally represented as converging to the original fixed-point.

In summary, the RNN learned to encode the instructions in a form integrated with the visual inputs and the current robot posture and to generate an appropriate robot action through the experience of sequential interaction data. It was also revealed that logical words, "true," "false," "and," "or" are processed along with the other referential words and encoded in a way that reflects the functions in the current task.

3.6. Generalization Ability

In the previous subsection, we showed the internal representations of relations between instructions and actions acquired through the experience of an imposed task. Empirically, when such kinds of systematic representation can be organized, the model achieves a certain level of generalization ability (Sugita and Tani, 2005; Ogata et al., 2007; Yamada et al., 2016). Thus, we conducted learning experiments again by removing 50 or 25% of the possible situations from the training dataset. We chose removed patterns regularly so that each word, robot posture, and flag arrangement would appear uniformly, as shown in **Table 1**. Here, we trained only three models with 100, 150, and 300 LSTM units. The results are shown in **Figure 10**.

We first explain the performance of the models trained with only 50% of possible situations. For types (2)–(4), the models behaved appropriately for many of the possible patterns, even

TABLE 1 | To evaluate the model's generalization ability for the flag task, we conducted learning experiments again by removing (a) 50% or (b) 25% of the possible situations from training dataset.

Posture	Colors	Instructions															
		LUT	LUF	LDT	LDF	RUT	RUF	RDT	RDF	AUT	AUF	ADT	ADF	OUT	OUF	ODT	ODF
DOWN-DOWN	R-G	○	◎		◎	◎		◎	○	○		◎	◎	◎		◎	○
	G-R	◎		◎	○		◎	○	◎	◎	○	◎			◎	○	◎
	G-B	○	◎	◎		○	◎	◎			◎	○	◎	◎	◎		○
	B-G	◎	○		◎	◎	○		◎	○	◎	◎		○		◎	◎
	B-R		○	◎	◎	◎	◎		○	◎	○		◎	◎	○	◎	
	R-B	◎	◎	○		○		◎	◎	◎	◎		○		◎	○	◎
DOWN-UP	R-G		○	◎	◎	◎	◎	○		◎		○	◎	◎	○	◎	
	G-R	◎	◎		○		○	◎	◎	◎	◎		○	○		◎	◎
	G-B	○	◎		◎	◎		◎	○	○		◎	◎	◎	◎		○
	B-G	◎	○	◎		○	◎		◎	◎	○	◎		◎	○		◎
	B-R		◎	◎	○		◎	◎	○		◎	○	◎		◎	◎	○
	R-B	◎		○	◎	◎	○		◎		◎	◎	○	○	◎		◎
UP-DOWN	R-G	○	◎	◎		○	◎	◎			◎	○	◎	○	◎	◎	
	G-R	◎		○	◎	◎		○	◎	○	◎	◎			◎	○	◎
	G-B	○		◎	◎	◎	◎		○	◎	○		◎	◎		◎	○
	B-G	◎	◎		○		○	◎	◎	◎	◎		○		○	◎	◎
	B-R		◎	○	◎	◎	○	◎		○		◎	◎	◎	◎	○	
	R-B	◎	○	◎			◎	○	◎	◎		◎	○	◎	○		◎
UP-UP	R-G	◎	◎	○		○		◎	◎	◎	◎	○		◎		◎	○
	G-R		○	◎	◎	◎	◎	○		◎	○		◎		◎	○	◎
	G-B	◎		○	◎	◎	○		◎		◎	◎	○	○		◎	◎
	B-G		◎	◎	○	○	◎	◎			◎	○	◎	◎	◎	○	
	B-R	◎		◎	○		◎	○	◎	◎	○	◎			○	◎	◎
	R-B	○	◎		◎	◎		◎	○	○		◎	◎	○	◎		◎

(a) In the former case, only the situations indicated by ◎ marks were included in training data. (b) In the latter case, situations indicated not only by ◎ marks but also by ○ marks were included in the training data. The situations denoted as an empty cell were included in traning data in neither case. In this table, instruction patterns are abbreviated as follows. L: Left flag color; R: right flag color; A: AND-concatenated objectives; O: OR-concatenated objectives; U: up; D: down; T: true; F: false. For example, the cell referred to as DOWN-DOWN, R-G, LUF is indicated by a ◎ mark. It means that it is possible that the robot grasping R-G flags and waiting in a DOWN-DOWN posture receives an instruction "red up false" during training in both cases of (a) and (b). As another example, the cell referred to as UP-UP, R-B, OUT is indicated by a ○ mark. It means that it is possible that the robot grasping R-B flags and waiting in an UP-UP posture receives an instruction "red or blue up true" and "blue or red up true" during training in only the case of (b). In the other example, the cell referred to as DOWN-UP, B-R, LUT is denoted as empty. It means that the robot grasping B-R flags and waiting in an DOWN-UP posture does not receive an instruction "blue up true" during training in either case.

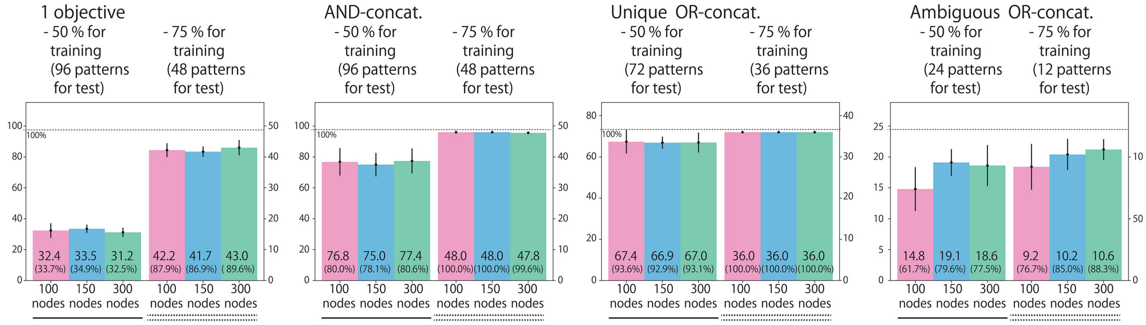


FIGURE 10 | To evaluate the model's generalization ability for the flag task, we conducted learning experiments again by removing (a) 50% or (b) 25% of the possible situations from training dataset. We evaluated the performance by counting how many unexperienced situation patterns each model dealt with appropriately. Similarly to **Figure 4**, we evaluated the performances with respect to each of the four situation types.

for the unexperienced ones. In contrast, only about one-third of the possible patterns of type (1) single-objective instructions, could be dealt with appropriately. In fact, this performance matches the level from chance, in which the robot uniformly randomly chooses one of three possible motions for a single-objective instruction (moving the left arm, moving the right arm, or keeping the current posture). To clarify why the network failed to generate appropriate motions, we checked some examples actually generated by the 100-node model (**Figure 11**). In one failure (indicated by the left rounded box), the final posture was correct but the trajectory was not stable, and so it did not satisfy the criterion that the error should be within 0.04. In another failure (right rounded box), a wrong action was chosen. The latter case indicates that although the model roughly learned to generate some possible actions after an instruction input, it failed to learn the relationships between color words and visual information.

One possible reason for failing to respond to (1) single-objective instructions is that only this type is actually linked with visual information. For example, in the case of type (2) AND-concatenated instructions, the RNN does not have to consider visual stimuli because, when the instruction includes "and," both arms have to be moved, regardless of the flag colors. In fact, when we tried to give the robot grasping R-B flags a contradictory instruction "green and blue up true," it raised both arms. In other contradictory cases, the results were similar. Also for types (3) and (4), when the instruction includes "or," either arm should be moved regardless of the flag colors. In that sense, type (1) single-objective instructions are more difficult than other types. It is possible that experiencing only half of the possible patterns is not enough to completely generalize the task space. Then, we performed the learning with the dataset in which only 25% of the situations were removed. In this case, the models responded appropriately to more than 80% of type (1) unexperienced situations in a generalized way.

In the next section, we describe another learning experiment based on the "bell task." The bell task is different from the flag task in two ways. First, the action sequences are more complicated because we collect motion data by using a real robot. Second, all the instructions including a logic word require referring to the visual information. We investigate whether a similar kind of representations of logic words that reflect their function can be organized in more realistic setting.

4. EXPERIMENT 2: BELL TASK

4.1. Task Overview

As a more realistic task, we conducted a learning experiment based on the bell task. In contrast with the first task, we collect motion data by using a real robot. First, a human places three bells colored red, green, and blue at random: one on the left, another to the center, and the other on the right front of the robot. Then, the human gives the robot a linguistic instruction consisting of a combination of a verb ("hit," "point"), an objective ("red," "green," "blue"), and an adverb ("slowly," "quickly"). When the left or right bell is indicated, the robot must hit (point at) the bell with the closer hand. However, when the center bell is indicated, the robot can hit (point at) the bell with either hand.

Similarly to the flag task, two objective (color) words can be concatenated by "and". In such cases, the robot has to hit (point at) the two indicated bells simultaneously. If two color words are concatenated by "or," hitting (pointing at) either bell indicated is correct. In another case, the logic word "not" can be prefixed to a color word (referred to as NOT-prefixed). In this case, hitting (pointing at) the two bells that are the complementary set of the indicated color is the correct response. For example, when the instruction is "hit not red quickly," the correct action is to simultaneously hit both the green and blue bells quickly.

The number of possible situations are 432: a combination of 72 possible instructions and 6 bell arrangements. In contrast to the flag task, in this task, the initial posture and end posture are the same, therefore the motion does not depend on the robot's initial posture. However, the actual action sequences are more complicated than the flag task, as shown in **Figure 12**.

4.2. Data Representation

We represent the execution of the bell task as a sequence of 26 dimensional vectors. The state $\boldsymbol{S}_t$ on time step t is represented as

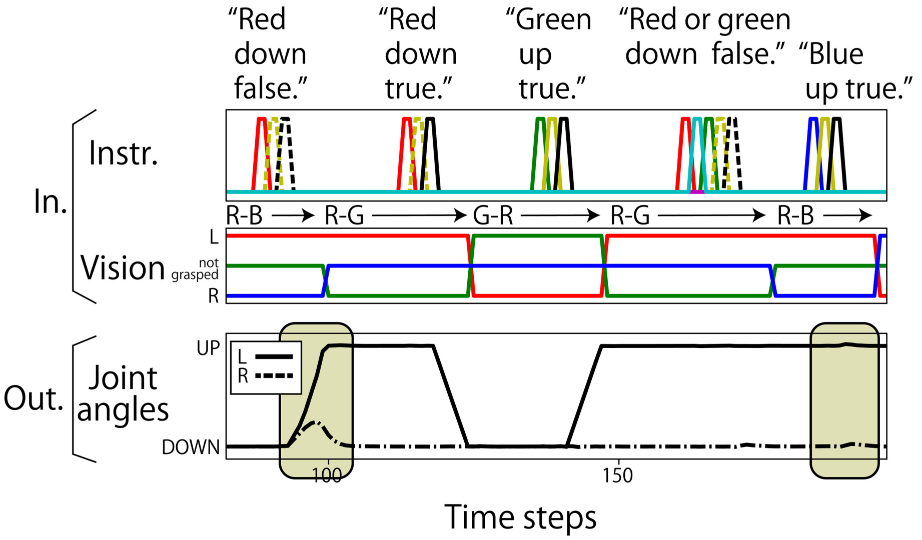


FIGURE 11 | In unexperienced situations of the flag task, some different patterns of failures could be seen (indicated by beige-colored rounded boxes). The first case, indicated by the left rounded box, was that the final posture was correct but the trajectory was not stable, thus it could not satisfy the criterion that the error should be within 0.04. The second pattern was that a wrong action was chosen. In the case indicated by the right rounded box, the right arm had to be raised. However, it was actually kept in the DOWN posture.

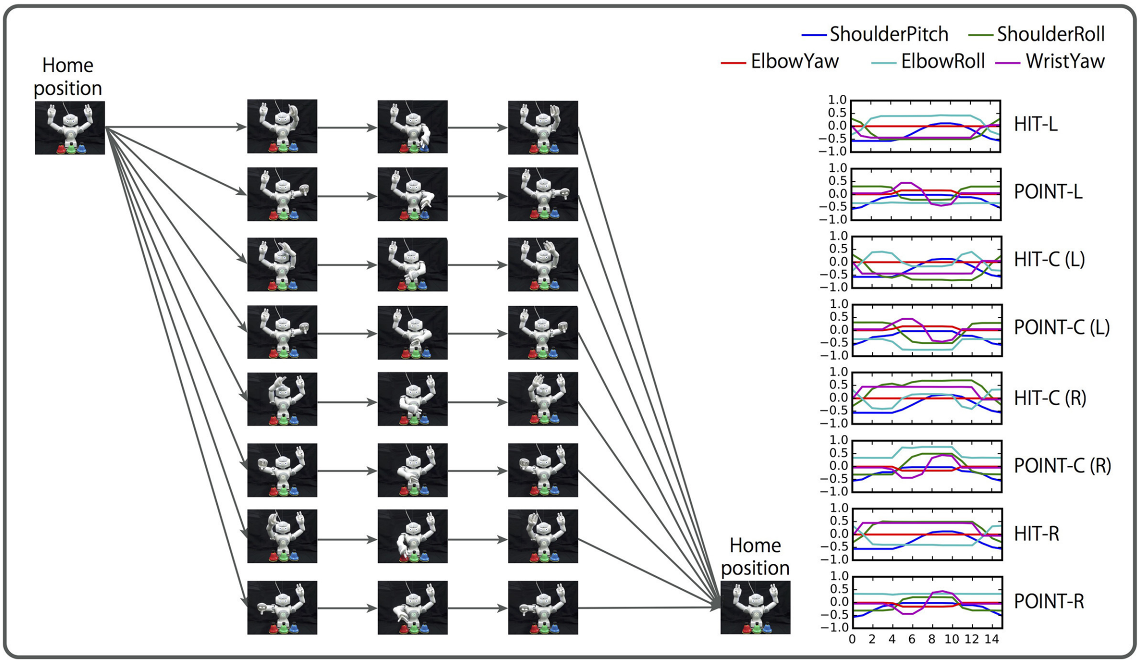


FIGURE 12 | Overview of the bell task. A human places three bells colored red, green, and blue in random order. The human gives the robot an instruction consisting of a combination of a verb ("hit," "point"), an objective ("red," "green," "blue"), and an adverb ("slowly," "quickly"). When the left or right bell is indicated, the robot must hit (point at) the bell with the closer hand. In the case of the center bell, the robot may hit (point at) it with either arm. Two color words can be concatenated by "and". In this case, the robot must act to both bells simultaneously (not presented in this figure). Two color words also can be concatenated by "or," in which case the robot may hit (point at) either bell. In another case, the logic word "not" can be prefixed to a color word. In this case, simultaneously hitting (pointing at) the two bells that are the complementary set of the indicated color is the correct response.

follows:

$$
\begin{aligned}
\boldsymbol{j}_t &= [j_{l0}^{(t)}, j_{l1}^{(t)}, j_{l2}^{(t)}, j_{l3}^{(t)}, j_{l4}^{(t)}, j_{r0}^{(t)}, j_{r1}^{(t)}, j_{r2}^{(t)}, j_{r3}^{(t)}, j_{r4}^{(t)}], && (8)\\
\boldsymbol{v}_t &= [v_{l0}^{(t)}, v_{l1}^{(t)}, v_{c0}^{(t)}, v_{c1}^{(t)}, v_{r0}^{(t)}, v_{r1}^{(t)}], && (9)\\
\boldsymbol{w}_t &= [w_0^{(t)}, w_1^{(t)}, w_2^{(t)}, w_3^{(t)}, w_4^{(t)}, w_5^{(t)}, w_6^{(t)}, w_7^{(t)}, w_8^{(t)}, w_9^{(t)}], && (10)\\
\boldsymbol{S}_t &= [\boldsymbol{j}_t; \boldsymbol{v}_t; \boldsymbol{w}_t]. && (11)
\end{aligned}
$$

To represent the robot joints, 10 elements that correspond to shoulder pitch, shoulder roll, elbow roll, elbow yaw, wrist yaw on each arm are assigned to the vector $\boldsymbol{j}_t$. Action sequences take approximately 16 steps in the case of QUICKLY actions, and approximately 25 steps in the case of SLOWLY actions. Action sequences are recorded by actually controlling the robot joints along predesigned trajectories. Visual information is encoded as a six-dimensional vector ($\boldsymbol{v}_t$). Three pairs of elements encode the bell colors. For example, v_{l0} and v_{l1} are used to represent the left bell color. In this task, it is assumed that the hues R, G, and B correspond to 0, 120, and 240° on the hue circle, respectively. The component $v_{l0}^{(t)}$ is the sine of the angle of the left bell color on the hue circle, $v_{l1}^{(t)}$ is its cosine. The pairs $v_{c0}^{(t)}$, $v_{c1}^{(t)}$ and $v_{r0}^{(t)}$, $v_{r1}^{(t)}$ encode the center and right bell colors, respectively, in the same way. This encoding method was used by Sugita and Tani (2005) and Yamada et al. (2016). Ten elements are assigned for language. Each element of $\boldsymbol{w}_t$ corresponds to one word, out of "hit," "point," "red," "green," "blue," "slowly," "quickly," "and," "or," and "not," and an instruction sentence is represented as a sequence of 1-hot vectors. Here, the instruction sentences and corresponding action sequences are concatenated on a computer, and sequences that represent interactions are similar to those for the flag task, with multiple repetitions of instructions and corresponding actions (and waiting phases).

4.3. Learning Setting and Evaluation Method

We made 512 sequential datasets for training, each of which includes eight episodes. All the possible situations were included at least once. We built models with 100, 300, 500, and 700 LSTM units, and trained them 10 times from randomly initialized learnable parameters. Adam is used as an optimizer. The number of learning iterations is 10,000, and the learning rate is set to 0.001.

After learning, we made another dataset for the evaluation which includes all possible situations 10 times. When the root mean squared errors between the generated angles and the correct ones per time step per joint during the action generations are less than 0.04, we judge that the RNN succeeds in generating an appropriate action. We regard the situation patterns in which the RNN succeeds in generating an appropriate action more than seven times out of 10 as "appropriately learned," just as in the flag task.

4.4. Task Performance after Training

We classify all the possible situations into four types: situations with (1) an instruction that includes only one objective word (72 situations); (2) an instruction is AND-concatenated (144 situations); (3) an instruction is OR-concatenated (144 situations); and (4) an instruction is NOT-prefixed (72 situations). We evaluate the performance by counting how many situation patterns each model learns appropriately with respect to each of four types. **Figure 13** shows the results. The task performance was improved by increasing the number of LSTM nodes. However, there is no significant difference between 500 and 700 node models for all situation types.

Next, we investigated which action was chosen by the model for instructions that had multiple correct actions. Here, we counted the result of the model with 500 LSTM units. The situations that have multiple correct actions are divided into three types. (a) The sentence instructs the robot to act on the center bell. In this case, acting with either arm is correct; therefore, two correct actions exist. (b) The sentence instructs the robot to act on the "left or right" bell. In this case, there are also two solutions. (c) The sentence instructs the robot to act on the "left or center" bell, or the "right or center" bell. In this case, there are three answers, (i) acting on the center bell with the left arm, (ii) acting on the center bell with the right arm, and (iii) acting on the left (right) bell with the left (right) arm. The results for these three types of situation are shown in **Table 2**. As shown in **Table 2**, the model could choose each of multiple solutions evenly. In fact, types (a), (b), and (c) have 24, 48, and 96 possible variations, respectively, and the test was conducted 10 times for each of

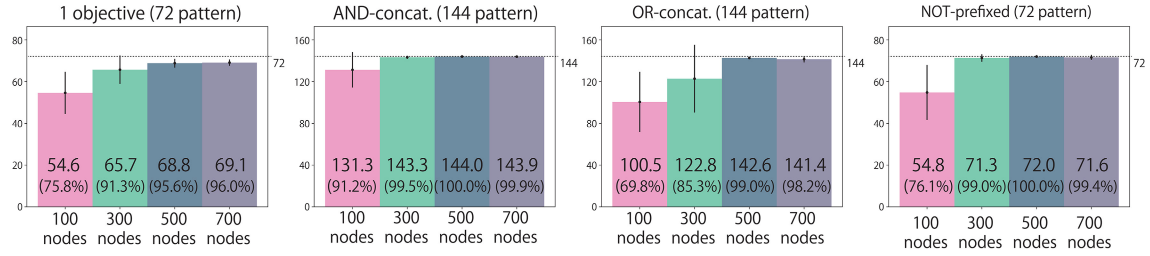


FIGURE 13 | Experiment 2 (bell task). Action generation performance. We evaluated performance by counting how many situation patterns each model learned appropriately with respect to each of the four situation types: (1) the instruction includes one objective; (2) the instruction is AND-concatenated; (3) the instruction is OR-concatenated; and (4) the instruction is NOT-prefixed. The written values are averages of 10 trials in which learning began with different seeds. Error bars represent standard deviations.

TABLE 2 | Ratios of chosen solutions for ambiguous cases (two or three acceptable answers) of bell task.

Situation	Choice 1 (%)	Choice 2 (%)	Choice 3	Failure (%)
(a)	49.2	42.9	–	7.9
(b)	50.0	44.4	–	5.6
(c)	29.1	29.4	36.6	5.0

The situations that have multiple correct actions are divided into three types. (a) The sentence instructs the robot to act on the center bell. In this case, acting with either arm is correct; therefore, two correct actions exist. (b) The sentence instructs the robot to act on the "left or right" bell. In this case, there are also two solutions. (c) The sentence instructs the robot to act on the "left or center" bell, or the "right or center" bell. In this case, there are three correct answers: (i) acting on the center bell with the left arm, (ii) acting on the center bell with the right arm, and (iii) acting on the left (resp., right) bell with the left (resp., right) arm.

them. In most of these ambiguous situations, the RNN chose each possible solution at least once. Just as in the flag task, the RNN could learn to behave appropriately even in such ambiguous situations.

4.5. Analyses of Internal Representations

4.5.1. Representations of "Or"

As in the flag task, we investigated the internal representations organized after learning by using PCA. First, we visualized the states of the memory cells after giving instructions in the form of "hit (objective part) slowly" that include one objective word or two OR-concatenated objective words (the left panel of **Figure 14**). This figure shows that the activations after the OR-concatenated instructions are located between the activations after the one objective word instructions. For example, "hit red or green slowly" and "hit green or red slowly" are embedded between the encodings of "hit red slowly" and "hit green slowly." This suggests the fact that "or" is represented by unstable points in the network dynamics, as in the flag task. In fact, the right panel of **Figure 14** shows an arch shaped activation space like the one in the flag task, although the shape is less clean. Note that although in the flag task, the meaning of "or" is always "left or right" regardless of the flag colors, in the current task the two candidate bells depend on the input color words and visual information. Even in this kind of situation, the functional meaning of "or" can be appropriately acquired in a way that is integrated with the objective color words.

4.5.2. Representations of "And" and "Not"

Figure 15 shows the memory cell states after giving instructions in the form of "hit (objective part) slowly," in which the objective part is AND-concatenated or NOT-prefixed. The bell arrangement was fixed in the order of R,G,B from left to right. In this task, "not" indicates the complementary set. Therefore, for example, "not green" and "red and blue" have the same meaning. Although the objective parts of these instructions are completely orthogonal to each other, they are located close each other in the space spanned by PC4 and PC5 and, as a result, instructions with the same meaning form clusters: that is, R-AND-G, G-AND-B, and B-AND-R. These instructions including logic words also require the RNN to consider visual information to determine the meaning of the sentence. Which two bells should be hit (pointed at) depends on both the input color words and visual information. The RNN learned to link these sentences flexibly in the sensorimotor information just from the experience of sequential data for the imposed task.

In summary, even in the bell task that requires both referring to visual information and processing of logic words simultaneously, the functional meaning of logic words could be appropriately organized in a way that was integrated with the referential words.

5. DISCUSSION

The current study conducted learning experiments involving translation from linguistic instructions, including both referential and logic words, into robot actions in order to investigate what kind of compositional representations emerged from the interactive experiences. In the case of referential words, objective words were merged with visual input, verbs were integrated with the robot's own posture, and as a result, appropriate actions were generated. Simultaneously, the model could also deal with the logic words "true," "false," "not," "and" and "or". By embedding these words as internal representations that reflect their functional properties, appropriate actions were achieved. In this following, we discuss three types of logic word separately.

5.1. True, False, Not

"True" and "false" in the flag task were understood as the goal-oriented action UP/DOWN by being combined with "up" and "down" in a X-OR manner. "Not" in the bell hitting task worked as an operation to choose a complementary set. For example, "not red" corresponded to "green and blue." The RNN learned to embed these completely orthogonal phrases as having the same meaning in the lower-ranking principal component space by its non-linear transformation. In the field of natural language understanding by deep learning, a similar kind of analysis has been performed. Li et al. (2016) showed that a model optimized for sentiment analysis changes its internal encoding drastically in response to the negation of an expression. Hence, for example, "not good" is encoded closer to "bad" than to "good". However, the visualization in the current study showed that even though the information that input sentences were completely different is still retained in the main component space, the combined representation corresponding to the behavioral meaning is reflected in the lower ranking principal components. In other words, not only information encoding compositionally integrated meaning but also information of compositional elements are retained in the model's memory.

This aspect seems to be important. For example, imagine that both of the sentences "hit red quickly" in the case of an RGB bell arrangement and "hit blue quickly" in the case of a BGR arrangement were encoded just as the action HIT-L-QUICKLY with the loss of the information about element words. In this case, it would be impossible for the model to respond appropriately to changes, such as a sudden replacement of bells during the action generation, because the color word information has been lost. By retaining the information about compositional elements, adaptive behavior to respond to such fluctuations would be

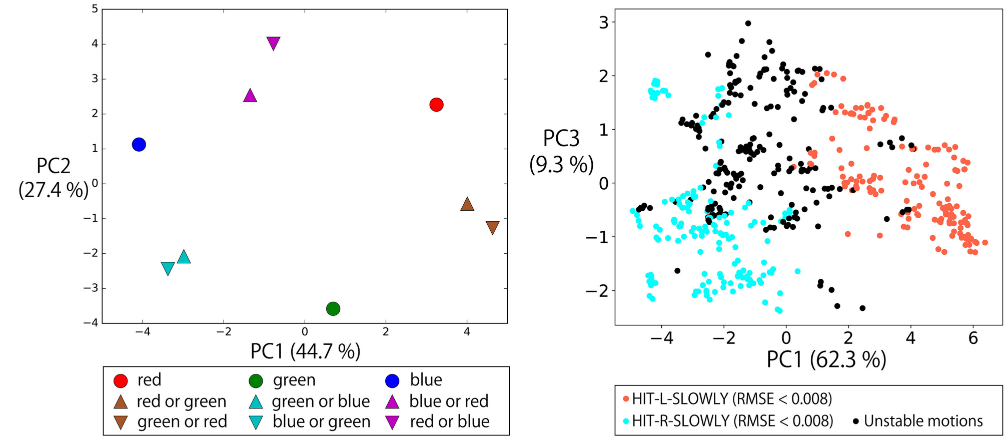


FIGURE 14 | Left: The states of the memory cells after giving instructions in the form of "hit (objective part) slowly" that include one objective word or two OR-concatenated objective words. The activations after the OR-concatenated instructions are located between the activations for the one objective word instructions. For example, "hit red or green slowly" and "hit green or red slowly" are embedded between the encodings of "hit red slowly" and "hit green slowly." **Right**: To a robot waiting with bells arranged in the order of RGB from left to right after 2,048 different contexts, we gave the instruction "hit red or blue slowly." The memory cell states after the instruction were arranged on an arch shaped space which was less defined than that for the flag task. When the activation was on the left side of the arch, the robot generated HIT-L-SLOWLY action. For activation on the right side, the robot generated HIT-R-SLOWLY action. When the activation was in the topmost area, an unstable action was generated.

possible, although it is not certain that our current model is capable of dealing with such situations because they were not included in training data.

5.2. And

In the flag task, "and" *per se* worked as a kind of universal quantifier without referring to objective words. For example, when a robot grasping R-B flags was given "green and blue up true," it lifted up both arms. In other contradictory cases, the results were similar. In other words, if the instruction includes "and," the color words are ignored and only the verb (and truth value) is considered. In that sense, "and" is represented as a concept one step higher. This interpretation of "and" by the neural network could not be expected before the experiment and is actually out of our common usage of "and"; but it can be seen as a reasonable and rational solution in the range of the current task. In contrast, in the bell task, AND-concatenated instructions required referring to visual information, and the model appropriately integrated them with the visual information and then generated correct both-hand actions.

In this way, "and" was represented in a different suitable manner with respect to each task. However, in general, there are more situations in which "and" is used in different ways to combine words, phrases, or sentences. The investigation of how such higher order or general types of "and" can be handled or represented is left for future work.

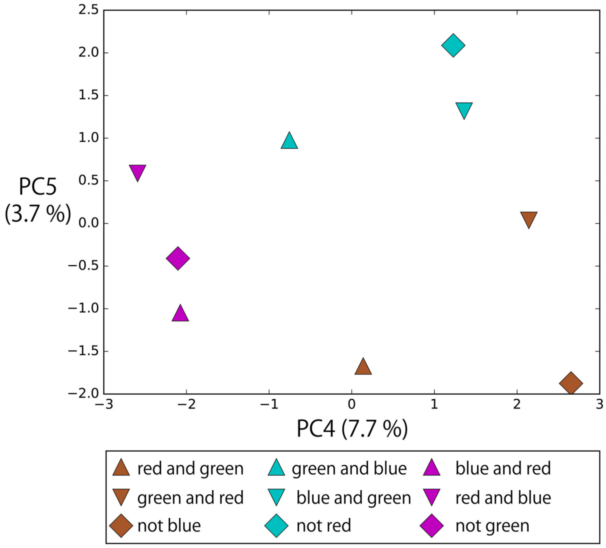


FIGURE 15 | The memory cell states after giving instructions in the form of "hit (objective part) slowly," in which objective part is AND-concatenated or NOT-prefixed. The bell arrangement was fixed in the order of R, G, B from left to right. In this task, "not" indicates the complementary set. Therefore, for example, "not green" and "red and blue" have the same meaning. Although the objective parts of these instructions are completely orthogonal, they are located close each other in the space spanned by PC4 and PC5 and, as a result, instructions with the same meaning form clusters: R-AND-G, G-AND-B, and B-AND-R.

5.3. Or

In the flag task it was shown that without noise applied to the joint angles, the model learned less successfully than it did with noise. This difference did not appear in preliminary experiments that did not include OR-concatenated instructions. We think that due to the inclusion of OR-concatenated instructions that introduce ambiguity by giving as correct either of the answers randomly each time, the optimization by minimization of the simple squared error became unstable. This is a very similar to a popular thought experiment called Buridan's ass. In the story, an ass is given grass feed on both its left and right sides, located at exactly the same distance away. Faced with this dilemma it could not choose a side and finally starved to death.

Our analysis shows the possibility that the network solved this problem, which the ass faced too honestly, by using the tiny amount of noise as a clue to determine which arm moves and by organizing unstable dynamics which converges to either of two fixed-point attractors. However, a more detailed analysis of the dynamical characteristics of the model is required. For example, Tani and Fukumura (1995) showed that a deterministic RNN model can reproduce a simple symbol sequence that is generated in accordance with probabilistic rules by producing a self-organizing chaotic dynamics. Namikawa et al. (2011) also demonstrated that a temporally hierarchical RNN could learn to generate pseudo-stochastic transitions between multiple motor primitives on a robot. Our experiment showed that a similar kind of function to generate actions as if they were generated probabilistically is achieved from the learning of an interactive instruction-action task that includes longer time dependency and more complexity. Our results also showed that the ability to deal with OR-concatenated instructions was improved by increasing LSTM node numbers. We think that by increasing the number of nodes and improving the representation ability the network could learn to forcibly embed the probabilistic experiences in a chaotic dynamics. We should analyze how the function is dynamically represented in the future.

5.4. Summary and Future Work

This study conducted learning experiments that translates linguistic sentences, including both referential and logic words, into robot actions to investigate what kind of compositional structures emerged from the experiences of interaction. Referential words were linked in the visual information and the robot's current state and then appropriate actions were generated. The logical words were also simultaneously represented by the model in accordance with their functions as logical operators. To be more precise, the words "true," "false" and "not" work as non-linear transformations to embed orthogonal phrases into the same area in a lower-rank principal component space. "And" in the flag task eliminated referring to the visual information in a rational way and worked as if it *per se* was a universal quantifier. "Or," which requires action generation that looks apparently random, was represented as an unstable space of the network's dynamical system.

Future work includes the following. First, we should confirm whether both referential and logic words are simultaneously learned when the complexity of the task is more extended. Although the scaling up of vocabulary size is one way to extend, the scaling up of syntactic variety is also required because the sentence patterns in this study were fixed in each task. In extended tasks, it would be possible that the logic words are used not only between words but also between phrases or clauses. Moreover, although the visual information in the current experiments is highly preprocessed, in more realistic tasks, the environment would include various meaningful information, not only color. Therefore, the relationships between language and the environment should be learned from low-level data (e.g., raw images) in a less arbitrary way. To deal with such tasks, we could extend our model by replacing the preprocessing module with another neural network model for vision, such as a convolutional neural network (CNN). In fact, some studies have actually combined a CNN with an RNN to learn the relationships between linguistic instructions and corresponding behavior in an end-to-end manner (Chaplot et al., 2017; Hermann et al., 2017).

Second, a more detailed analysis of the internal representations is required. This includes the analysis of more dynamical characteristics and the visualization of the activation patterns of each neuron. In particular, the latter seems to be valuable, because, although in the current study we visualized activation only in the principal component space, models that have memory cells, such as gated recurrent units or LSTM, are expected to encode different information and functions in specific nodes.

Finally, we are planning to build a bi-directional neural model to translate between linguistic and behavioral sequences. In fact, human language systems are bi-directionally translatable. To build a bi-directional model would be valuable both for understanding symbol grounding structure more deeply and for developing more flexible communicative agents.

AUTHOR CONTRIBUTIONS

TY, SM, HA, and TO conceived and designed the research, and wrote the paper. TY performed the experiment and analyzed the data.

FUNDING

This work was supported by a Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Young Scientists (A) (No. 16H05878), a JSPS Grant-in-Aid for JSPS Research Fellow (No. 17J1058), a JST CREST Grant (No. JPMJCR15E3), and the Program for Leading Graduate Schools, "Graduate Program for Embodiment Informatics" of the Ministry of Education, Culture, Sports, Science, and Technology.

REFERENCES

Arie, H., Endo, T., Jeong, S., Lee, M., Sugano, S., and Tani, J. (2010). "Integrative learning between language and action: a neuro-robotics experiment," in *20th International Conference on Artificial Neural Networks (ICANN2010)* (Thessaloniki), 256–265.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). "Neural machine translation by jointly learning to align and translate," in *IEEE International Conference on Learning Representations (ICLR2015)* (San Diego, CA).

Bleys, J., Loetzsch, M., Spranger, M., and Steels, L. (2009). "The grounded colour naming game," in *18th IEEE International Symposium on Robot and Human Interactive Communication (Ro-man 2009)* (Toyama).

Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S., Nehaniv, C., Fischer, K., et al. (2010). Integration of action and language knowledge : a roadmap for developmental robotics. *IEEE Trans. Autonom. Mental Dev.* 2, 167–195. doi: 10.1109/TAMD.2010.2053034

Chaplot, D. S., Sathyendra, K. M., Pasumarthi, R. K., Rajagopal, D., and Salakhutdinov, R. (2017). Gated-attention architectures for task-oriented language grounding. arXiv:1706.0723.

Chuang, L. W., Lin, C. Y., and Cangelosi, A. (2012). "Learning of composite actions and visual categories via grounded linguistic instructions: humanoid robot simulations," in *Proceedings of the International Joint Conference on Neural Networks* (Brisbane, QLD), 1–8.

Elman, J. L. (1990). Finding structure in time. *Cogn. Sci.* 14, 179–211.

Gers, F. A., and Schmidhuber, J. (2000). "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks* (Como), 189–194.

Harnad, S. (1990). The symbol grounding problem. *Phys. D* 42, 335–346.

Havrylov, S., and Titov, I. (2017). "Emergence of language with multi-agent games: learning to communicate with sequences of symbols," in *ICLR2017 Workshop* (Toulon).

Heinrich, S., Magg, S., and Wermter, S. (2015). "Analysing the multiple timescale recurrent neural network for embodied language understanding," in *Artificial Neural Networks - Methods and Applications in Bio- and Neuroinformatics*, eds P. D. Koprinkova-Hristova, V. M. Mladenov, and N. K. Kasabov (Chem: Springer), 149–174.

Heinrich, S., and Wermter, S. (2014). Interactive language understanding with multiple timescale recurrent neural networks. *Artif. Neural Netw. Mach. Lear.* 8681, 193–200. doi: 10.1007/978-3-319-11179-7_25

Hermann, K. M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., et al. (2017). Grounded language learning in a simulated 3D world. arXiv:1706.06551.

Hinaut, X., Petit, M., Pointeau, G., and Dominey, P. F. (2014). Exploring the acquisition and production of grammatical constructions through human-robot interaction with echo state networks. *Front. Neurorobot.* 8:16. doi: 10.3389/fnbot.2014.00016

Hinoshita, W., Arie, H., Tani, J., Okuno, H. G., and Ogata, T. (2011). Emergence of hierarchical structure mirroring linguistic composition in a recurrent neural network. *Neural Netw.* 24, 311–320. doi: 10.1016/j.neunet.2010.12.006

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

Kingma, D., and Ba, J. (2015). "Adam: a method for stochastic optimization," in *International Conference on Learning Representations (ICLR2015)* (San Diego, CA).

Kirby, S. (2002). Natural language from artificial life. *Artif. Life* 8, 185–215. doi: 10.1162/106454602320184248

Lazaridou, A., Peysakhovich, A., and Baroni, M. (2017). "Multi-agent cooperation and the emergence of (natural) language," in *International Conference on Learning Representations (ICLR2017)* (Toulon).

Li, J., Chen, X., Hovy, E., and Jurafsky, D. (2016). "Visualizing and understanding neural models in NLP," in *The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (San Diego, CA).

Mordatch, I., and Abbeel, P. (2017). Emergence of grounded compositional language in multi-agent populations. arXiv:1703.04908v1.

Namikawa, J., Nishimoto, R., and Tani, J. (2011). A neurodynamic account of spontaneous behaviour. *PLoS Comput. Biol.* 7:e1002221. doi: 10.1371/journal.pcbi.1002221

Ogata, T., Murase, M., Tani, J., Komatani, K., and Okuno, H. G. (2007). "Two-way translation of compound sentences and arm motions by recurrent neural networks," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, CA: IEEE), 1858–1863.

Ogata, T., and Okuno, H. G. (2013). "Integration of behaviors and languages with a hierarchal structure self-organized in a neuro-dynamical model," in *Proceedings of the 2013 IEEE Workshop on Robotic Intelligence in Informationally Structured Space, RiiSS 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013* (Singapore), 89–95.

Partee, B. H. (2004). *Compositionality in Formal Semantics: Selected Papers by Barbara H. Partee.* Oxford: Blackwell Publishers.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition* (MIT Press), 318–362.

Sasahara, K., Merker, B., and Okanoya, K. (2007). Simulated evolution of discourse with coupled recurrent networks. *Prog. Artif. Life* LNAI4828, 107–118. doi: 10.1007/978-3-540-76931-6_10

Schueller, W., and Oudeyer, P. Y. (2015). "Active learning strategies and active control of complexity growth in naming games," in *5th Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EPIROB)* (Providence, RI), 220–227.

Spranger, M. (2015). "Incremental grounded language learning in robot-robot interactions - examples from spatial language," in *5th Joint IEEE International Conference on Development and Learning and on Epigenetic Robotics (ICDL-EPIROB)* (Providence, RI), 196–201.

Steels, L. (2001). Language games for autonomous robots. *IEEE Intel. Syst.* 16, 16–22. doi: 10.1109/MIS.2001.956077

Steels, L., and Kaplan, F. (1998). "Stochasticity as a source of innovation in language games," in *the Conference on Artificial Life VI (Alife VI)* (Los Angels, CA), 368–376.

Steels, L., and McIntyre, A. (1998). Spatially distributed naming games. *Adv. Complex Syst.* 1, 301–323. doi: 10.1142/S021952599800020X

Stramandinoli, F., Marocco, D., and Cangelosi, A. (2012). The grounding of higher order concepts in action and language: a cognitive robotics model. *Neural Netw.* 32, 165–173. doi: 10.1016/j.neunet.2012.02.012

Stramandinoli, F., Marocco, D., and Cangelosi, A. (2017). Making sense of words: a robotic model for language abstraction. *Autonom. Robot.* 41, 367–383. doi: 10.1007/s10514-016-9587-8

Sugita, Y., and Tani, J. (2005). Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adapt. Behav.* 13, 33–52. doi: 10.1177/105971230501300102

Sugita, Y., and Tani, J. (2008). "A sub-symbolic process underlying the usage-based acquisition of a compositional representation," in *7th IEEE International Conference on Development and Learning (ICDL2008)* (Monterey, CA), 127–132.

Sukhbaatar, S., Szlam, A., and Fergus, R. (2016). "Learning multiagent communication with backpropagation," in *Neural Information Processing Systems 2016 (NIPS2016)* (Barcelona).

Sutskever, I., Vinyals, O., and Le, V. Q. (2014). "Sequence to sequence learning with neural networks," in *Neural Information Processing Systems 2014 (NIPS2014)* (Montreal).

Tani, J., and Fukumura, N. (1995). Embedding a grammatical description in deterministic chaos: an experiment in recurrent neural learning. *Biol. Cybern.* 72, 365–370. doi: 10.1007/BF00202792

Taniguchi, T., Nagai, T., Nakamura, T., Iwahashi, N., Ogata, T., and Asoh, H. (2016). Symbol emergence in robotics: a survey. *Adv. Robot*. 30, 706–728. doi: 10.1080/01691864.2016.1164622

Tuci, E., Ferrauto, T., Zeschel, A., Massera, G., and Nolfi, S. (2011). An experiment on behavior generalization and the emergence of linguistic compositionality in evolving robots. *IEEE Trans. Auton. Mental Dev.* 3, 176–189. doi: 10.1109/TAMD.2011.2114659

Vinyals, O., and Le, V. Q. (2015). "A neural conversational model," in *Proceedings of the 31st International Conference on Machine Learning* (Lille).

Wang, S. I., Liang, P., and Manning, C. D. (2016). "Learning language games through interaction," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* (Berlin), 2368–2378.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., et al. (2016). Google's neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

Yamada, T., Murata, S., Arie, H., and Ogata, T. (2015). "Attractor representations of language–behavior structure in a recurrent neural network for human–robot interaction," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2015)* (Hamburg), 4179–4184.

Yamada, T., Murata, S., Arie, H., and Ogata, T. (2016). Dynamical integration of language and behavior in a recurrent neural network for human–robot interaction. *Front. Neurorobot.* 10:5. doi: 10.3389/fnbot.2016.00005

Zhong, J., Cangelosi, A., and Ogata, T. (2017). Toward abstraction from multi-modal data: empirical studies on multiple time-scale recurrent models. in *The International Joint Conference on Neural Networks 2017 (IJCNN2017)* (Anchorage).

Zhong, J., Cangelosi, A., and Wermter, S. (2014). Toward a self-organizing pre-symbolic neural model representing sensorimotor primitives. *Front. Behav. Neurosci.* 8:22. doi: 10.3389/fnbeh.2014.00022

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 13 March 2018
doi: 10.3389/fnbot.2018.00011

Hierarchical Spatial Concept Formation Based on Multimodal Information for Human Support Robots

Yoshinobu Hagiwara, Masakazu Inoue, Hiroyoshi Kobayashi and Tadahiro Taniguchi*

Emergent Systems Laboratory, College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan

In this paper, we propose a hierarchical spatial concept formation method based on the Bayesian generative model with multimodal information e.g., vision, position and word information. Since humans have the ability to select an appropriate level of abstraction according to the situation and describe their position linguistically, e.g., "I am in my home" and "I am in front of the table," a hierarchical structure of spatial concepts is necessary in order for human support robots to communicate smoothly with users. The proposed method enables a robot to form hierarchical spatial concepts by categorizing multimodal information using hierarchical multimodal latent Dirichlet allocation (hMLDA). Object recognition results using convolutional neural network (CNN), hierarchical k-means clustering result of self-position estimated by Monte Carlo localization (MCL), and a set of location names are used, respectively, as features in vision, position, and word information. Experiments in forming hierarchical spatial concepts and evaluating how the proposed method can predict unobserved location names and position categories are performed using a robot in the real world. Results verify that, relative to comparable baseline methods, the proposed method enables a robot to predict location names and position categories closer to predictions made by humans. As an application example of the proposed method in a home environment, a demonstration in which a human support robot moves to an instructed place based on human speech instructions is achieved based on the formed hierarchical spatial concept.

Keywords: spatial concept, hierarchy, human-robot interaction, multimodal categorization, human support robot, unsupervised learning

OPEN ACCESS

Edited by:
Keum-Shik Hong,
Pusan National University,
South Korea

Reviewed by:
Cornelius Weber,
University of Hamburg, Germany
Zhong Yin,
University of Shanghai for Science and Technology, China

****Correspondence:***
Yoshinobu Hagiwara
yhagiwara@em.ci.ritsumei.ac.jp

Received: *29 November 2017*
Accepted: *26 February 2018*
Published: *13 March 2018*

Citation:
Hagiwara Y, Inoue M, Kobayashi H and Taniguchi T (2018) Hierarchical Spatial Concept Formation Based on Multimodal Information for Human Support Robots.
Front. Neurorobot. 12:11.
doi: 10.3389/fnbot.2018.00011

1. INTRODUCTION

Space categorization is an important function for human support robots. It is believed that humans predict unknown information flexibly by forming categories of space through their multimodal experiences. We define categories of spaces formed by self-organization from experience as spatial concepts. Furthermore, prediction based on the connection between concepts and words is thought to lead to a semantic understanding of words. It means that spatial concept formation is an important function of human intelligence, and having this ability is important for human support robots.

Spatial concepts form a hierarchical structure. The use of this hierarchical structure enables humans to predict unknown information using concepts in an appropriate layer. For example, humans can linguistically represent their own positions at an appropriate level of abstraction according to the context of communication, such as "I'm in my home" at the global level, "I'm in the living room" at the intermediate level, and "I'm in front of the TV" at the local level. In this case, the living room has the home in the higher layer and front of the TV in the lower layer. By learning such a hierarchical structure, even if the unknown place does not have features such as front of the TV, its characteristics can be predicted if it has features of the living room. It is expected that the robot acquires spatial concepts in a higher layer by learning the commonality of features in spatial concepts at the lower layer.

Furthermore, the hierarchical structure of spatial concepts plays an important role when a robot moves based on linguistic instructions from a user. As shown in **Figure 1**, even if multiple tables are present in a room, robots can recognize them individually by using a spatial concept at a higher layer, such as "the front of the table in the living space." Indeed, in RoboCup@Home, an international competition in which intelligent robots coexist with humans in home environments, location names are defined as two layers in the tasks of a General Purpose Service Robot[1] as shown in **Table 1**. This table indicates that having sense of space relations is important for a robot coexisting with humans, e.g., that the living space has a center table. By having such hierarchical spatial concepts, it becomes possible to describe and move within a space based on linguistic communication with a user.

We assume that a computational model, which considers the hierarchical structure of spatial concepts, enables robots to acquire not only the spatial concepts, but also the hierarchical structure hiding among the spatial concepts through a bottom-up approach and form spatial concepts similar to those perceived by humans. The goal of this study was to develop a robot that can predict unobserved location names and positions from observed information using formed hierarchical spatial concepts. The main contributions of this paper are as follows.

- We propose a method of forming hierarchical spatial concepts using a Bayesian generative model based on multimodal information, e.g., vision, position, and word information.
- We show that spatial concepts formed by the proposed method enable a robot to predict location names and positions similar to prediction made by humans.
- We demonstrate application examples in which a robot moves within and describes a space based on linguistic communication with a user through the hierarchical spatial concept formed by the proposed method.

The rest of this paper is structured as follows. Section 2 describes related works. Section 3 presents an overview and the computational model of hierarchical spatial concept formation. Section 4 presents experimental results evaluating the effectiveness of the proposed method in space categorization. Section 5 describes application examples of using hierarchical spatial concepts in a home environment. Finally, section 6 presents conclusions.

2. RELATED WORKS

In order for a robot to move within a space, a metric map consisting of occupancy grids that encode whether or not an area is navigable is generally used. The simultaneous localization and mapping (SLAM) (Durrant-Whyte and Bailey, 2006) is a famous localization method for mobile robots. However, the tasks that are coordinated with a user cannot be performed using only a metric map, since semantic information is required for interaction with a user. Nielsen et al. (2004) proposed a method of expanding a metric map into a semantic map by attaching a single-frame snapshot in order to share spatial information between a user and a robot. As a bridge between a metric map and human-robot interaction, research on semantic maps that provide semantic attributes (such as object recognition results) to metric maps has been performed (Pronobis et al., 2006; Ranganathan and Dellaert, 2007). Studies have also been reported on giving semantic object annotations to 3D point cloud data (Rusu et al., 2008, 2009). Moreover, in terms of studies based on multiple cues, Espinace et al. (2013) proposed a method of characterizing places according to low-level visual features associated to objects. Although these approaches could categorize spaces based on semantic information, they did not deal with linguistic information about the names that represent spaces.

In the field of navigation tasks with human-robot interaction, methods of classifying corridors and rooms using a predefined ontology based on shape and image features have been proposed (Zender et al., 2008; Pronobis and Jensfelt, 2012). In studies on semantic space categorization, Kostavelis and Gasteratos (2013) proposed a method of generating a 3D metric map that is semantically categorized by recognizing a place using bag of features and support vector machines. Granda et al. (2010) performed spatial labeling and region segmentation by applying a Gaussian model to the SLAM module of a robot operating system (ROS). Mozos and Burgard (2006) proposed a method of classifying metric maps into semantic classes by using adaboost as a supervised learning method. Galindo et al. (2008) utilized semantic maps and predefined hierarchical spatial information for robot task planning. Although these approaches were able to ground several predefined names to spaces, the learning of location names through human-robot communication in a bottom-up manner has not been achieved.

Many studies have been conducted on spatial concept formation based on multimodal information observed in individual environments (Hagiwara et al., 2016; Heath et al., 2016; Rangel et al., 2017). Spatial concepts are formed in a bottom-up manner based on multimodal observed information, and allow predictions of different modalities. This makes it possible to estimate the linguistic information representing a space from position and image information in a probabilistic way. Gu et al. (2016) proposed a method of learning relative space categories from ambiguous instructions. Taniguchi et al. (2014, 2016) proposed computational models for a mobile robot to acquire spatial concepts based on information from recognized speech and estimated self-location. Here, the spatial concept was defined as the distributions of names and positions at each place.

[1] GPSR Command Generator: https://github.com/kyordhel/GPSRCmdGen

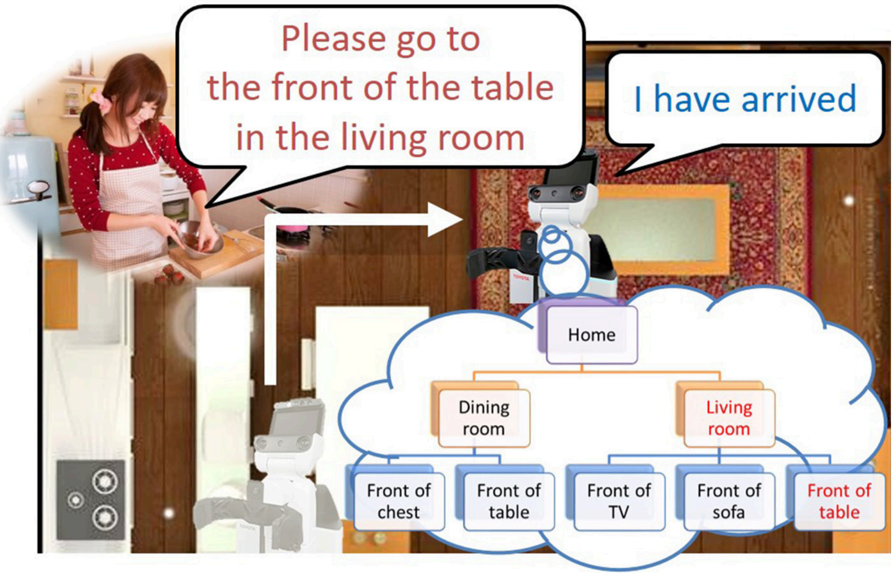


FIGURE 1 | Example of movement based on linguistic instructions with a hierarchical space structure.

The method enables a robot to predict a positional distribution from recognized human speech through formed spatial concepts. Ishibushi et al. (2015) proposed a method of learning the spatial regions at each place by stochastically integrating image recognition results and estimated self-positions. In these studies, it was possible to form a spatial concept conforming to human perception such as an entrance and a corridor by inferring the parameters of the model.

However, these studies did not focus on the hierarchical structure of spatial concepts. In particular, the features of the higher layer, such as the living space, are included in the features of the lower layer, such as the front of the television, and it was difficult to form the spatial concept in the abstract layer. Furthermore, the ability to understand and describe a place linguistically in different layers is an important function in robots that provide services through linguistic communication with humans. Despite the importance of the hierarchical structure of spatial concepts, a method that enables such concept formation has not been proposed in previous studies. We propose a method that forms a hierarchical spatial concept in a bottom-up manner from multimodal information and demonstrate the effectiveness of the formed spatial concepts in predicting location names and positions.

3. HIERARCHICAL SPACE CONCEPT FORMATION METHOD

3.1. Overview

An overview of the proposed method of forming hierarchical spatial concepts is shown in **Figure 2**. First, a robot was controlled manually in an environment based on a map generated by simultaneous localization and mapping (SLAM) (Durrant-Whyte and Bailey, 2006) and acquires multimodal information, i.e., vision, position, and word information from attached sensors. Vision information is acquired as a feature vector generated by a convolutional neural network (CNN) (Krizhevsky et al., 2012). Position information is acquired as coordinate values in the map estimated by Monte Carlo localization (MCL) (Dellaert et al., 1999). Word information is acquired as set of words by word recognition. Text input is used for word recognition in this study. Second, acquired vision, position, and word information is represented as histograms. The histograms are utilized as observations in each modality. Third, the formation of hierarchical spatial concepts is performed by using hierarchical multimodal latent Dirichlet allocation (hMLDA) (Ando et al., 2013) on the observations. The proposed method enables a robot to form hierarchical spatial concepts in a bottom-up manner based on observed multimodal information. Therefore, it is possible to adaptively learn location names and the hierarchical structure of a space, which depend on the environment.

TABLE 1 | Definition of location names with two layers in RoboCup@Home.

Name (1st layer)	Name (2nd layer)
Living room	Bar
Living room	TV stand
Living room	Center table
Office	Drawer
Office	Desk
Kitchen	Bar
Kitchen	Cupboard
Bathroom	Cupboard

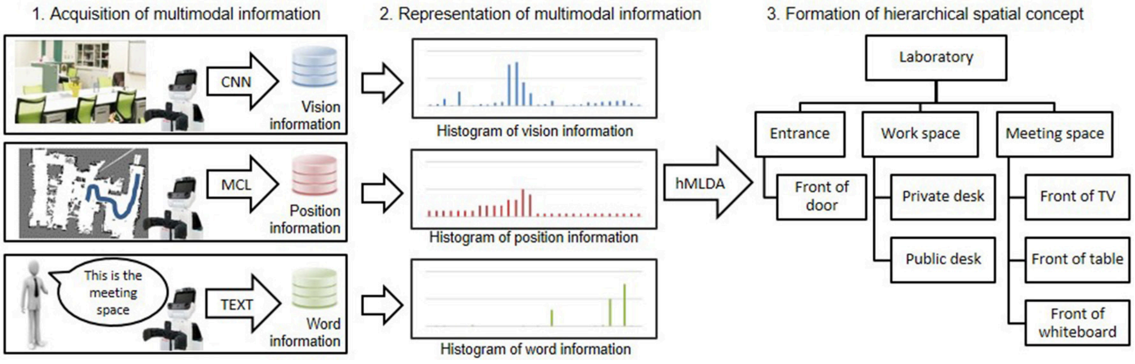

1. Acquisition of multimodal information
2. Representation of multimodal information
3. Formation of hierarchical spatial concept
CNN
Vision information
MCL
Position information
This is the meeting space
TEXT
Word information
Histogram of vision information
Histogram of position information
Histogram of word information
hMLDA
Laboratory
Entrance
Work space
Meeting space
Front of door
Private desk
Public desk
Front of TV
Front of table
Front of whiteboard

FIGURE 2 | Overview of the proposed method for hierarchical spatial concept formation.

3.2. Acquisition and Feature Extraction of Multimodal Information

3.2.1. Vision Information

Vision information was acquired as the object recognition results of a captured image by Caffe (Jia et al., 2014), which is a framework of CNN (Krizhevsky et al., 2012) provided by Berkeley Vision and Learning Center. The parameters of CNN were trained by using the dataset from the ImageNet Large Scale Visual Recognition Challenge 2013[2], which comprises 1,000 object classes, e.g., television, cup, and desk. The output of Caffe is given as a probability $p(a_i)$ at an object class $a_i \in \{a_1, a_2, ..., a_I\}$ where I is the number of object classes and was set to 1,000. The probability $p(a_i)$ was represented as a 1,000-dimensional histogram of vision information $\boldsymbol{w}^{(v)} = (w_1^{(v)}, w_2^{(v)}, \cdots, w_{1,000}^{(v)})^T$ by the following equation:

$$w_i^{(v)} = p(a_i) * 10^2. \tag{1}$$

3.2.2. Position Information

The position information (x, y) in the map generated by SLAM was estimated by MCL (Dellaert et al., 1999). It is assumed that the observed information is generated from a multinomial distribution in hMLDA. For this reason, the observed information with a continuous value is generally converted into a finite dimensional histogram by vector quantization. Ando et al. (2013) replaced the observed information with typical patterns by k-means clustering to form a finite dimensional histogram. The proposed method converts a position information (x, y) into a finite dimensional histogram of position information $\boldsymbol{w}^p$ by hierarchical k-means clustering. The positional information (x, y) was classified hierarchically into 2, 4, 8, 16, 32, and 64 clusters with six layers by applying k-means clustering with $k = 2$ six times. If a position (x, y) was classified into a cluster $c_i \in \{0, 1\}$ at the ith layer, a path for the position information was described as $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$. The path C has the structure of a binary tree with six layers. The number of nodes at the 6th layer is $2^6 = 64$. The position information (x, y) is represented as a 64-dimensional histogram of the position information $\boldsymbol{w}^{(p)} = (w_1^{(p)}, w_2^{(p)}, \cdots, w_{64}^{(p)})^T$ by incrementing $w_i^{(p)}$ based on the path C. For example, in a path C of position information (x, y), when $c_1 = 0$, $w_1^{(p)}$ to $w_{32}^{(p)}$ corresponding to nodes at the 6th layer are incremented, and when $c_1 = 1$, $w_{33}^{(p)}$ to $w_{64}^{(p)}$ are incremented. Similarly, $w^{(p)}$ corresponding to nodes at the 6th layer below it are incremented in each layer.

3.2.3. Word Information

The voice information uttered by a user is converted manually into text data, which is then used as word information. In section 5, rospeex (Sugiura and Zettsu, 2015) is used to convert human speech into text data. The location names are manually extracted from the text data. The word information is described as a set of location names, which is a Bag of Words (Harris, 1954) with a location name as a word. The user could give not only one name but also several names to a robot at a given position. The given word information was represented as a histogram of word information $\boldsymbol{w}^{(w)} = (w_1^{(w)}, w_2^{(w)}, \cdots, w_J^{(w)})^T$. J is the dimension of $\boldsymbol{w}^{(w)}$, and depends on the number of location names in a dictionary $S = \{s_1, s_2, \cdots, s_J\}$, which was obtained through the training phase. $w_j^{(w)}$ was incremented when a location name s_j was taught from user. J is the number of location names. Histograms of vision, position, and word information were used as observations in hMLDA.

3.3. Hierarchical Categorization by hMLDA

The hierarchical structure of spatial concepts is supported by nested Chinese restaurant process (nCRP) (Blei et al., 2010) in hMLDA (Ando et al., 2013). nCRP is an extended model of the Chinese restaurant process (CRP) (Aldous, 1985), which is a Dirichlet process used to generate multinomial distribution with infinite dimensions. nCRP stochastically calculates the hierarchical structure based on the idea that there are infinite Chinese restaurants with infinite number of tables. **Figure 3** shows the overview of nCRP. A box and a circle represent a restaurant and a customer, respectively. The customer

[2] ILSVRC2013: http://www.image-net.org/challenges/LSVRC/2013/

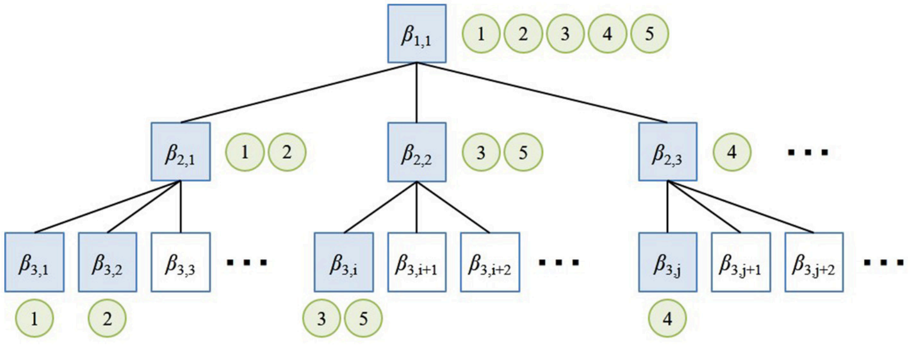


FIGURE 3 | Overview of nested Chinese restaurant process (nCRP).

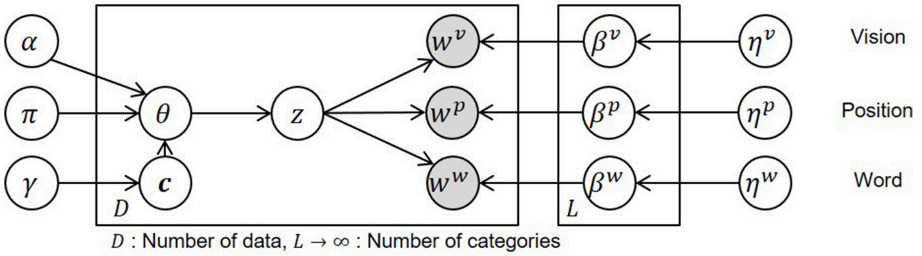


FIGURE 4 | Graphical model of hierarchical spatial concept formation.

stochastically decides the restaurant to visit. In the proposed method, a box and a circle mean a spatial concept and data, respectively. Data is stochastically allocated to a spatial concept in each layer by the nCRP. In hMLDA, each spatial concept has a probability distribution with parameter $\beta_{l,i}$ to generate data. The proposed method forms a hierarchical spatial concept by hierarchical probabilistic categorization using nCRP. In the non-hierarchical approach, a place called "meeting space" and its partial places called "front of the table" and "front of the TV" are formed in the same layer. Therefore, the meeting space is learned as a place different from places called "front of the table" and "front of the TV." The proposed method enables the robot to learn the meeting space as a upper concept encompassing places called "front of the table" and "front of the TV" as lower concepts.

The graphical model of hMLDA in the proposed method and the definition of the variables are shown in **Figure 4** and **Table 2**, respectively. In **Figure 4**, $\boldsymbol{c}$ is a tree-structured path generated by nCRP with a parameter γ and z is a category index for a spatial concept that is generated by a stick-breaking process (Pitman, 2002) with parameters α and π. $w^{\upsilon}, w^{p}, w^{w}$ are acquired vision, position, and word information generated by multinomial distributions with a parameter β^m at a modality m ($m \in \upsilon, p, w$). β^m was determined according to a Dirichlet prior distribution with a parameter η^m. D and L written on plates are the number of acquired data and the number of categories, respectively.

The generation process of the model is described as follows:

$$\beta_k^m \sim \text{Dirichlet}(\eta^m) \tag{2}$$

$$\boldsymbol{c}_d \sim \text{nCRP}(\gamma) \tag{3}$$

$$\theta_d \sim \text{GEM}(\alpha, \pi) \tag{4}$$

$$z_{d,n}^m \sim \text{Multi}(\theta_d) \tag{5}$$

$$w_d^m \sim \text{Multi}(\beta_{\boldsymbol{c}_d}[z_{d,n}^m]), \tag{6}$$

where:

- The parameter β_k^m of a multinomial distribution is generated by a Dirichlet prior distribution with a parameter η^m in a table $k(k \in T)$, e.g., $\beta_{1,1}$ and $\beta_{2,1}$ in **Figure 3**.
- The path c_d in a tree structure for the data d ($d \in 1, 2, ..., D$) is decided by nCRP with a parameter γ. c_d is represented by the sequence of numbers assigned to each node in the path, e.g., $\{(1, 1), (2, 1), (3, 2)\}$ at data 2 in **Figure 3**.
- The parameter θ_d of a multinomial distribution is generated by the stick-breaking process based on a GEM distribution which forms θ_d from a Beta$(\alpha\pi, (1-\alpha)\pi)$ distribution with the parameters $\alpha (0 \le \alpha \le 1)$ and $\pi (\pi > 0)$ (Pitman, 2006). θ_d represents the selection probability of a layer in a path c_d and corresponds to the generation probability of a category index z in a path c_d.
- $z_{d,n}^m$, which is a category index at the nth feature of the observed information w_d^m, is generated by a multinomial distribution with a parameter θ_d.

TABLE 2 | Definition of variables in the graphical model.

Variable	Definition
w^v, w^p, w^n	Observation of vision, position and word information
z	Index of category
$\beta^v, \beta^p, \beta^n$	Parameter of multinomial distribution in vision, position and word information
θ	Parameter of multinomial distribution in category
$\boldsymbol{c}$	Path of tree structure
η^v, η^v, η^w	Parameter of Dirichlet prior distribution
γ	Hyper parameter of $\boldsymbol{c}$
α, π	Hyper parameter of θ

- w_d^m is the observed information generated by a multinomial distribution with a parameter β from a category $z_{d,n}^m$ at a path c_d.

In this study, z is equivalent to a spatial concept expressed by the location name such as "the living room" or "front of the table."

Model parameter learning was performed by a Gibbs sampler. Parameters were calculated by alternately sampling a path c_d for each datum and a category $z_{d,n}^m$ assigned to the nth feature value of a modality m of the data d in the path. Category $z_{d,n}^m$ was sampled according to the following formula.

$$\begin{aligned} z_{d,n}^m &\sim p(z_{d,n}^m | \boldsymbol{z}_{-(d,n)}^m, \boldsymbol{c}, \boldsymbol{w}^m, \alpha, \pi, \eta^m) \\ &\propto p(z_{d,n}^m, \boldsymbol{z}_{-(d,n)}^m, \boldsymbol{c}, \boldsymbol{w}^m | \alpha, \pi, \eta) \\ &\propto p(z_{d,n}^m | \boldsymbol{z}_{d,-n}^m, \alpha, \pi) p(w_{d,n}^m | \boldsymbol{z}, \boldsymbol{c}, \boldsymbol{w}_{-(d,n)}^m, \eta^m), \end{aligned} \tag{7}$$

where $-(d, n)$ means excluding the nth feature value of the data d. $p(z_{d,n}^m | \boldsymbol{z}_{d,-n}^m, \alpha, \pi)$ is a multinomial distribution generated by the stick-breaking process. The probability, that k is assigned to a category of the n-th feature of modality m of the d-th data, was calculated by the following formula.

$$\begin{aligned} p(z_{d,n}^m = k | \boldsymbol{z}_{d,-n}^m, \alpha, \pi) &= E\left[V_k \prod_{j=1}^{k-1} (1 - V_j) | \boldsymbol{z}_{d,-n}^m, \alpha, \pi \right] \\ &= E\left[V_k | \boldsymbol{z}_{d,-n}^m, \alpha, \pi \right] \prod_{j=1}^{k-1} E\left[1 - V_j | \boldsymbol{z}_{d,-n}^m, \alpha, \pi \right] \\ &= \frac{(1-\alpha)\pi + \#[\boldsymbol{z}_{d,-n}^m = k]}{\pi + \#[\boldsymbol{z}_{d,-n}^m \geq k]} \prod_{j=1}^{k-1} \frac{\alpha\pi + \#[\boldsymbol{z}_{d,-n}^m > j]}{\pi + \#[\boldsymbol{z}_{d,-n}^m \geq j]}, \end{aligned} \tag{8}$$

where $\#[\cdot]$ is a number that satisfies a given condition and V_k and V_j are values that determine the rate of folding a branch in categories k and j by the stick-breaking process, respectively.

In Formula (7), $p(w_{d,n}^m | \boldsymbol{z}, \boldsymbol{c}, \boldsymbol{w}_{-(d,n)}^m, \eta^m)$ is the probability that a feature value is generated from a path $\boldsymbol{c}_d$ and a category $z_{d,n}^m$. Since it is assumed that the parameters of the multinomial distribution that generates a feature value are generated from a Dirichlet prior distribution, the following formula is obtained.

$$\begin{aligned} p(w_{d,n}^m | \boldsymbol{z}, \boldsymbol{c}, \boldsymbol{w}_{d,n}^m, \eta^m) \propto \#[\boldsymbol{z}_{-(d,n)}^m &= z_{d,n}^m, \boldsymbol{c}_{z_{d,n}^m} = c_{d,z_{d,n}^m}, \boldsymbol{w}_{-(d,n)}^m \\ &= w_{d,n}^m] + \eta^m \end{aligned} \tag{9}$$

This gives the number of times that a category $z_{d,n}^m$ is assigned to a feature value $w_{d,n}^m$ in a path $\boldsymbol{c}_d$. A path c_d was sampled by the following formula.

$$\begin{aligned} \boldsymbol{c}_d &\sim p(\boldsymbol{c}_d | \boldsymbol{w}^v, \boldsymbol{w}^p, \boldsymbol{w}^w, \boldsymbol{c}_{-d}, \boldsymbol{z}, \eta^v, \eta^p, \eta^w, \gamma) \\ &\propto p(\boldsymbol{c}_d | \boldsymbol{c}_{-d}, \gamma) p(\boldsymbol{w}_d^v | \boldsymbol{c}, \boldsymbol{w}_{-d}^v, \boldsymbol{z}^v, \eta^v) p(\boldsymbol{w}_d^p | \boldsymbol{c}, \boldsymbol{w}_{-d}^p, \boldsymbol{z}^p, \eta^p) \\ &\quad p(\boldsymbol{w}_d^w | \boldsymbol{c}, \boldsymbol{w}_{-d}^w, \boldsymbol{z}^w, \eta^w), \end{aligned} \tag{10}$$

where $\boldsymbol{c}_{-d}$ is a set of paths excluding $\boldsymbol{c}$ from $\boldsymbol{c}_d$. Sampling based on Formulas (9) and (10) was repeated for each training datum $d \in \{d_1, d_2, \cdots, d_D\}$. In this process, paths and categories for all observed data converge to $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{z}}$.

3.4. Name Prediction and Position Category Prediction

If vision information w_t^v and position information $\boldsymbol{w}_t^p$ are observed at a time t, then the posterior probability of word information w_t^w can be calculated with estimated parameters $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{z}}$ by the following formula.

$$\begin{aligned} &p(w_t^w | \hat{\boldsymbol{z}}, \hat{\boldsymbol{c}}, \boldsymbol{w}^w, \boldsymbol{w}^v, \boldsymbol{w}^p, \boldsymbol{c}_t, w_t^v, w_t^p, \alpha, \pi, \eta^n, \eta^v, \eta^p) = \\ &\sum_{z_t} p(w_t^w | z_t, \hat{\boldsymbol{z}}^w, \hat{\boldsymbol{c}}, \boldsymbol{w}^w, \eta^w) \\ &p(z_t | \hat{\boldsymbol{z}}^v, \hat{\boldsymbol{z}}^p, \hat{\boldsymbol{c}}, \boldsymbol{w}^v, \boldsymbol{w}^p, \boldsymbol{c}_t, w_t^v, w_t^p, \alpha, \pi, \eta^v, \eta^p) \end{aligned} \tag{11}$$

The location name $\hat{n}$ can be predicted by the maximum value of the calculated posterior probability.

If word information w_t^w is obtained at a time t, then a category z_t^w can be predicted by Formula (12) and selecting position $\hat{p}$ randomly from dataset $D_{z_t^w}$, which is a set of position data categorized into z_t^w. $D_{z_t^w}$ was automatically generated by the robot itself as a part of the categorization process.

$$z_t^w \sim p(z_t^w | \boldsymbol{z}_{-t}^w, w_t^w, \hat{\boldsymbol{c}}, \boldsymbol{w}^w, \boldsymbol{w}^v, \boldsymbol{w}^p, \eta^w, \eta^v, \eta^p, \alpha, \pi) \tag{12}$$

4. EXPERIMENT

4.1. Purpose

We conducted experiments to verify whether the proposed method can form hierarchical spatial concepts, which enable a robot to predict location names and position categories close to predictions made by humans. In the experiment, (1) the influence of multimodal information, i.e., words, on the formation of a hierarchical spatial concept was evaluated by comparing the space categorization results of using the proposed method and those of hierarchical latent Dirichlet allocation (hLDA) (Blei et al., 2010), which is a hierarchical categorization method with single modality; (2) the similarity between the hierarchical spatial concepts formed by the proposed method and those made by humans was evaluated in terms of predicting location names and position categories.

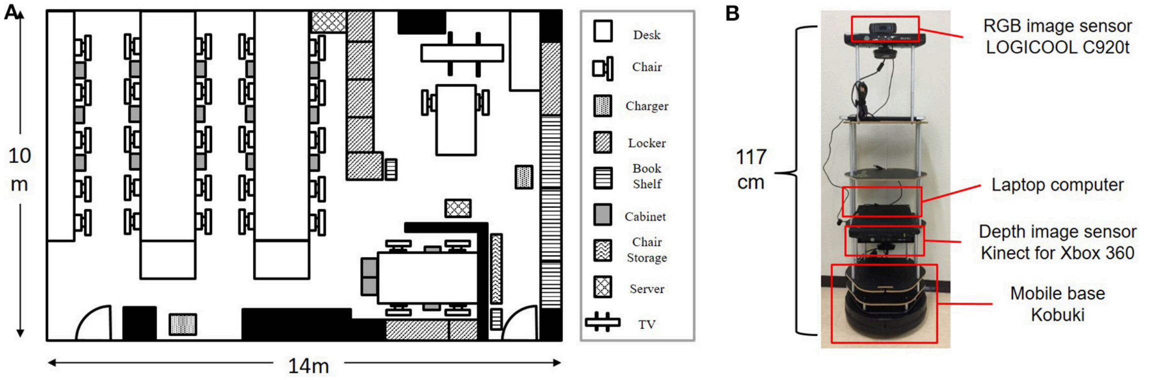


FIGURE 5 | Experimental environment and mobile robot for learning and testing. **(A)** Experimental environment for data collection. **(B)** Mobile robot used for experiments.

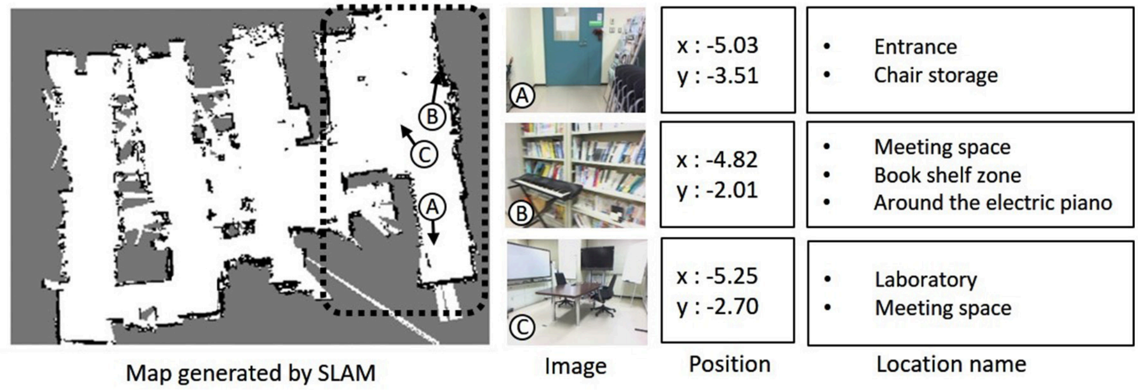


FIGURE 6 | Map generated by SLAM and examples of collected data: image, position, and location name.

4.2. Experimental Conditions

Figure 5A shows an experimental environment which includes furniture, e.g., tables, chairs, and a book shelf, in order to collect training and test data. **Figure 5B** shows a mobile robot, which consists of a mobile base, a depth sensor, an image sensor, and a computer, used to generate a map and collect multimodal information in the test environment. The height of the camera attached to the robot was 117 cm in consideration of the typical eye level in the human body. This is equivalent to the average height of a 5-year-old boy in Japan. The Navigation Stack package[3] was used with ROS Hydro[4] for mapping, localization, and moving in the experiment. The robot was manually controlled to collect data from the environment. The orientation of the robot was controlled in as many different orientations as possible.

Figure 6 shows a map generated in the environment by the robot using SLAM and examples of the collected data. Collected data consisted of image, position, and word information as shown in the samples of collected data at A, B, and C. In the experiment, 900 data points were used for training and 100 data points were used for testing from a total of 1,000 data points collected in the area surrounded by a dotted line in the map. The robot simultaneously acquired images and self-position data (x, y) at times of particle re-sampling for MCL. Words were given as location names by a user who was familiar with the experimental environment. The user gave one or more location names suitable for the place at a data point during the training. In example A, not only a name such as "front of the door" but also a name representing a space such as "entrance" and a name meaning a room such as "laboratory" were given as word information. Word information was partially supplied as training data. Five training data sets were prepared to evaluate robustness of the naming rate in training data as 1, 2, 5, 10, and 20%.

The similarity between the spatial concepts formed by the proposed method and those made by humans was evaluated in experiments of location name prediction and position category prediction based on the ground truth. The ground truth information was given for 100 test data points according to the agreement of three experts who were familiar with the environment. The hierarchy of the space in the experimental environment was defined as global, intermediate, and local.

[3] Navigation Stack: http://wiki.ros.org/navigation
[4] ROS Hydro: http://wiki.ros.org/hydro

TABLE 3 | List of location names and ground truth in the hierarchy.

Global	Laboratory		
Intermediate	Entrance	Meeting space	
Local	Front of the door	Umbrella storage	Magazine rack zone
	Chair storage	Book shelf zone	Around Skype PC
	Around the charger	Around the electric piano	Locker zone
	Front of the white board	Front of the display	Front of the table

Location names assigned to each hierarchy are shown in **Table 3**. As the ground truth for name prediction, three location names were uniformly given to each test datum considering the hierarchy to evaluate the accuracy of name prediction. As the ground truth for the position category prediction, regions corresponding to the 15 location names in **Table 3** were decided on the map. **Figure 7** shows the three regions of the "laboratory," "entrance," and "front of the table." The environment was divided into a grid of 50 units in length and 25 units in width, and the gray grids show the ground truth.

In the name prediction experiment, the accuracy of name prediction compared with the ground truth was calculated as an index of similarity. Formula (11) was used to predict names using the proposed method. The accuracy of name prediction at global, intermediate, and local levels was calculated by the following formula.

$$Accuracy = \frac{M_l}{D}, \tag{13}$$

where M_l is a number matching the predicted names with the ground truth at layer l in the test dataset and D is the number of test data. In the experiment, l was set as ($l \in \{global, intermediate, local\}$) and D was 100.

In the position category prediction experiment, the precision, recall, and F-measure of the predicted position categories compared with the ground truth were calculated as an index of similarity. In the proposed method, a position (x, y) sampled multiple times for each location name by Formula (12). The precision, recall, and F-measure of position category prediction were calculated by the following formulas.

$$Precision = \frac{T_n}{T_n + F_n} \tag{14}$$

$$Recall = \frac{T_n}{G_n} \tag{15}$$

$$F\text{-}measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision}, \tag{16}$$

where T_n is a number matching the position with the ground truth for location name n, F_n is a number that does not match the position with the ground truth, and G_n is the number of grids for the ground truth. In the experiment, n was set as ($n \in \{1, 2, \cdots, 15\}$).

In the proposed method, the hyper-parameters $\alpha, \pi, \gamma, \eta$ were set as $\alpha = 0.5, \pi = 100, \gamma = 1.0, \eta^{\upsilon} = 1.0 \times 10^{-1}, \eta^{p} = 1.0 \times 10^{-3}, \eta^{w} = 1.0 \times 10^{-2}$, respectively. The path c and category z of each data were trained with the hyper-parameters. In the experiment, the dimensions of the information vectors w^{υ}, w^{p}, and w^{w} were 1,000, 64, and 15, respectively.

4.3. Baseline Methods

The most frequent class, nearest neighbor method, multimodal hierarchical Dirichlet process (HDP), and spatial concept formation model were used as baseline methods for evaluating the performance of the proposed method in the name prediction and position category prediction experiments. In the latter, the sampling of position for each location name was performed 100 times.

4.3.1. Most Frequent Class

The training dataset $D = \{d_1, d_2, \cdots, d_I\}$ is used in this method. The datum d_i consists of the position information $p_i = (x_i, y_i)$ and the word information w_i, which is a set of location names. The frequency cnt_{n_j} of each location name $n_j (j \in \{1, 2, \cdots, 15\})$ is counted in the training dataset D. The location name n_j is classified into three clusters by k-means ($k = 3$) based on cnt_{n_j}. The three clusters of location names are C_{global}, $C_{intermediate}$, and C_{local} in descending order of the frequency of the location name based on the assumption that global location names are more frequent than local location names. In the training dataset D, if a datum d_i includes a location name in C_{global}, $C_{intermediate}$, and C_{local}, the datum d_i is set as a global dataset D_g, an intermediate dataset D_i, and a local dataset D_l. The location names in the global, intermediate, and local levels are predicted as the most frequent location name in each dataset D_g, D_i, and D_l, respectively.

In the position category prediction, the positions are predicted by sampling the position information $\hat{p}$ randomly from the datasets $D_{g,f}$, $D_{i,f}$, and $D_{l,f}$, which have the most frequent location names in each dataset D_g, D_i, and D_l, respectively. The sampling of position information for each location name was performed 100 times.

4.3.2. Nearest Neighbor (Position and Word)

The nearest neighbor method (Friedman et al., 1977) discriminates data based on Euclidean distance. A datum d_i involves position information $p_i = (x_i, y_i)$ and word information w_i. w_i consists of a set of location names that obtained at a position p_i in the training. For example, w_i at data point B in **Figure 6** contains the following location names: "Meeting space," "Book shelf zone," and "Around the electric piano." If position information p_t is observed, then word information $\hat{w}_t$ is calculated with the training dataset $D = \{(p_1, w_1), (p_2, w_2), \cdots, (p_I, w_I)\}$ by the following formulas.

$$k = \underset{1 \le i \le I}{\arg\min} \, |p_t - p_i| \tag{17}$$

$$\hat{w}_t = w_k \tag{18}$$

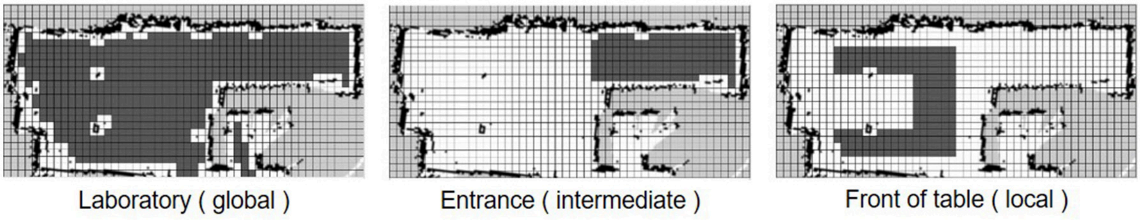


FIGURE 7 | Examples of ground truth for regions where the location names are at the global, intermediate, and local levels. The area is mapped by a grid of 50 columns and 25 rows. The region of ground truth is represented by the gray grids.

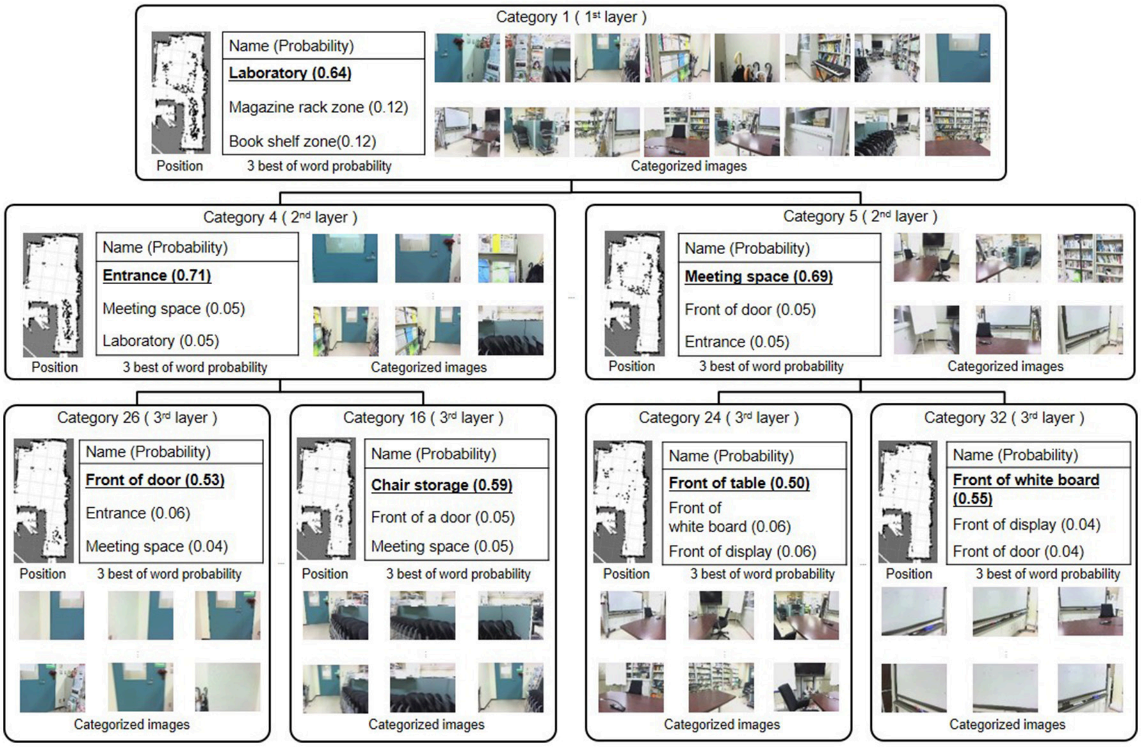


FIGURE 8 | Hierarchical spatial concept formed by the proposed method.

The location name $\hat{n}$ can be predicted by randomly selecting a location name from location names in $\hat{w}_t$ of the nearest data point.

If word information w_t is observed, then position information $\hat{p}_t$ is randomly selected from dataset D_{n_t}, which is a set of data $d_i = (p_i, w_i)$ satisfying the formula $w_i \in w_t$. The sampling of position information for each location name was performed 100 times.

4.3.3. Nearest Neighbor (Vision, Position and Word)

This method is used only in the name prediction experiment. A datum d_i includes vision information v_i, position information $p_i = (x_i, y_i)$ and word information w_i. υ_i is a value calculated by Formula (1) at a position p_i during training. w_i consists of a set of location names that are obtained at a position p_i during the training. If the vision information v_t and the position information p_t are observed, then the word information $\hat{w}_t$ can be calculated with the training dataset $D = \{(\upsilon_1, p_1, w_1), (\upsilon_2, p_2, w_2), \cdots, (\upsilon_I, p_I, w_I)\}$ by using the following formulas.

$$k = \underset{1 \le i \le I}{\arg\min} (\alpha |v_t - v_i| + (\alpha - 1)|p_t - p_i|) \tag{19}$$

$$\hat{w}_t = w_k \tag{20}$$

where α is the weight coefficient between vision and position information. α was set as 0.3 in the validation dataset empirically. The location name $\hat{n}$ can be predicted by randomly selecting a location name from the location names in $\hat{w}_t$ of the nearest data point.

4.3.4. Multimodal HDP

Multimodal HDP (Nakamura et al., 2011) enables the multimodal handling of HDP (Teh et al., 2005), which is a method of categorizing observed data based on a Bayes generative model, in the topic distribution of latent Dirichlet allocation (LDA) as HDP. The graphical model and definition of variables in the multimodal HDP are shown in the Supplementary Material. Here, multimodal HDP was trained using vision, position, and word information. If vision information w_t^v and position information $\boldsymbol{w}_t^p$ are observed at a time t, then the posterior probability of word information w_t^w can be calculated by the following formula:

$$p(w_t^w|\hat{\boldsymbol{z}}, \boldsymbol{w}^w, \boldsymbol{w}^\upsilon, \boldsymbol{w}^p, w_t^\upsilon, w_t^p, \pi, \eta^w, \eta^\upsilon, \eta^p) = \\ \sum_{z_t} p(w_t^w|z_t, \hat{\boldsymbol{z}}^w, \hat{\boldsymbol{c}}, \boldsymbol{w}^w, \eta^w) p(z_t|\hat{\boldsymbol{z}}^\upsilon, \hat{\boldsymbol{z}}^p, \boldsymbol{w}^\upsilon, \boldsymbol{w}^p, w_t^\upsilon, w_t^p, \pi, \eta^\upsilon, \eta^p) \tag{21}$$

The location name $\hat{n}$ can be predicted by the maximum value of the calculated posterior probability.

If word information w_t^w is obtained at a time t, then a category z_t^w can be predicted by Formula (22) and selecting position information $\hat{p}$ randomly from dataset $D_{z_t^w}$, which is a set of position data categorized into z_t^w.

$$z_t^w \sim p(z_t^w|\boldsymbol{z}_{-t}^w, w_t^w, \boldsymbol{w}^w, \boldsymbol{w}^\upsilon, \boldsymbol{w}^p, \eta^w, \eta^\upsilon, \eta^p, \pi) \tag{22}$$

The sampling of position information for each location name was performed 100 times. In the multimodal HDP, the hyper-parameters π, η were set as $\pi = 50, \eta^\upsilon = 5.0 \times 10^{-1}, \eta^p = 1.0 \times 10^{-1}, \eta^w = 1.0 \times 10^{-1}$ in the validation dataset. The category z of each data is trained with the hyper-parameters.

4.3.5. Spatial Concept Formation

Spatial concept formation (SpCoFo)[5] is a model that integrates name modalities into the spatial region learning model (Ishibushi et al., 2015). The model forms concepts from multimodal information and predicts unobserved information. The graphical model and definition of variables in the spatial concept formation model are shown in the Supplementary Material. The posterior probability of word information w_t^n after obtaining vision information w_t^υ and position information p_t was calculated by the following formula:

$$\begin{aligned} p(w_t^n|p_t, w_t^\upsilon) &= \sum_{z_t} p(w_t^n|z_t) p(z_t|p_t, w_t^\upsilon) \\ &= \sum_{z_t} p(w_t^n|\beta_{z_t}^n) p(p_t|\mu_{z_t}, \Sigma_{z_t}) p(w_t^\upsilon|\beta_{z_t}^\upsilon) \end{aligned} \tag{23}$$

The location name $\hat{n}$ can be predicted by the maximum value of the calculated posterior probability.

The prediction of position $\hat{p_t}$ after obtaining word information w_t^n was calculated by estimating a category z_t and sampling position information $\hat{p}$ using the following formulas.

$$\begin{aligned} z_t &= \arg\max_{z_t} p(z_t|w_t^n) \\ \hat{p_t} &\sim p(p_t|\mu_{z_t}, \Sigma_{z_t}) \end{aligned} \tag{24}$$

The sampling of position information for each location name was performed 100 times. In the spatial concept formation, the hyper-parameters π, η, μ_0, κ_0, ψ_0, and ν_0 were set as $\pi = 50$, $\eta^\upsilon = 5.0 \times 10^{-1}$, $\eta^w = 1.0 \times 10^{-1}$, $\mu_0 = (x_{center}, y_{center})$, $\kappa_0 = 3.0 \times 10^{-2}$, $\psi_0 = diag[0.05, 0.05, 0.05, 0.05]$, and $\nu_0 = 15$ in the validation dataset, respectively. (x_{center}, y_{center}) indicates the center of the map. The category z of each data is trained with the hyper-parameters.

4.4. Experimental Results

4.4.1. Hierarchical Space Categorization

Figure 8 shows some categories formed by the proposed method. Categorized training data at each category are shown by positions, images, and the best three examples from the word probability. The category corresponds to the formed spatial concept. Each category was classified into an appropriate layer in the hierarchy of spatial concepts. One, four, and 28 categories were classified into the 1st, 2nd, and 3rd layers, respectively. The number of categories in each layer was determined by the nCRP based on the model parameter γ, which controls the probability that the data is allocated to a new category.

The 1st layer included only category 1, into which 900 data were allocated. The high-probability word of category 1 was "laboratory," which referred to the entire experimental environment. Since category 1 contains all the location names, the probabilities for location names becomes relatively low. Nonetheless, the proposed method was able to learn "laboratory," which was given only about 10% to the training dataset, with high probability compared to the second candidate. In the 2nd layer, 343 data in the vicinity of the entrance in the experimental environment were allocated into category 4. The location name of category 4 with the greatest probability was "entrance." The 389 data in the region deeper than the entrance in the experimental environment were categorized into category 5, in which "meeting space" had the greatest probability. In the 3rd layer, the data categorized into categories 4 and 5 in the second layer were further, more finely categorized. In categories 26 and 16, which were formed under category 4, "front of the door" and "front of the chair storage" had the greatest probabilities, respectively. 53 and 81 data were allocated into categories 26 and 16, respectively. Position and image data corresponding to "front of the door" and "front of the chair storage" were finely allocated. These results demonstrated that the proposed method can form not only categories in a lower layer such as "front of the chair storage" and "front of the door" but also categories at higher layers such as "entrance" and "laboratory," and can form its inclusion relations as a hierarchical structure.

[5] Spatial Concept Formation: https://github.com/EmergentSystemLabStudent/Spatial_Concept_Formation

TABLE 4 | Mutual information for categorization of location names when changing the number of layers in hLDA with word information and the proposed method with vision, position, and word information.

Method	Modality	2 layers	3 layers	4 layers	5 layers
hLDA	Word	0.87	0.71	0.44	0.41
Proposed method	Vision, position, and word	**0.97**	**1.28**	**0.94**	**0.89**

Mutual information was calculated by Formula 25. Underlined and bold values mean the maximum value in the experimental parameter.

4.4.2. Evaluation of Categorization

To evaluate the effectiveness of multimodal information on hierarchical space categorization, we compared the categorization results of using the proposed method and those obtained using hLDA, which is a hierarchical categorization method with single modality, i.e., based only on word information. Although the number of layers in ground truth in this experiment is 3, robots can not know the number of hierarchies of the spatial concepts in advance. Therefore, in the proposed method and hLDA, categorization was performed with the number of layers changed from 2 to 5. The accuracy of space categorization was evaluated by calculating mutual information between the ground truth, which consisted of a location name given by humans, and the estimated name, which was the best item in the word probability at a category allocated by the proposed method or by hLDA. Mutual information $I(E; G)$ between estimated name E and ground truth G in each layer i and j was calculated by the following formula:

$$I(E; G) = \sum_{g_j \in G} \sum_{e_i \in E} p(e_i, g_j) \log \frac{p(e_i, g_j)}{p(e_i)p(g_j)}. \qquad (25)$$

When the mutual information become high, the dependency of e_i and g_j can be regarded as high. By using mutual information, accuracy of categorization can be evaluated when the number of layers on ground truth and estimation result is different. **Table 4** shows the mutual information for categorization results between hLDA with word information and the proposed method with vision, position, and word information in the training data set. The effectiveness of multimodal information in space categorization was clarified, since the proposed method had a high level of mutual information in all layers. In addition, mutual information was maximized when using the same hierarchical number as in the ground truth. In the subsequent evaluations, the number of layers of the proposed method is set to 3.

4.4.3. Evaluation of Name Prediction and Position Category Prediction

We conducted experiments to verify whether or not the proposed method could form hierarchical spatial concepts, which enable a robot to predict location names and position categories similar to predictions made by humans. In the experiment, (1) the influence of multimodal information on the formation of a hierarchical spatial concept was evaluated by comparing the space-categorization results obtained using the proposed method and using hLDA, which is a hierarchical categorization method with single modality; (2) the similarity between the hierarchical spatial concepts formed by the proposed method and those of humans was evaluated in predicting location names and position categories. The evaluation experiments were performed by cross verification with three data sets that consist of 900 training data and 100 test data with ground truth. The experimental results are indicated by the mean and standard deviation in the three data sets.

To verify whether or not the proposed method can form hierarchical spatial concepts, accuracy evaluation of name prediction and position category prediction through spatial concept use was performed. In the evaluation of name prediction, vision, position, and word information were given to the robot at the training data points. In the test data points, only vision and position information were given. Therefore, the robot has to predict the unobserved word information from the observed vision and position information. **Table 5** shows the accuracy of name prediction using the baseline methods, the proposed method, and those made by humans. The most frequent class, nearest neighbor (position and word), nearest neighbor (vision, position, and word), multimodal HDP, and spatial concept formation model were used as the baseline methods. The accuracy of name prediction was calculated by Formula (13) at global, intermediate, and local layers in ground truth. The proposed method and humans predicted location names in three layers. The results of humans consisted of the average accuracy of three subjects familiar with the experimental environment.

Compared with the accuracy obtained using the baseline methods, higher accuracies were obtained by the proposed method in the 1st, 2nd, and 3rd layers. It was assumed that weak features buried in the lower layer in the baseline methods were categorized as features of the higher layer in the proposed method. The proposed method enabled a robot to predict location names close to predictions made by humans by selecting the appropriate layer depending on the situation.

Table 6 shows the evaluation results of position category prediction using the baseline methods, the proposed method, and those made by humans. In the evaluation, the most frequent class, nearest neighbor (position and word), multimodal HDP, and spatial concept formation model were used as the baseline methods. The position category prediction was evaluated in terms of precision, recall, and F-measure, which were calculated by Formula (14).

Compared with results obtained by the baseline methods, higher values of precision and recall were obtained by the proposed method in the global and intermediate layers. In the local layer, higher values of precision and recall were obtained by Nearest neighbor and Spatial Concept Formation (SpCoFo), respectively. However, in the F-measure, which is a harmonic mean between precision and recall, the proposed method has the largest values in the global, intermediate, and local layers. The reason why the recall and F-measure values were lower than the precision is that only 100 data points were predicted and plotted for regions with 100 grids or more, as shown in **Figure 7**. In the result of F-measure, independent t-tests were performed in nine samples consisting of three data

TABLE 5 | Accuracy of name prediction using the baseline methods, the proposed method, and those made by humans; the accuracy was calculated by using Formula (13).

			Mean (s.d.)		
Method	**Modality**	**Layer**	**Global**	**Intermediate**	**Local**
Most frequent class	Position and word		**1.00 (0.00)**	0.18 (0.32)	0.09 (0.02)
Nearest neighbor	Position and word		0.12 (0.01)	0.24 (0.02)	0.20 (0.03)
Nearest neighbor	Vision, position and word		0.18 (0.03)	0.28 (0.04)	0.31 (0.04)
Multimodal HDP	Vision, position, and word		0.13 (0.02)	0.54 (0.06)	0.24 (0.07)
SpCoFo	Vision, position, and word		0.25 (0.13)	0.23 (0.15)	0.36 (0.13)
Proposed method	Vision, position, and word	1st	**1.00 (0.00)**	0.00 (0.00)	0.00 (0.00)
		2nd	0.00 (0.00)	**0.96 (0.04)**	0.01 (0.02)
		3rd	0.00 (0.00)	0.04 (0.04)	**0.55 (0.07)**
Humans		1st	1.00 (0.00)	0.00 (0.00)	0.00 (0.00)
		2nd	0.00 (0.00)	0.98 (0.02)	0.00 (0.00)
		3rd	0.00 (0.00)	0.03 (0.04)	0.74 (0.10)

The accuracy is indicated by the mean and standard deviation (s.d.). Underlined and bold values mean the maximum value in the experimental parameter.

sets with three types of ground truth: global, intermediate, and local. In the proposed method, the p-values of the Most frequent class, Nearest neighbor, multimodal HDP, and SpCoFo were 0.00012, 0.00004, 0.00003, and 0.00051, respectively, and significant differences were observed with ($p < 0.05$). As the reason why the result of humans were not perfect, some errors were found in the boundary of the place. For example, the boundary between "Book shelf zone" and "front of the table," and the edge of the region called "front of the door" were different depending on the human. The centricity of the place is consistent, but the region includes ambiguity even among humans. The experimental results show that the proposed method enabled a robot to predict position categories closer to predictions made by humans than possible using the baseline methods.

In the experiments for location name and position category prediction, the proposed method showed higher performance than the baseline methods. In the baseline methods, i.e., multimodal HDP and SpCoFo, since the feature space is classified uniformly, the location concepts are formed non-hierarchically. For example, an upper concept, e.g., meeting space, is embedded in the lower concepts, e.g., front of the table and front of the display. Therefore, the place called "Meeting space" is learned as a place different from the places called "front of the table" and "front of the display." Since the proposed method forms concepts by extracting the similarity of knowledge in the upper concept, it is possible to form an upper concept without interfering with the formation of the lower concept. For this reason, the proposed method was able to show high performance in the experiments of name and position category prediction with global, intermediate, and local.

In human-robot interactions in home environments, location names as word information are given to only a part of the training data from a user. We evaluated the robustness of the proposed method in terms of the naming rate in order to verify how name and position category prediction performance changes with decreasing naming rate. In this experiment, the formation of spatial concepts using the proposed method was performed using the training data with the naming rate changed to 1, 2, 5, 10, and 20% successively. The naming rates of 1 or 20% mean that 9 or 180 of the 900 training data contained location names, while the remaining data did not contain any location name. **Table 7** shows the accuracy of name prediction and the F-measure of position category prediction for each naming rate. In the results of name prediction and position category prediction, it was confirmed that learning progresses in the global layer earlier than in the intermediate and local layers. It was clarified that overall prediction ability did not decrease greatly owing to the decreased naming rate, but gradually decreased from the lower layer. In this experiment, we performed spatial concept formation without prior knowledge in only one environment, but it is possible to increase learning efficiency by giving parameters of models estimated in other environments as prior probabilities. The transfer learning of spatial concepts will be performed in the future.

5. APPLICATION EXAMPLES FOR HUMAN SUPPORT ROBOTS

Application examples of the hierarchical spatial concept using the proposed method are demonstrated in this section. We implemented the proposed method for the Toyota human support robot (HSR)[6] and created application examples in which the robot moves based on human linguistic instructions and describes its self-position linguistically in an experimental field assuming a home environment.

The home environment and the robot used are shown in **Figure 9**. There were two tables as shown in **Figure 9A**, A and B. In the environment, whether the robot could move based on linguistic instructions including the hierarchical structure of spaces such as "front of the table in the living room" and "front of the table in the dining room" was verified. In **Figure 9B**, an

[6] Toyota Global Site—Partner Robot Family: http://www.toyota-global.com/innovation/partner_robot/family_2.html

TABLE 6 | Precision, recall, and F-measure evaluation of position category prediction using the baseline methods, the proposed method, and those made by humans in global, intermediate, and local; the precision, recall, and F-measure were calculated by using Formula (14).

Method	Precision			Recall			F-measure		
	Global	Intermediate	Local	Global	Intermediate	Local	Global	Intermediate	Local
Most frequent class	**1.00 (0.01)**	0.49 (0.01)	0.37 (0.03)	0.12 (0.02)	0.17 (0.02)	0.15 (0.03)	0.22 (0.03)	0.25 (0.02)	0.20 (0.02)
Nearest neighbor	**1.00 (0.00)**	0.93 (0.03)	**0.67 (0.03)**	0.12 (0.03)	0.26 (0.04)	0.23 (0.04)	0.22 (0.04)	0.41 (0.05)	0.33 (0.03)
Multimodal HDP	**1.00 (0.00)**	0.95 (0.02)	0.53 (0.03)	0.12 (0.01)	0.26 (0.04)	0.26 (0.02)	0.21 (0.02)	0.40 (0.05)	0.33 (0.02)
SpCoFo	0.82 (0.00)	0.62 (0.04)	0.35 (0.04)	0.16 (0.01)	0.32 (0.02)	**0.38 (0.04)**	0.27 (0.02)	0.42 (0.01)	0.35 (0.04)
Proposed method	**1.00 (0.00)**	**0.96 (0.03)**	0.59 (0.05)	**0.18 (0.01)**	**0.34 (0.02)**	0.36 (0.04)	**0.30 (0.02)**	**0.50 (0.02)**	**0.43 (0.01)**
Humans	1.00	0.99	0.76	0.19	0.50	0.49	0.32	0.65	0.56

In the experiment, the modalities of the nearest neighbor were position and word. The results are indicated by the mean and standard deviation as mean (s.d.). Underlined and bold values mean the maximum value in the experimental parameter.

TABLE 7 | Robustness evaluation of the proposed method with respect to naming rate: accuracy in name prediction indicates the maximum value of the three layers.

Naming rate	Name prediction (accuracy)			Position prediction (F-measure)		
	Global	Intermediate	Local	Global	Intermediate	Local
1%	1.00	0.68	0.14	0.29	0.46	0.30
2%	1.00	0.77	0.26	0.29	0.47	0.31
5%	1.00	0.92	0.35	0.28	0.36	0.37
10%	1.00	0.92	0.58	0.30	0.46	0.37
20%	1.00	0.92	0.63	0.31	0.50	0.44
Humans	1.00	0.96	0.76	0.32	0.65	0.56

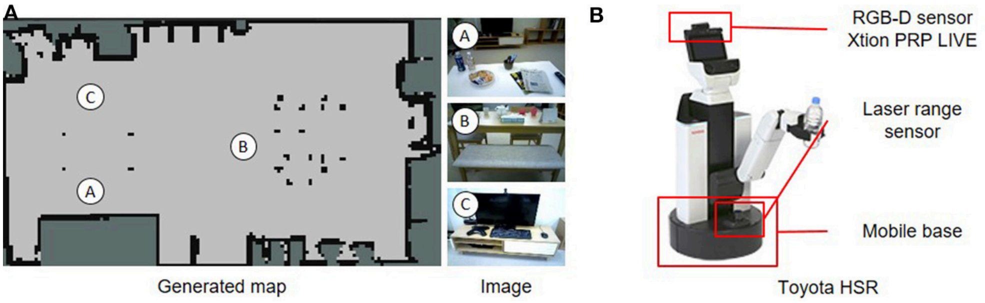


FIGURE 9 | Experimental environment and robot for demonstrating application examples. **(A)** Environments A, B, and C in the generated map show positions in which images were captured. **(B)** Human support robot produced by the Toyota Company.

RGB-D sensor and a laser range sensor were used to capture images and to estimate self-position, respectively. The packages[7]: hector_slam and omni_base for mapping, localization, and moving were used with ROS Indigo[8] to navigate the robot to the predicated position.

The robot collected 715 training data consisting of images, positions, and word information and formed a hierarchical spatial concept using the proposed method. Location names were given to 20% of total training data. Rospeex (Sugiura and Zettsu, 2015) was used to recognize human speech instructions and convert them into text information. In the experiment, the dimensions of the information vectors w^v, w^p, and w^w were 1,000, 64, and 16, respectively.

The two places predicted by Formula (12) based on the speech instructions, i.e., "go to the front of the table in the living room" and "go to the front of the table in the dining room" are shown in **Figures 10A,B**, respectively. Predicted position categories indicated by red dots show that the "front of the table in the living room" and the "front of the table in the dining room" were recognized as different places using the space concept in the higher layer.

[7] hector_slam: http://wiki.ros.org/hector_slam
[8] ROS Indigo: http://wiki.ros.org/indigo

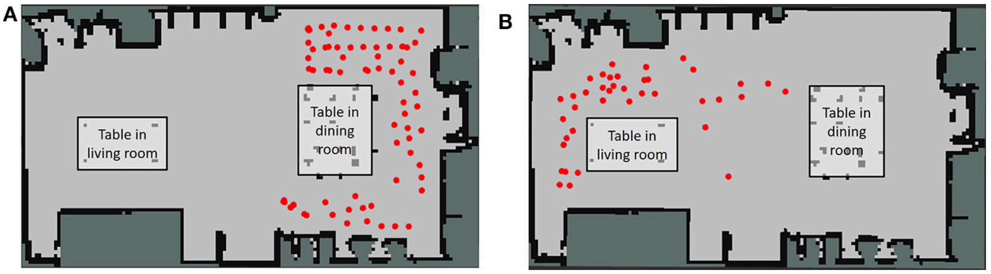


FIGURE 10 | Position category prediction using a hierarchical structure based on linguistic instructions from the user. **(A)** Positions for the front of the table in the living room. **(B)** Positions for the front of the table in the dining room.

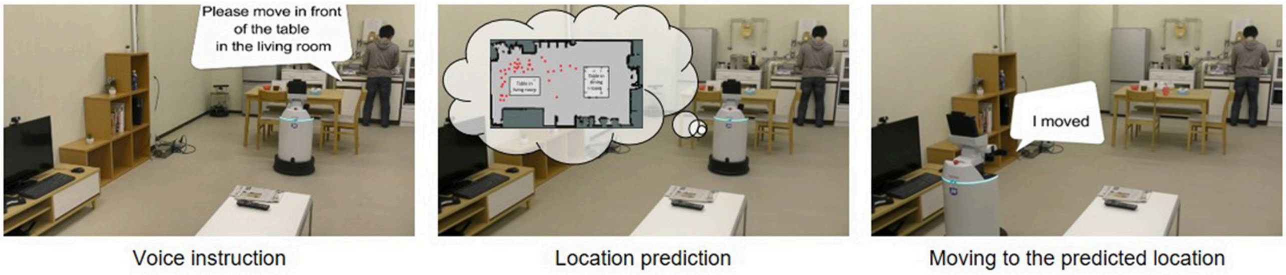
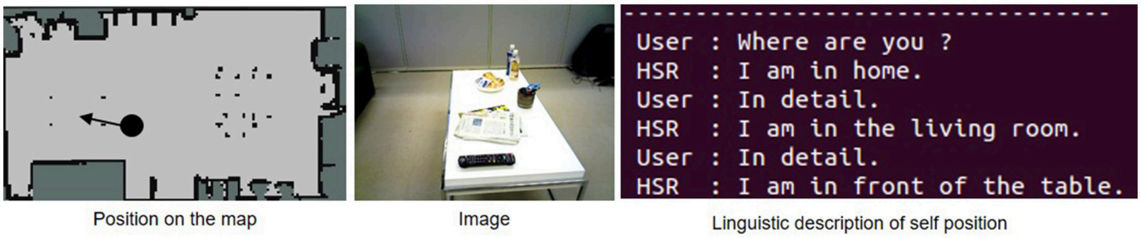

Please move in front of the table in the living room
I moved
Voice instruction
Location prediction
Moving to the predicted location

FIGURE 11 | Movement based on speech instructions from the user through the hierarchical spatial concept.



```
------------------------------------------
User  : Where are you ?
HSR   : I am in home.
User  : In detail.
HSR   : I am in the living room.
User  : In detail.
HSR   : I am in front of the table.
```

FIGURE 12 | Linguistic description of self-position based on communication between the user and the robot using the hierarchical spatial concept.

Figure 11 shows how the robot moved based on human speech instructions in the experiment. The robot recognized human speech instructions using rospeex and predicted position categories with the Formula (12) using a hierarchical spatial concept. It moved to the instructed place by sampling randomly from the predicted positions. **Figure 12** shows an application example in which the robot described its self-position linguistically. The robot observed its self-position and image and predicted the name of its self-position by calculating Formula 11 using the hierarchical spatial concept. As shown in the left side of **Figure 12**, the proposed method enabled the robot to describe its self-position linguistically with different layers. We demonstrated application examples using the formed hierarchical spatial concept in the service scene in a home environment. The movie of the demonstration and training dataset can be found at the URL[9].

6. CONCLUSIONS

We assumed that a computational model that considers the hierarchical structure of space enables robots to predict the name and position of a space close to the corresponding prediction by humans. In our assumptions, we proposed a hierarchical spatial concept formation method based on a Bayesian generative model with multimodal information, i.e., vision, position, and word information, and developed a robot that can predict unobserved location names and position categories based on observed information using the formed hierarchical spatial concept. We conducted experiments to form a hierarchical spatial concept using a robot and evaluated its ability in name prediction and position category prediction.

The experimental results for name and position category prediction demonstrated that, relative to baseline methods, the proposed method enabled the robot to predict location names and position categories closer to predictions made by

[9] Multimedia - emlab page: https://emlab.jimdo.com/multimedia/

humans. Application examples using the hierarchical spatial concept in a home environment demonstrated that a robot could move to an instructed place based on human speech instructions and describe its self-position linguistically through the formed hierarchical spatial concept. The experimental results and application example demonstrated that the proposed method enabled the robot to form spatial concepts in abstract layers and use the concepts for human-robot communications in a home environment. This study showed that it the name and position of a location could be predicted, even in a home, using generalized spatial concepts. Furthermore, by conducting additional learning in each house, a spatial concept adapted to the environment can be formed.

The limitation of this study is as follows. In the feature extraction of the position information, hierarchical k-means method was utilized to convert the position information (x, y) into the position histogram. In the experiment, 389 and 511 data were allocated to two clusters at the top layer c_1. In the bottom layer c_6, the number and standard deviation of the data allocated to each of the 64 clusters were 14.1 and 12.2, respectively. There is some bias between the clusters. The hierarchical k-means makes it possible to convert the position information into the position histogram including hierarchical spatial features. However, nearby data points at a classification boundary, which are classified into different clusters on a high level, are regarded as very different. We are considering a method to reduce bias in space while maintaining hierarchical features of space. As for the number of location names, at section 4 and 5 in the experiments, the numbers of location names were 15 and 16, respectively. The number of location names increases with increase in the numbers of teachings and users. If the robot learns the location names from several users over a long term, an algorithm to remove location names with low probability of observation is needed in order to improve the learning efficiency.

As future work, we will generalize the spatial concepts for various environments and perform transition learning of spatial concepts with the generalized spatial concepts as prior knowledge.

AUTHOR CONTRIBUTIONS

YH designed the study, and wrote the initial draft of the manuscript. HK and MI contributed to analysis and interpretation of data, and assisted in the preparation of the manuscript. TT has contributed to data collection and interpretation, and critically reviewed the manuscript. All authors approved the final version of the manuscript, and agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

FUNDING

This work was supported by MEXT/JSPS KAKENHI Grant Number JP17H06383 in #4903 (Evolinguistics), JP16K16133 and JPMJCR15E3.

ACKNOWLEDGMENTS

We would like to thank Dr. Takayuki Nagai and Dr. Tomoaki Nakamura for sharing their source code with us.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2018.00011/full#supplementary-material

REFERENCES

Aldous, D. J. (1985). "Exchangeability and related topics," in *École d'Été de Probabilités de Saint-Flour XIII–1983*, Lecture Notes in Mathematics, Vol. 1117 (Berlin; Heidelberg: Springer), 1–198.

Ando, Y., Nakamura, T., and Nagai, T. (2013). "Formation of hierarchical object concept using hierarchical latent dirichlet allocation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Tokyo), 2272–2279.

Blei, D. M., Griffiths, T. L., and Jordan, M. I. (2010). The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *J. ACM* 57, 7:1–7:30. doi: 10.1145/1667053.1667056

Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). "Monte Carlo localization for mobile robots," in *Proceedings of 1999 IEEE International Conference on Robotics and Automation* (Detroit, MI), 1322–1328.

Durrant-Whyte, H., and Bailey, T. (2006). Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* 13, 99–110. doi: 10.1109/MRA.2006.1638022

Espinace, P., Kollar, T., Roy, N., and Soto, A. (2013). Indoor scene recognition by a mobile robot through adaptive object detection. *Robot. Auton. Syst.* 61, 932–947. doi: 10.1016/j.robot.2013.05.002

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.* 3, 209–226. doi: 10.1145/355744.355745

Galindo, C., Madrigal, J. A., Gonzalez, J., and Saffiotti, A. (2008). Robot task planning using semantic maps. *Robot. Auton. Syst.* 56, 955–966. doi: 10.1016/j.robot.2008.08.007

Granda, N. C., Rogers, J. G., Trevor, A. J., and Christensen, H. I. (2010). "Semantic map partitioning in indoor environments using regional analysis," in *IEEE International Conference on Intelligent Robots and Systems* (Taipei), 1451–1456.

Gu, Z., Taguchi, R., Hattori, K., Hoguro, M., and Umezaki, T. (2016). "Learning of relative spatial concepts from ambiguous instructions," in *13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*, Vol. 49 (Kyoto), 150–153.

Hagiwara, Y., Inoue, M., and Taniguchi, T. (2016). "Place concept learning by hMLDA based on position and vision information," in *13th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*, Vol. 49 (Kyoto), 216–220.

Harris, Z. (1954). Distributional structure. *Word* 10, 146–162. doi: 10.1080/00437956.1954.11659520

Heath, S., Ball, D., and Wiles, J. (2016). Lingodroids: cross-situational learning for episodic elements. *IEEE Trans. Cogn. Dev. Syst.* 8, 3–14. doi: 10.1109/TAMD.2015.2442619

Ishibushi, S., Taniguchi, A., Takano, T., Hagiwara, Y., and Taniguchi, T. (2015). "Statistical localization exploiting convolutional neural network for an autonomous vehicle," in *41th Annual Conference of the IEEE Industrial Electronics Society (IECON)* (Yokohama), 1369–1375.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., et al. (2014). "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia* (ACM) (Orlando, FL), 675–678.

Kostavelis, I., and Gasteratos, A. (2013). Learning spatially semantic representations for cognitive robot navigation. *Robot. Auton. Syst.* 61, 1460–1475. doi: 10.1016/j.robot.2013.07.008

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (Lake Tahoe, NV), 1097–1105.

Mozos, O. M., and Burgard, W. (2006). "Supervised learning of topological maps using semantic information extracted from range data," in *IEEE International Conference on Intelligent Robots and Systems* (Beijing), 2772–2777.

Nakamura, T., Nagai, T., and Iwahashi, N. (2011). "Multimodal categorization by hierarchical dirichlet process," in *Proceedings of 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Francisco, CA), 1520–1525.

Nielsen, C. W., Ricks, B., Goodrich, M. A., Bruemmer, D., Few, D., and Few, M. (2004). "Snapshots for semantic maps," in *IEEE International Conference on Systems*, Vol. 3 (The Hague), 2853–2858.

Pitman, J. (2006). *Combinatorial Stochastic Processes*. Lecture Notes in Mathematics. Berkeley, CA: Springer-Verlag.

Pitman, J. (2002). *Combinatorial Stochastic Processes*. Technical Report of Department of Statistics, UC Berkeley, 2002. Lecture notes for St. Flour Course, 621.

Pronobis, A., Caputo, B., Jensfelt, P., and Christensen, H. I. (2006). "A discriminative approach to robust visual place recognition," in *IEEE International Conference on Intelligent Robots and Systems* (Beijing), 3829–3836.

Pronobis, A., and Jensfelt, P. (2012). "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *IEEE International Conference on Robotics and Automation* (Saint Paul, MN), 3515–3522. doi: 10.1109/ICRA.2012.6224637

Ranganathan, A., and Dellaert, F. (2007). "Semantic modeling of places using objects," in *Proceedings of the 2007 Robotics: Science and Systems Conference*, Vol. 3 (Atlanta, GA), 27–30.

Rangel, J. C., Martínez-Gómez, J., García-Varea, I., and Cazorla, M. (2017). Lextomap: lexical-based topological mapping. *Adv. Robot.* 31, 268–281. doi: 10.1080/01691864.2016.1261045

Rusu, R. B., Marton, Z. C., Blodow, N., Holzbach, A., and Beetz, M. (2009). "Modelbased and learned semantic object labeling in 3d point cloud maps of kitchen environments," in *IEEE International Conference on Intelligent Robots and Systems* (St. Louis, MO), 3601–3608.

Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., and Beetz, M. (2008). Towards 3d point cloud based object maps for household environments. *Robot. Auton. Syst.* 56, 927–941. doi: 10.1016/j.robot.2008.08.005

Sugiura, K., and Zettsu, K. (2015). "Rospeex: a cloud robotics platform for human-robot spoken dialogues," in *Proceedings of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Hamburg), 6155–6160.

Taniguchi, A., Taniguchi, T., and Inamura, T. (2016). Spatial concept acquisition for a mobile robot that integrates self-localization and unsupervised word discovery from spoken sentences. *IEEE Trans. Cogn. Dev. Syst.* 8, 285–297. doi: 10.1109/TCDS.2016.2565542

Taniguchi, A., Yoshizaki, H., Inamura, T., and Taniguchi, T. (2014). Research on simultaneous estimation of self-location and location concepts. *Trans. Inst. Syst. Control Inform. Eng.* 27, 166–177. doi: 10.5687/iscie.27.166

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2005). "Sharing clusters among related groups: hierarchical dirichlet processes," in *Advances in Neural Information Processing Systems 17*, eds L. K. Saul, Y. Weiss, and L. Bottou (Long Beach, CA: MIT Press), 1385–1392.

Zender, H., Mozos, O. M., Jensfelt, P., Kruijff, G. J., and Burgard, W. (2008). Conceptual spatial representations for indoor mobile robots. *Robot. Auton. Syst.* 56, 493–502. doi: 10.1016/j.robot.2008.03.007

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 22 May 2018
doi: 10.3389/fnbot.2018.00022

Multimodal Hierarchical Dirichlet Process-Based Active Perception by a Robot

Tadahiro Taniguchi[1]*, *Ryo Yoshino*[1] and *Toshiaki Takano*[2]

[1] *Emergent Systems Laboratory, College of Information Science and Engineering, Ritsumeikan University, Ksatsu, Japan,* [2] *Adaptive Systems Laboratory, Department of Computer Science, Shizuoka Institute of Science and Technology, Fukuroi, Japan*

In this paper, we propose an active perception method for recognizing object categories based on the multimodal hierarchical Dirichlet process (MHDP). The MHDP enables a robot to form object categories using multimodal information, e.g., visual, auditory, and haptic information, which can be observed by performing actions on an object. However, performing many actions on a target object requires a long time. In a real-time scenario, i.e., when the time is limited, the robot has to determine the set of actions that is most effective for recognizing a target object. We propose an active perception for MHDP method that uses the information gain (IG) maximization criterion and lazy greedy algorithm. We show that the IG maximization criterion is optimal in the sense that the criterion is equivalent to a minimization of the expected Kullback–Leibler divergence between a final recognition state and the recognition state after the next set of actions. However, a straightforward calculation of IG is practically impossible. Therefore, we derive a Monte Carlo approximation method for IG by making use of a property of the MHDP. We also show that the IG has submodular and non-decreasing properties as a set function because of the structure of the graphical model of the MHDP. Therefore, the IG maximization problem is reduced to a submodular maximization problem. This means that greedy and lazy greedy algorithms are effective and have a theoretical justification for their performance. We conducted an experiment using an upper-torso humanoid robot and a second one using synthetic data. The experimental results show that the method enables the robot to select a set of actions that allow it to recognize target objects quickly and accurately. The numerical experiment using the synthetic data shows that the proposed method can work appropriately even when the number of actions is large and a set of target objects involves objects categorized into multiple classes. The results support our theoretical outcomes.

Keywords: active perception, cognitive robotics, topic model, multimodal machine learning, submodular maximization

Edited by:
Minoru Asada,
Osaka University, Japan

Reviewed by:
Shingo Murata,
National Institute of Informatics, Japan
J. Michael Herrmann,
University of Edinburgh,
United Kingdom
Jivko Sinapov,
Tufts University, United States

****Correspondence:***
Tadahiro Taniguchi
taniguchi@ci.ritsumei.ac.jp

Received: *24 August 2017*
Accepted: *23 April 2018*
Published: *22 May 2018*

Citation:
Taniguchi T, Yoshino R and Takano T (2018) Multimodal Hierarchical Dirichlet Process-Based Active Perception by a Robot. Front. Neurorobot. 12:22. doi: 10.3389/fnbot.2018.00022

1. INTRODUCTION

Active perception is a fundamental component of our cognitive skills. Human infants autonomously and spontaneously perform actions on an object to determine its nature. The sensory information that we can obtain usually depends on the actions performed on the target object. For example, when people find a gift box placed in front of them, they cannot perceive its weight without holding the box, and they cannot determine its sound without hitting or shaking it. In other words, we can obtain sensory information about an object by selecting and executing actions to manipulate it. Adequate action selection is important for recognizing objects quickly and accurately. This example about a human also holds for a robot. An autonomous robot that moves and helps people in a living environment should also select adequate actions to recognize target objects. For example, when a person asks an autonomous robot to bring an empty plastic bottle, the robot has to examine many objects by applying several actions (**Figure 1**). This type of information is important, because our object categories are formed on the basis of multimodal information, i.e., not only visual information is used, but also auditory, haptic, and other information. Therefore, a computational model of the active perception should be consistently based on a computational model for multimodal object categorization and recognition.

In spite of the wide range of studies about active perception (e.g., Borotschnig et al., 2000; Dutta Roy et al., 2004; Eidenberger and Scharinger, 2010; Krainin et al., 2011; Ferreira et al., 2013) and multimodal categorization for robots (e.g., Nakamura et al., 2007, 2011a; Sinapov and Stoytchev, 2011; Celikkanat et al., 2014; Sinapov et al., 2014), active perception methods for a robot, i.e., action selection methods for perception for unsupervised multimodal categorization, have not been sufficiently explored (see section 2).

This paper considers the active perception problem for unsupervised multimodal object categorization under the condition that a robot has already obtained several action primitives that are used to examine target objects. In the context of this study, we need to study active perception on an unsupervised multimodal categorization method having generality as much as possible because it is believed that unsupervised multimodal categorization is important for future language learning by robots, and the findings obtained in this study should be able to be applied to other unsupervised multimodal categorization models. It was suggested that a child forms a category based on his/her sensorimotor experience before learning a word for the category in a Bayesian manner, and learning the word is a matter of attaching a new label to this preexisting category (Kemp et al., 2010). The multimodal hierarchical Dirichlet process (MHDP) is a mathematically very general and sophisticated nonparametric Bayesian multimodal categorization method. Therefore, we adopt MHDP proposed by Nakamura et al. (2011b) as a representative computational model for unsupervised multimodal object categorization.

We develop an active perception method based on the MHDP in this paper. The MHDP is a sophisticated, fully Bayesian, probabilistic model for multimodal object categorization (Nakamura et al., 2011b) that is developed by enabling hierarchical Dirichlet process (HDP) (Teh et al., 2006) to have multimodal emission distributions corresponding to multiple sensor information[1]. Nakamura et al. (2011b) showed that the MHDP enables a robot to form object categories using multimodal information, i.e., visual, auditory, and haptic information, in an unsupervised manner. The MHDP can estimate the number of object categories as well because of the nature of Bayesian nonparametrics.

This paper describes a new MHDP-based active perception method for multimodal object recognition based on object categories formed by a robot itself. We found that an active perception method that has a good theoretical nature, i.e., the performance of the greedy algorithm is theoretically guaranteed (see section 4), can be derived for MHDP. Our formulation is based on a hierarchical Bayesian model. If a cognitive system of a robot is modeled by using hierarchical Bayesian model, a recognition state are usually represented by posterior distribution over latent variables, e.g., object categories. The purpose of an active perception is to infer appropriate posterior distribution with a small number of actions. In our approach, we propose an action selection method that can reduce the distance between inferred posterior distributions and true posterior distributions.

In this study, we define the active perception problem in the context of unsupervised multimodal object categorization as following. Which set of actions should a robot take to recognize a target object as accurately as possible under the constraint that the number of actions is restricted[2]? Our MHDP-based active perception method uses an IG maximization criterion, Monte Carlo approximation, and the lazy greedy algorithm. In this paper, we show that the MHDP provides the following three advantages for deriving an efficient active perception method.

1. The *IG maximization criterion* is *optimal* in the sense that a selected set of actions minimizes the expected Kullback–Leibler (KL) divergence between the final posterior distribution estimated using the information regarding all modalities and the posterior distribution of the category estimated using the selected set of actions (see section 4.1).
2. The IG has a *submodular* and non-decreasing property as a set function. Therefore, for performance, the greedy and lazy greedy algorithms are guaranteed to be near-optimal strategies (see section 4.2).
3. A *Monte Carlo approximation* method for the IG can be derived by exploiting MHDP's properties (see section 4.3).

Although the above properties follow from the theoretical characteristics of the MHDP, this has never been pointed out in previous studies.

The main contributions of this paper are that we

- develop an MHDP-based active perception method, and
- show its effectiveness through experiments using an upper-torso humanoid robot and synthetic data.

The proposed active perception method can be used for general purposes, i.e., not only for robots but also for other target

[1] HDP is a nonparametric Bayesian extension of latent Dirichlet allocation (LDA) (Blei et al., 2003), which has been widely used for document-word clustering. The nonparametric Bayesian extension allows HDP to estimate the number of topics, i.e., clusters, as well.

[2] We can consider an extension of this problem by introducing different cost to each action, i.e., different action requires different time and energy. However, for simplicity, this paper focuses on the problem in which cost for each action is the same.

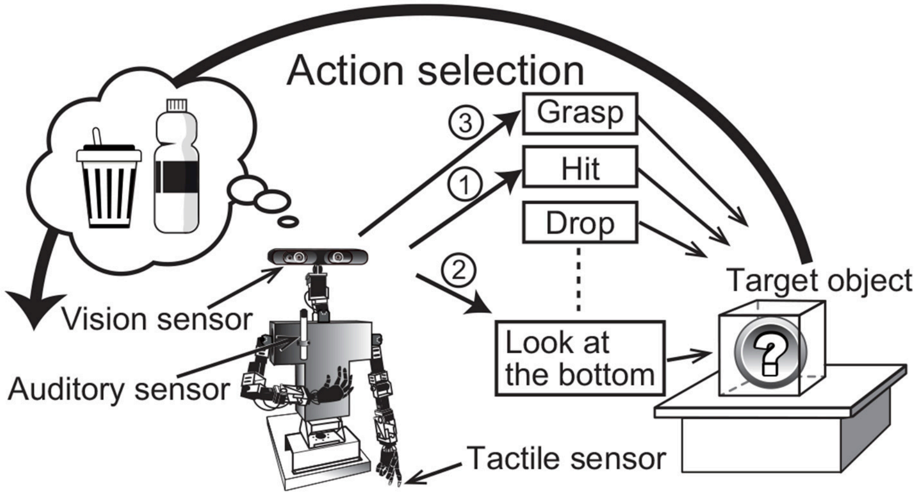


FIGURE 1 | Overview of active perception for multimodal object category recognition. The numbers attached to the arrows show a sample of the order of action selection by the robot.

domains to which the MHDP can be applied. In addition, The proposed method can be easily extended for other multimodal categorization methods with similar graphical models, e.g., multimodal latent Dirichlet allocation (MLDA) (Nakamura et al., 2009). However, in this paper, we focus on the MHDP and the robot active perception scenario, and explain our method on the basis of this task.

The remainder of this paper is organized as follows. Section 2 describes the background and work related to our study. Section 3 briefly introduces the MHDP, proposed by Nakamura et al. (2011b), which enables a robot to obtain object categories by fusing multimodal sensor information in an unsupervised manner. Section 4 describes our proposed action selection method. Section 5 discusses the effectiveness of the action selection method through experiments using an upper-torso humanoid robot. Section 6 describes a supplemental experiment using synthetic data. Section 7 concludes this paper.

2. BACKGROUND AND RELATED WORK

2.1. Multimodal Categorization

The human capability for object categorization is a fundamental topic in cognitive science (Barsalou, 1999). In the field of robotics, adaptive formation of object categories that considers a robot's embodiment, i.e., its sensory-motor system, is gathering attention as a way to solve the symbol grounding problem (Harnad, 1990; Taniguchi et al., 2016).

Recently, various computational models and machine learning methods for multimodal object categorization have been proposed in artificial intelligence, cognitive robotics, and related research fields (Roy and Pentland, 2002; Natale et al., 2004; Nakamura et al., 2007, 2009, 2011a,b, 2014; Iwahashi et al., 2010; Sinapov and Stoytchev, 2011; Araki et al., 2012; Griffith et al., 2012; Ando et al., 2013; Celikkanat et al., 2014; Sinapov et al., 2014). For example, Sinapov and Stoytchev (2011) proposed a graph-based multimodal categorization method that allows a robot to recognize a new object by its similarity to a set of familiar objects. They also built a robotic system that categorizes 100 objects from multimodal information in a supervised manner (Sinapov et al., 2014). Celikkanat et al. (2014) modeled the context in terms of a set of concepts that allow many-to-many relationships between objects and contexts using LDA.

Our focus of this paper is not a supervised learning-based, but an unsupervised learning-based multimodal categorization method and an active perception method for categories formed by the method. Of these, a series of statistical unsupervised multimodal categorization methods for autonomous robots have been proposed by extending LDA, i.e., a topic model (Nakamura et al., 2007, 2009, 2011a,b, 2014; Araki et al., 2012; Ando et al., 2013). All these methods are Bayesian generative models, and the MHDP is a representative method of this series (Nakamura et al., 2011b). The MHDP is an extension of the HDP, which was proposed by Teh et al. (2006), and the HDP is a nonparametric Bayesian extension of LDA (Blei et al., 2003). Concretely, the generative model of the MHDP has multiple types of emissions that correspond to various sensor data obtained through various modality inputs. In the HDP, observation data are usually represented as a bag-of-words (BoW). In contrast, the observation data in the MHDP use bag-of-features (BoF) representations for multimodal information. BoF is a histogram-based feature representation that is generated by quantizing observed feature vectors. Latent variables that are regarded as indicators of *topics* in the HDP correspond to *object categories* in the MHDP. Nakamura et al. (2011b) showed that the MHDP enables a robot to categorize a large number of objects in a home environment into categories that are similar to human categorization results.

To obtain multimodal information, a robot has to perform actions and interact with a target object in various ways, e.g., grasping, shaking, or rotating the object. If the number of actions and types of sensor information increase, multimodal categorization and recognition can require a longer time. When the recognition time is limited and/or if quick recognition is required, it becomes important for a robot to select a small number of actions that are effective for accurate recognition. Action selection for recognition is often called active perception. However, an active perception method for the MHDP has not been proposed. This paper aims to provide an active perception method for the MHDP.

2.2. Active Perception

Generally, active perception is one of the most important cognitive capabilities of humans. From an engineering viewpoint, active perception has many specific tasks, e.g., localization, mapping, navigation, object recognition, object segmentation, and self–other differentiation.

In machine learning, *active learning* is defined as a task in which a method interactively queries an information source to obtain the desired outputs at new data points to learn efficiently Settles (2012). Active learning algorithms select an unobserved input datum and ask a user (labeler) to provide a training signal (label) in order to reduce uncertainty as quickly as possible (Cohn et al., 1996; Muslea et al., 2006; Settles, 2012). These algorithms usually assume a supervised learning problem. This problem is related to the problem in this paper, but is fundamentally different.

Historically, active vision, i.e., active visual perception, has been studied as an important engineering problem in computer vision. Dutta Roy et al. (2004) presented a comprehensive survey of active three-dimensional object recognition. For example, Borotschnig et al. (2000) proposed an active vision method in a parametric eigenspace to improve the visual classification results. Denzler and Brown (2002) proposed an information theoretic action selection method to gather information that conveys the true state of a system through an active camera. They used the mutual information (MI) as a criterion for action selection. Krainin et al. (2011) developed an active perception method in which a mobile robot manipulates an object to build a three-dimensional surface model of it. Their method uses the IG criterion to determine when and how the robot should grasp the object.

Modeling and/or recognizing a single object as well as modeling a scene and/or segmenting objects are also important tasks in the context of robotics. Eidenberger and Scharinger (2010) proposed an active perception planning method for scene modeling in a realistic environment. van Hoof et al. (2012) proposed an active scene exploration method that enables an autonomous robot to efficiently segment a scene into its constituent objects by interacting with the objects in an unstructured environment. They used IG as a criterion for action selection. InfoMax control for acoustic exploration was proposed by Rebguns et al. (2011).

Localization, mapping, and navigation are also targets of active perception. Velez et al. (2012) presented an online planning algorithm that enables a mobile robot to generate plans that maximize the expected performance of object detection. Burgard et al. (1997) proposed an active perception method for localization. Action selection is performed by maximizing the weighted sum of the expected entropy and expected costs. To reduce the computational cost, they only consider a subset of the next locations. Roy and Thrun (1999) proposed a coastal navigation method for a robot to generate trajectories for its goal by minimizing the positional uncertainty at the goal. Stachniss et al. (2005) proposed an information-gain-based exploration method for mapping and localization. Correa and Soto (2009) proposed an active perception method for a mobile robot with a visual sensor mounted on a pan-tilt mechanism to reduce localization uncertainty. They used the IG criterion, which was estimated using a particle filter.

In addition, various studies on active perception by a robot have been conducted (Natale et al., 2004; Ji and Carin, 2006; Schneider et al., 2009; Tuci et al., 2010; Saegusa et al., 2011; Fishel and Loeb, 2012; Pape et al., 2012; Sushkov and Sammut, 2012; Gouko et al., 2013; Hogman et al., 2013; Ivaldi et al., 2014; Zhang et al., 2017). In spite of a large number of contributions about active perception, few theories of active perception for multimodal object category recognition have been proposed. In particular, an MHDP-based active perception method has not yet been proposed, although the MHDP-based categorization method and its series have obtained many successful results and extensions.

2.3. Active Perception for Multimodal Categorization

Sinapov et al. (2014) investigated multimodal categorization and active perception by making a robot perform 10 different behaviors; obtain visual, auditory, and haptic information; explore 100 different objects, and classify them into 20 object categories. In addition, they proposed an active behavior selection method based on confusion matrices. They reported that the method was able to reduce the exploration time by half by dynamically selecting the next exploratory behavior. However, their multimodal categorization is performed in a supervised manner, and the theory of active perception is still heuristic. The method does not have theoretical guarantees of performance.

IG-based active perception is popular, as shown above, but the theoretical justification for using IG in each task is often missing in many robotics papers. Moreover, in many cases in robotics studies, IG cannot be evaluated directly, reliably, or accurately. When one takes an IG criterion-based approach, how to estimate the IG is an important problem. In this study, we focus on MHDP-based active perception and develop an efficient near-optimal method based on firm theoretical justification.

3. MULTIMODAL HIERARCHICAL DIRICHLET PROCESS FOR STATISTICAL MULTIMODAL CATEGORIZATION

We assume that a robot forms object categories using the MHDP from multimodal sensory data. In this section, we briefly introduce the MHDP on which our proposed active perception method is based (Nakamura et al., 2011b). The MHDP assumes that an observation node in its graphical model corresponds to an action and its corresponding modality. Nakamura et al. (2011b) employed three observation nodes in their graphical model, i.e., haptic, visual, and auditory information nodes. Three actions, i.e., grasping, looking around, and shaking, correspond to these modalities, respectively. However, the MHDP can be easily extended to a model with additional types of sensory inputs. It is without doubt that autonomous robots will also gain more types of action for perception. For modeling more general cases, an MHDP with M actions is described in this paper. A graphical model of the MHDP is illustrated in **Figure 2**. In this section, we describe the MHDP briefly. For more details, please refer to Nakamura et al. (2011b).

The index $m \in \mathbf{M}$ ($\#(\mathbf{M}) = M$) in **Figure 2** represents the type of information that corresponds to an action for perception, e.g., hitting an object to obtain its sound, grasping an object to test its shape and hardness, or looking at all of an object by rotating it. We assume that a robot has action primitives and it can execute one of the actions by selecting the index of the action primitives. The observation $x^m_{jn} \in X^m$ is the m-th modality's n-th feature for the j-th target object. X^m represents a set of observation of m-th modality. The observation x^m_{jn} is assumed to be drawn from a categorical distribution whose parameter is θ^m_k, where k is an index of a latent topic. Each index k is drawn from a categorical distribution whose parameter is β that is drawn from a Dirichlet distribution parametrized by γ. Parameter θ^m_k is assumed to be drawn from the Dirichlet prior distribution whose parameter is α^m_0. The MHDP assumes that a robot obtains each modality's sensory information as a BoF representation. Each latent variable t^m_{jn} is drawn from a topic proportion, i.e., a parameter of a multinomial distribution, of the j-th object π_j whose prior is a Dirichlet distribution parametrized by λ.

Similarly to the generative process of the original HDP (Teh et al., 2006), the generative process of the MHDP can be described as a Chinese restaurant franchise, which is the name of a special type of probabilistic process in Bayesian nonparametrics (Teh et al., 2005). The learning and recognition algorithms are both derived using Gibbs sampling. In its learning process, the MHDP estimates a latent variable t^m_{jn} for each feature of the j-th object and a topic index k_{jt} for each latent variable t. The combination of latent variable and topic index corresponds to a topic in LDA (Blei et al., 2003). Using the estimated latent variables, the categorical distribution parameter θ^m_k and topic proportion of the j-th object π_j are drawn from the posterior distribution.

The selection procedure for latent variable t^m_{jn} is as follows. The prior probability that x^m_{jn} selects t is

$$P(t^m_{jn} = t|\lambda) = \begin{cases} \frac{\sum_m w^m N^m_{jt}}{\lambda + \sum_m w^m N^m_j - 1}, & (t = 1, \cdots, T_j), \\ \frac{\lambda}{\lambda + \sum_m w^m N^m_j - 1}, & (t = T_j + 1), \end{cases}$$

where w^m is a weight for the m-th modality, To balance the influence of different modalities, w^m are set as hyperparameters. The weight w^m increases the influence of the modality m on multimodal category formation. N^m_{jt} is the number of m-th modality observations that are allocated to t in the j-th object, and λ is a hyperparameter. In the Chinese restaurant process, if the number of observed features $N_{jt} = \sum_m w^m N^m_{jt}$ that are allocated to t increases, the probability at which a new observation is allocated to the latent variable t increases. Using the prior distribution, the posterior probability that observation x^m_{jn} is allocated to the latent variable t becomes

$$\begin{aligned} P(t^m_{jn} = t|X^m, \lambda) &= \frac{P(x^m_{jn}|X^m_{k = k_{jt}})P(t^m_{jn} = t|\lambda)}{P(x^m_{jn}|X^m \setminus \{x^m_{jn}\}, \lambda)} \\ &\propto \begin{cases} P(x^m_{jn}|X^m_{k = k_{jt}}) \frac{\sum_m w^m N^m_{jt}}{\lambda + \sum_m w^m N^m_j - 1}, & (t = 1, \cdots, T_j), \\ P(x^m_{jn}|X^m_{k = k_{jt}}) \frac{\lambda}{\lambda + \sum_m w^m N^m_j - 1}, & (t = T_j + 1), \end{cases} \end{aligned}$$

where N^m_j is the number of the m-th modality's observations about the j-th object. The set of observations that correspond to the m-th modality and have the k-th topic in any object are represented by X^m_k.

In the Gibbs sampling procedure, a latent variable for each observation is drawn from the posterior probability distribution. If $t = T_j + 1$, a new observation is allocated to a new latent variable. The dish selection procedure is as follows. The prior probability that the k-th topic is allocated on the t-th latent variable becomes

$$P(k_{jt} = k|\gamma) = \begin{cases} \frac{M_k}{\gamma + M - 1}, & (k = 1, \cdots, K), \\ \frac{\gamma}{\gamma + M - 1}, & (k = K + 1), \end{cases}$$

where K is the number of topic types, and M_k is the number of latent variables on which the k-th topic is placed. Therefore, the posterior probability that the k-th topic is allocated on the t-th latent variable becomes

$$\begin{aligned} P(k_{jt} = k|X, \gamma) &= P(X_{jt}|X_k)P(k_{jt} = k|\gamma) \\ &= \begin{cases} P(X_{jt}|X_k) \frac{M_k}{\gamma + M - 1}, & (k = 1, \cdots, K), \\ P(X_{jt}|X_k) \frac{\gamma}{\gamma + M - 1}, & (k = K + 1) \end{cases} \end{aligned}$$

where $X = \cup_m X^m$, $X_k = \cup_m X^m_k$, and X_{jt} is the set of the j-th object's observations allocated to the t-th latent variable. A topic index for the latent variable t for the j-th object is drawn using the posterior probability, where γ is a hyperparameter. If $k = K + 1$, a new topic is placed on the latent variable.

By sampling t^m_{jn} and k_{jt}, the Gibbs sampler performs probabilistic object clustering:

$$t^m_{jn} \sim P(t^m_{jn}|X^{-mjn}, \lambda), \tag{1}$$

$$k_{jt} \sim P(k_{jt}|X^{-jt}, \gamma), \tag{2}$$

where $X^{-mjn} = X \setminus \{x^m_{jn}\}$, and $X^{-jt} = X \setminus X_{jt}$. By sampling t^m_{jn} for each observation in every object using (1) and sampling k_{jt} for each latent variable t in every object using (2), all of the latent variables in the MHDP can be inferred.

If t^m_{jn} and k_{jt} are given, the probability that the j-th object is included in the k-th category becomes

$$P(k|X_j) = \frac{\Sigma^{T_j}_{t = 1} \delta_k(k_{jt}) \sum_m w^m N^m_{jt}}{\sum_m w^m N^m_j}, \tag{3}$$

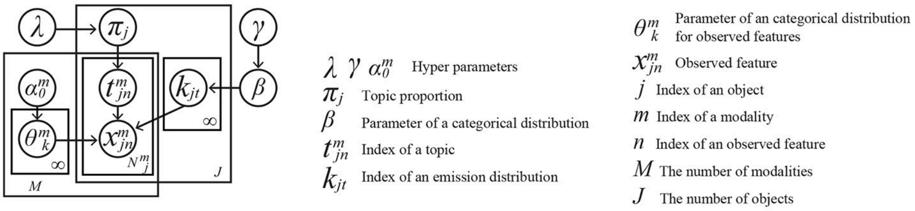

λ
π_j
γ
α_0^m
t_{jn}^m
k_{jt}
β
θ_k^m
x_{jn}^m
∞
∞
N_j^m
J
M
$\lambda\ \gamma\ \alpha_0^m$ Hyper parameters
π_j Topic proportion
β Parameter of a categorical distribution
t_{jn}^m Index of a topic
k_{jt} Index of an emission distribution
θ_k^m Parameter of an categorical distribution for observed features
x_{jn}^m Observed feature
j Index of an object
m Index of a modality
n Index of an observed feature
M The number of modalities
J The number of objects

FIGURE 2 | Graphical representation of an MHDP with *M* modalities corresponding to actions for perception.

where $X_j = \cup_m X_j^m$, w^m is the weight for the m-th modality and $\delta_a(x)$ is a delta function.

When a robot attempts to recognize a new object after the learning phase, the probability that feature x_{jn}^m is generated from the k-th topic becomes

$$P(x_{jn}^m|X_k^m) = \frac{w^m N_{kx_{jn}^m}^m + \alpha_0^m}{w^m N_k^m + d^m \alpha_0^m},$$

where d^m denotes the dimension of the m-th modality input, and $N_{kx_{jn}^m}^m$ represents the number of features x_{jn}^m that is corresponding to the index k. Topic k_t allocated to t for a new object is sampled from

$$k_t \sim P(k_{jt} = k|X, \gamma) \propto P(X_{jt}|X_k)\frac{\gamma}{\gamma + M - 1}.$$

These sampling procedures play an important role in the Monte Carlo approximation of our proposed method (see section 4.2.).

For a more detailed explanation of the MHDP, please refer to Nakamura et al. (2011b). Basically, a robot can autonomously learn object categories and recognize new objects using the multimodal categorization procedure described above. The performance and effectiveness of the method was evaluated in the paper.

4. ACTIVE PERCEPTION METHOD

4.1. Basic Formulation

A robot should have already conducted several actions and obtained information from several modalities when it attempts to select next action set for recognizing a target object. For example, visual information can usually be obtained by looking at the front face of the j-th object from a distance before interacting with the object physically. We assume that a robot has already obtained information corresponding to a subset of modalities $\mathbf{m_o}_j \subset \mathbf{M}$, where the subscript **o** means“originally” obtained modality information. When a robot faces a new object and has not obtained any information, $\mathbf{m_o}_j = \emptyset$.

The purpose of object recognition in multimodal categorization is different from conventional supervised learning-based pattern recognition problems. In supervised learning, the recognition result is evaluated by checking whether the output is the same as the truth label. However, in unsupervised learning, there are basically no truth labels. Therefore, the performance of active perception should be measured in a different manner.

The action set the robot selects is described as $\mathbf{A} = \{a_1, a_2, \ldots, a_{N_\mathbf{A}}\} \in \mathbf{2}^{\mathbf{M}\setminus\mathbf{m_o}_j}$, where $\mathbf{2}^{\mathbf{M}\setminus\mathbf{m_o}_j}$ is a family of subsets of $\mathbf{M} \setminus \mathbf{m_o}_j$, i.e., $\mathbf{A} \subset \mathbf{M} \setminus \mathbf{m_o}_j$, $a_i \in \mathbf{M} \setminus \mathbf{m_o}_j$ and N_A represents the number of available actions. We consider an effective action set for active perception to be one that largely reduces the distance between the final recognition state after the information from all modalities **M** is obtained and the recognition state after the robot executes the selected action set **A**. The recognition state is represented by the posterior distribution $P(\mathbf{z}_j|X_j^{\mathbf{m_o}_j\cup\mathbf{A}})$. Here, $\mathbf{z}_j = \{\{k_{jt}\}_{1\le t\le T_j}, \{t_{jn}^m\}_{m\in\mathbf{M},1\le n\le N_j^m}\}$ is a latent variable representing the j-th object’s topic information, where $X_j^\mathbf{A} = \cup_{m\in\mathbf{A}} X_j^m, X_j^m = \{x_{j1}^m, \ldots, x_{jn}^m, \ldots, x_{jN_j^m}^m\}$. Probability $P(\mathbf{z}_j|X_j^{\mathbf{m_o}_j\cup\mathbf{A}})$ represents the posterior distribution related to the object category after taking actions $\mathbf{m_o}_j$ and **A**.

The final recognition state, i.e., posterior distribution over latent variables after obtaining the information from all modalities **M**, becomes $P(\mathbf{z}_j|X_j^\mathbf{M})$. The purpose of active perception is to select a set of actions that can estimate the posterior distribution most accurately. When L actions can be executed, if we employ KL divergence as the metric of the difference between the two probability distributions,

$$\underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m_o}_j}}{\text{minimize}}\, \text{KL}\left(P(\mathbf{z}_j|X_j^\mathbf{M}), P(\mathbf{z}_j|X_j^{\mathbf{m_o}_j\cup\mathbf{A}})\right) \tag{4}$$

is a reasonable evaluation criterion for realizing effective active perception, where $\mathbf{F}_L^{\mathbf{m_o}_j} = \{\mathbf{A}|\mathbf{A} \subset \mathbf{M} \setminus \mathbf{m_o}_j, N_\mathbf{A} \le L\}$ is a feasible set of actions.

However, neither the true $X_j^\mathbf{M}$ nor $X_j^{\mathbf{m_o}_j\cup\mathbf{A}}$ can be observed before taking **A** on the j-th target object, and hence cannot be used at the moment of action selection. Therefore, a rational alternative for the evaluation criterion is the expected value of the KL divergence at the moment of action selection:

$$\underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m_o}_j}}{\text{minimize}}\, \mathbb{E}_{X_j^{\mathbf{M}\setminus\mathbf{m_o}_j}|X_j^{\mathbf{m_o}_j}}[\text{KL}\left(P(\mathbf{z}_j|X_j^\mathbf{M}), P(\mathbf{z}_j|X_j^{\mathbf{m_o}_j\cup\mathbf{A}})\right)]. \tag{5}$$

Here, we propose to use the IG maximization criterion to select the next action set for active perception:

$$\mathbf{A}_j^* = \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m_o}_j}}{\text{argmax}}\, \text{IG}(\mathbf{z}_j; X_j^\mathbf{A}|X_j^{\mathbf{m_o}_j}) \tag{6}$$

$$= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m_o}_j}}{\text{argmin}}\, \mathbb{E}_{X_j^\mathbf{A}|X_j^{\mathbf{m_o}_j}}[\text{KL}\left(P(\mathbf{z}_j|X_j^{\mathbf{m_o}_j\cup\mathbf{A}}), P(\mathbf{z}_j|X_j^{\mathbf{m_o}_j})\right)], \tag{7}$$

where $\mathrm{IG}(X; Y|Z)$ is the IG of Y for X, which is calculated on the basis of the probability distribution commonly conditioned by Z as follows:

$$\mathrm{IG}(X; Y|Z) = \mathrm{KL}\left(P(X, Y|Z), P(X|Z)P(Y|Z)\right).$$

By definition, the expected KL divergence is the same as $\mathrm{IG}(X; Y)$. The definition of IG and its relation to KL divergence are as follows.

$$\begin{aligned}\mathrm{IG}(X; Y) &= H(X) - H(X|Y)\\ &= \mathrm{KL}\left(P(X, Y), P(X)P(Y)\right)\\ &= \mathbb{E}_Y[\mathrm{KL}\left(P(X|Y), P(X)\right)].\end{aligned}$$

The optimality of the proposed criterion (6) is supported by Theorem 1.

Theorem 1. *The set of next actions* $\mathbf{A} \in \mathbf{F}_L^{\mathbf{m_o}_j}$ *that maximizes the* $\mathrm{IG}(\mathbf{z}_j; X_j^{\mathbf{A}}|X_j^{\mathbf{m_o}_j})$ *minimizes the expected KL divergence between the posterior distribution over* $\mathbf{z}_j$ *after all modality information has been observed and after* $\mathbf{A}$ *has been executed.*

$$\begin{aligned}&\underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m_o}_j}}{\mathrm{argmin}}\,\mathbb{E}_{X_j^{\mathbf{M}\setminus\mathbf{m_o}_j}|X_j^{\mathbf{m_o}_j}}[\mathrm{KL}\left(P(\mathbf{z}_j|X_j^{\mathbf{M}}), P(\mathbf{z}_j|X_j^{\mathbf{m_o}_j\cup\mathbf{A}})\right)]\\ &\quad = \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m_o}_j}}{\mathrm{argmax}}\,\mathrm{IG}(\mathbf{z}_j; X_j^{\mathbf{A}}|X_j^{\mathbf{m_o}_j})\end{aligned}$$

Proof: See Appendix A.

This theorem is essentially the result of well-known characteristics of IG (see MacKay, 2003; Russo and Van Roy, 2016 for example). This means that maximizing IG is the optimal policy for active perception in an MHDP-based multimodal object category recognition task. As a special case, when only a single action is permitted, the following corollary is satisfied.

Corollary 1.1. *The next action* $m \in \mathbf{M} \setminus \mathbf{m_o}_j$ *that maximizes* $\mathrm{IG}(\mathbf{z}_j; X_j^m|X_j^{\mathbf{m_o}_j})$ *minimizes the expected KL divergence between the posterior distribution over* $\mathbf{z}_j$ *after all modality information has been observed and after the action has been executed.*

$$\begin{aligned}&\underset{m\in\mathbf{M}\setminus\mathbf{m_o}_j}{\mathrm{argmin}}\ \mathbb{E}_{X_j^{\mathbf{M}\setminus\mathbf{m_o}_j}|X_j^{\mathbf{m_o}_j}}[\mathrm{KL}\left(P(\mathbf{z}_j|X_j^{\mathbf{M}}), P(\mathbf{z}_j|X_j^{\{m\}\cup\mathbf{m_o}_j})\right)]\\ &\quad = \underset{m\in\mathbf{M}\setminus\mathbf{m_o}_j}{\mathrm{argmax}}\ \mathrm{IG}(\mathbf{z}_j; X_j^m|X_j^{\mathbf{m_o}_j}). \qquad (8)\end{aligned}$$

Proof: By substituting $\{m\}$ into $\mathbf{A}$ in Theorem 1, we can obtain the corollary.

Using IG, the active perception strategy for the next single action is simply described as follows:

$$m_j^* = \underset{m\in\mathbf{M}\setminus\mathbf{m_o}_j}{\mathrm{argmax}}\ \mathrm{IG}(\mathbf{z}_j; X_j^m|X_j^{\mathbf{m_o}_j}). \qquad (9)$$

This means that the robot should select the action m_j^* that can obtain the $X_j^{m_j^*}$ that maximizes the IG for the recognition result $\mathbf{z}_j$ under the condition that the robot has already observed $X_j^{\mathbf{m_o}_j}$.

However, we still have two problems, as follows.

1. The argmax operation in (6) is a combinatorial optimization problem and incurs heavy computational cost when $\#(\mathbf{M}\setminus\mathbf{m_o}_j)$ and L become large.
2. The calculation of $\mathrm{IG}(\mathbf{z}_j; X_j^{\mathbf{A}}|X_j^{\mathbf{m_o}_j})$ cannot be performed in a straightforward manner.

Based on some properties of the MHDP, we can obtain reasonable solutions for these two problems.

4.2. Sequential Decision Making as a Submodular Maximization

If a robot wants to select L actions $\mathbf{A}_j = \{a_1, a_2, \ldots, a_L\}$ ($a_i \in \mathbf{M} \setminus \mathbf{m_o}_j$), it has to solve (6), i.e., a combinatorial optimization problem. The number of combinations of L actions is ${}_{\#(\mathbf{M}\setminus\mathbf{m_o}_j)}C_L$, which increases dramatically when the number of possible actions $\#(\mathbf{M} \setminus \mathbf{m_o}_j)$ and L increase. For example, Sinapov et al. (2014) gave a robot 10 different behaviors in their experiment on robotic multimodal categorization. Future autonomous robots will have more available actions for interacting with a target object and be able to obtain additional types of modality information through these interactions. Hence, it is important to develop an efficient solution for the combinatorial optimization problem.

Here, the MHDP has advantages for solving this problem.

Theorem 2. *The evaluation criterion for multimodal active perception* $\mathrm{IG}(\mathbf{z}_j; X_j^{\mathbf{A}}|X_j^{\mathbf{m_o}_j})$ *is a submodular and non-decreasing function with regard to* $\mathbf{A}$.

Proof: As shown in the graphical model of the MHDP in **Figure 2**, the observations for each modality X_j^m are conditionally independent under the condition that a set of latent variables $\mathbf{z}_j = \{\{k_{jt}\}_{1\le t\le T_j}, \{t_{jn}^m\}_{m\in\mathbf{M}, 1\le n\le N_j^m}\}$is given. This satisfies the conditions of the theorem by Krause and Guestrin (2005). Therefore, $\mathrm{IG}(\mathbf{z}_j; X_j^m|X_j^{\mathbf{m_o}_j})$ is a submodular and non-decreasing function with regard to X_j^m.

Submodularity is a property similar to the convexity of a real-valued function in a vector space. If a set function $F: V \to R$ satisfies

$$F(A \cup x) - F(A) \ge F(A' \cup x) - F(A'),$$

where V is a finite set $\forall A \subset A' \subseteq V$ and $x \notin A$, the set function F has submodularity and is called a submodular function.

Function IG is not always a submodular function. However, Krause et al. proved that $\mathrm{IG}(U; A)$ is submodular and non-decreasing with regard to $A \subseteq S$ if all of the elements of S are conditionally independent under the condition that U is given. With this theorem, Krause and Guestrin (2005) solved the sensor allocation problem efficiently. Theorem 2 means that the problem (6) is reduced to a *submodular maximization problem*.

It is known that the greedy algorithm is an efficient strategy for the submodular maximization problem. Nemhauser et al. (1978) proved that the greedy algorithm can select a subset that is at most a constant factor $(1 - 1/e)$ worse than the optimal set, if the evaluation function $F(A)$ is submodular, non-decreasing, and $F(\emptyset) = 0$, where $F(\cdot)$ is a set function, and A

is a set. If the evaluation function is a submodular set function, a greedy algorithm is practically sufficient for selecting subsets in many cases. In sum, a greedy algorithm gives a near-optimal solution. However, the greedy algorithm is still inefficient because it requires an evaluation of all choices at each step of a sequential decision making process.

Minoux (1978) proposed lazy greedy algorithm to make the greedy algorithm more efficient for the submodular evaluation function. The lazy greedy algorithm can reduce the number of evaluations by using the characteristics of a submodular function.

4.3. Monte Carlo Approximation of IG

Equations (6) and (9) provide a robot with an appropriate criterion for selecting an action to efficiently recognize a target object. However, at first glance, it looks difficult to calculate the IG. First, the calculation of the expectation procedure $\mathbb{E}_{X_j^{\mathbf{A}}|X_j^{\mathbf{m}_{\mathbf{o}j}}}[\cdot]$ requires a sum operation over all possible $X_j^{\mathbf{A}}$. The number of possible $X_j^{\mathbf{A}}$ exponentially increases when the number of elements in the BoF increases. Second, the calculation of $P(\mathbf{z}_j|X_j^{\mathbf{A}\cup\mathbf{m}_{\mathbf{o}j}})$ for each possible observation $X_j^{\mathbf{A}}$ requires the same computational cost as recognition in the multimodal categorization itself. Therefore, the straightforward calculation for solving (9) is computationally impossible in a practical sense.

However, by exploiting a characteristic property of the MHDP, a Monte Carlo approximation can be derived. First, we describe IG as the expectation of a logarithm term.

$$\begin{aligned}
\mathrm{IG}(\mathbf{z}_j; X_j^m|X_j^{\mathbf{m}_{\mathbf{o}j}}) &= \sum_{\mathbf{z}_j, X_j^m} P(\mathbf{z}_j, X_j^m|X_j^{\mathbf{m}_{\mathbf{o}j}}) \log \frac{P(\mathbf{z}_j, X_j^m|X_j^{\mathbf{m}_{\mathbf{o}j}})}{P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}})P(X_j^m|X_j^{\mathbf{m}_{\mathbf{o}j}})} \\
&= \mathbb{E}_{\mathbf{z}_j, X_j^m|X_j^{\mathbf{m}_{\mathbf{o}j}}}\Big[\log \frac{P(\mathbf{z}_j, X_j^m|X_j^{\mathbf{m}_{\mathbf{o}j}})}{P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}})P(X_j^m|X_j^{\mathbf{m}_{\mathbf{o}j}})}\Big]. \quad (10)
\end{aligned}$$

An analytic evaluation of (10) is also practically impossible. Therefore, we adopt a Monte Carlo method. Equation (10) suggests that an efficient Monte Carlo approximation can be performed as shown below if we can sample

$$(z_j^{[k]}, X_j^{m[k]}) \sim P(\mathbf{z}_j, X_j^m|X_j^{\mathbf{m}_{\mathbf{o}j}}), \quad (k \in \{1, \ldots, K\}).$$

Fortunately, the MHDP provides a sampling procedure for $z_j^{[k]} \sim P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}})$ and $X_j^{m[k]} \sim P(X_j^m|z_j^{[k]})$ in its original paper (Nakamura et al., 2011b). In the context of multimodal categorization by a robot, $X_j^{m[k]} \sim P(X_j^m|z_j^{[k]})$ is a prediction of an unobserved modality's sensation using observed modalities' sensations, i.e., cross-modal inference. The sampling process of $(z_j^{[k]}, X_j^{m[k]})$ can be regarded as a mental simulation by a robot that predicts the unobserved modality's sensation leading to a categorization result based on the predicted sensation and observed information.

$$\begin{aligned}
(10) &\approx \frac{1}{K}\sum_k \log \frac{P(\mathbf{z}_j^{[k]}, X_j^{m[k]}|X_j^{\mathbf{m}_{\mathbf{o}j}})}{P(\mathbf{z}_j^{[k]}|X_j^{\mathbf{m}_{\mathbf{o}j}})P(X_j^{m[k]}|X_j^{\mathbf{m}_{\mathbf{o}j}})} \\
&= \frac{1}{K}\sum_k \log \frac{P(X_j^{m[k]}|\mathbf{z}_j^{[k]}, X_j^{\mathbf{m}_{\mathbf{o}j}})}{\underbrace{P(X_j^{m[k]}|X_j^{\mathbf{m}_{\mathbf{o}j}})}_{*}}. \quad (11)
\end{aligned}$$

In (11), $P(X_j^{m[k]}|\mathbf{z}_j^{[k]}, X_j^{\mathbf{m}_{\mathbf{o}j}})$ in the numerator can be easily calculated because all the parent nodes of $X_j^{m[k]}$ are given in the graphical model shown in **Figure 2**. However, $P(X_j^{m[k]}|X_j^{\mathbf{m}_{\mathbf{o}j}})$ in the denominator cannot be evaluated in a straightforward way. Again, a Monte Carlo method can be adopted, as follows:

$$\begin{aligned}
(*) = P(X_j^{m[k]}|X_j^{\mathbf{m}_{\mathbf{o}j}}) &= \sum_{\mathbf{z}_j} P(X_j^{m[k]}|\mathbf{z}_j, X_j^{\mathbf{m}_{\mathbf{o}j}})P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}) \\
&= \mathbb{E}_{\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}}[P(X_j^{m[k]}|\mathbf{z}_j, X_j^{\mathbf{m}_{\mathbf{o}j}})] \\
&\approx \frac{1}{K'}\sum_{k'} P(X_j^{m[k]}|\mathbf{z}_j^{[k']}, X_j^{\mathbf{m}_{\mathbf{o}j}}) \quad (12)
\end{aligned}$$

where K' is the number of samples for the second Monte Carlo approximation. Fortunately, in this Monte Carlo approximation (12), we can reuse the samples drawn in the previous Monte Carlo approximation efficiently, i.e., $K' = K$. By substituting (12) for (11), we finally obtain the approximate IG for the criterion of active perception, i.e., our proposed method, as follows:

$$\mathrm{IG}(\mathbf{z}_j; X_j^m|X_j^{\mathbf{m}_{\mathbf{o}j}}) \approx \frac{1}{K}\sum_k \log \frac{P(X_j^{m[k]}|\mathbf{z}_j^{[k]}, X_j^{\mathbf{m}_{\mathbf{o}j}})}{\frac{1}{K}\sum_{k'} P(X_j^{m[k]}|\mathbf{z}_j^{[k']}, X_j^{\mathbf{m}_{\mathbf{o}j}})}.$$

Note that the computational cost for evaluating IG becomes $O(K^2)$. In summary, a robot can approximately estimate the IG for unobserved modality information by generating virtual observations based on observed data and evaluating their likelihood.

4.4. MHDP-Based Active Perception Methods

We propose the use of the *greedy* and *lazy greedy algorithms* for selecting L actions to recognize a target object on the basis of the submodular property of IG. The final greedy and lazy greedy algorithms for MHDP-based active perception, i.e., our proposed methods, are shown in Algorithms 1 and 2, respectively.

The main contribution of the lazy greedy algorithm is to reduce the computational cost of active perception. The majority of the computational cost originates from the number of times a robot evaluates IG_m for determining action sequences. When a robot has to choose L actions, the brute-force algorithm that directly evaluates all alternatives $\mathbf{A} \in \mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}$ using (6) requires ${}_{\#(\mathbf{M}\setminus\mathbf{m}_{\mathbf{o}j})}C_L$ evaluations of $\mathrm{IG}(\mathbf{z}_j; X_j^{\mathbf{A}}|X_j^{\mathbf{m}_{\mathbf{o}j}})$. In contrast, the greedy algorithm requires $\{\#(\mathbf{M} \setminus \mathbf{m}_{\mathbf{o}j}) + (\#(\mathbf{M} \setminus \mathbf{m}_{\mathbf{o}j}) - 1) + \ldots +$

Algorithm 1 Greedy algorithm.

```
Require: MHDP is trained using a training data set.
  The $j$-th object is found.
  $\mathbf{m}_{\mathbf{o}j}$ is initialized, and $X_j^{\mathbf{m}_{\mathbf{o}j}}$ is observed.
  for $l = 1$ to $L$ do
    for all $m \in \mathbf{M} \setminus \mathbf{m}_{\mathbf{o}j}$ do
      for $k = 1$ to $K$ do
        Draw
          $(z_j^{[k]}, X_j^{m[k]}) \sim P(\mathbf{z}_j, X_j^m | X_j^{\mathbf{m}_{\mathbf{o}j}})$
      end for
      $\mathrm{IG}_m \leftarrow \frac{1}{K}\sum_k \log \frac{P(X_j^{m[k]}|\mathbf{z}_j^{[k]}, X_j^{\mathbf{m}_{\mathbf{o}j}})}{\frac{1}{K}\sum_{k'} P(X_j^{m[k]}|\mathbf{z}_j^{[k']}, X_j^{\mathbf{m}_{\mathbf{o}j}})}$
    end for
    $m^* \leftarrow \operatorname*{argmax}_m \mathrm{IG}_m$
    Execute the $m^*$-th action to the $j$-th target object and obtain $X_j^{m^*}$.
    $\mathbf{m}_{\mathbf{o}j} \leftarrow \mathbf{m}_{\mathbf{o}j} \cup \{m^*\}$
  end for
```

Algorithm 2 Lazy greedy algorithm.

```
Require: The MHDP is trained using a training data set.
  The $j$-th object is found.
  $\mathbf{m}_{\mathbf{o}j}$ is initialized, and $X_j^{\mathbf{m}_{\mathbf{o}j}}$ is observed.
  for all $m \in \mathbf{M} \setminus \mathbf{m}_{\mathbf{o}j}$ do
    for $k = 1$ to $K$ do
      Draw
        $(z_j^{[k]}, X_j^{m[k]}) \sim P(\mathbf{z}_j, X_j^m | X_j^{\mathbf{m}_{\mathbf{o}j}})$
    end for
    $\mathrm{IG}_m \leftarrow \frac{1}{K}\sum_k \log \frac{P(X_j^{m[k]}|\mathbf{z}_j^{[k]}, X_j^{\mathbf{m}_{\mathbf{o}j}})}{\frac{1}{K}\sum_{k'} P(X_j^{m[k]}|\mathbf{z}_j^{[k']}, X_j^{\mathbf{m}_{\mathbf{o}j}})}$
  end for
  $m^* \leftarrow \operatorname*{argmax}_m \mathrm{IG}_m$
  Execute the $m^*$-th action to the $j$-th target object and obtain $X_j^{m^*}$.
  $\mathbf{m}_{\mathbf{o}j} \leftarrow \mathbf{m}_{\mathbf{o}j} \cup \{m^*\}$
  Prepare a stack $S$ for the modality indices and initialize it.
  for all $m \in \mathbf{M} \setminus \mathbf{m}_{\mathbf{o}j}$ do
    $push(S, (m, \mathrm{IG}_m))$
  end for
  for $l = 1$ to $L - 1$ do
    repeat
      $S \leftarrow descending\_sort(S)$ // w.r.t. $\mathrm{IG}_m$
      $(m^1, \mathrm{IG}_{m^1}) \leftarrow pop(S)$ , $(m^2, \mathrm{IG}_{m^2}) \leftarrow pop(S)$
      // Re-evaluate $\mathrm{IG}_{m^1}$ as follows.
      for $k = 1$ to $K$ do
        Draw
          $(z_j^{[k]}, X_j^{m^1[k]}) \sim P(\mathbf{z}_j, X_j^{m^1} | X_j^{\mathbf{m}_{\mathbf{o}j}})$
      end for
      $\mathrm{IG}_{m^1} \leftarrow \frac{1}{K}\sum_k \log \frac{P(X_j^{m^1[k]}|\mathbf{z}_j^{[k]}, X_j^{\mathbf{m}_{\mathbf{o}j}})}{\frac{1}{K}\sum_{k'} P(X_j^{m^1[k]}|\mathbf{z}_j^{[k']}, X_j^{\mathbf{m}_{\mathbf{o}j}})}$
      $push(S, (m^2, \mathrm{IG}_{m^2}))$, $push(S, (m^1, \mathrm{IG}_{m^1}))$
    until $\mathrm{IG}_{m^1} \geq \mathrm{IG}_{m^2}$
    $m^* \leftarrow m^1$
    $pop(S)$
    Execute the $m^*$-th action to the $j$-th target object and obtain $X_j^{m^*}$.
    $\mathbf{m}_{\mathbf{o}j} \leftarrow \mathbf{m}_{\mathbf{o}j} \cup \{m^*\}$
  end for
```

$(\#(\mathbf{M} \setminus \mathbf{m}_{\mathbf{o}j}) - L + 1)\}$ evaluations of $\mathrm{IG}(\mathbf{z}_j; X_j^m | X_j^{\mathbf{m}_{\mathbf{o}j}})$, i.e., $O(ML)$. The lazy greedy algorithm incurs the same computational cost as the greedy algorithm only in the worst case. However, practically, the number of re-evaluations in the lazy greedy algorithm is quite small. Therefore, the computational cost of the lazy greedy algorithm increases almost in proportion to L, i.e., almost linearly. The memory requirement of the proposed method is also quite small. Both the greedy and lazy greedy algorithms only require memory for IG_m for each modality and K samples for the Monte Carlo approximation. These requirements are negligibly small compared with the MHDP itself.

Note that the IG_m is not the exact IG, but an approximation. Therefore, the differences between IG and IG_m may harm the performance of greedy and lazy greedy algorithms to a certain extent. However, the algorithms are expected to work practically. We evaluated the algorithms through experiments.

5. EXPERIMENT 1: HUMANOID ROBOT

5.1. Conditions

An experiment using an upper-torso humanoid robot was conducted to verify the proposed active perception method in the real-world environment. In this experiment, RIC-Torso, developed by the RT Corporation, was used (see **Figure 3**). RIC-Torso is an upper-torso humanoid robot that has two robot hands. We prepared an experimental environment that is similar to the one in the original MHDP paper (Nakamura et al., 2011b). The robot has four available actions and four corresponding modality information. The set of modalities was $\mathbf{M} = \{m^v, m^{as}, m^{ah}, m^h\}$, which represent visual information, auditory information obtained by shaking an object, one by hitting an object and haptic information, respectively.

5.1.1. Visual Information (m^v)

Visual information was obtained from the Xtion PRO LIVE set on the head of the robot. The camera was regarded as the eyes of the robot. The robot captured 74 images of a target object while it rotated on a turntable (see **Figure 3**). The size of each image was re-sized to 320 × 240. Scale-invariant feature transform (SIFT) feature vectors were extracted from each captured image (Lowe, 2004). A certain number of 128-dimensional feature vectors were obtained from each image. Note that the SIFT feature did not consider hue information. All of the obtained feature vectors were transformed into BoF representations using k-means clustering with $k = 25$. The number of clusters k was determined empirically, considering prior works (Nakamura et al., 2011b; Araki et al., 2012). The k-means clustering was performed using data from all objects in a training set, and the centroids of the clusters were determined. BoF representations were used as observation data for the visual modality of the MHDP. The index for this modality was defined as m^v.

5.1.2. Auditory Information (m^{as} and m^{ah})

Auditory information was obtained from a multipowered shotgun microphone NTG-2 by RODE Microphone. The microphone was regarded as the ear of the robot. In this experiment, two types of auditory information were acquired. One was generated by hitting the object, and the other was generated by shaking it. The two sounds were regarded as different auditory information and hence different modality observations in the MHDP model. The two actions, i.e., hitting and shaking, were manually programmed for the robot. Each action was implemented as a fixed trajectory. When the robot began to execute an action, it also started recording the objects's sound (see **Figure 3**). The sound was recorded until two seconds after the robot finished the action. The recorded auditory data were temporally divided into frames, and each frame was transformed into 13-dimensional Mel-frequency cepstral coefficients (MFCCs). The MFCC feature vectors were transformed into BoF representations using k-means clustering with $k = 25$ in the same way as the visual information. The indices of these modalities were defined as m^{as} and m^{ah}, respectively, for "shake" and "hit."

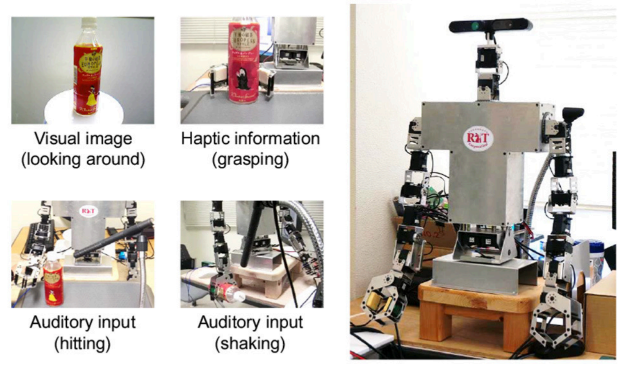


FIGURE 3 | A humanoid robot used in the experiment.

5.1.3. Haptic Information (m^h)

Haptic information was obtained by grasping a target object using the robot's hand. When the robot attempted to obtain haptic information from an object placed in front of it, it moved its hand to the object and gradually closed its hand until a certain amount of counterforce was detected (see **Figure 3**). The joint angle of the hand was measured when the hand touched the target object and when the hand stopped. The two variables and difference between the two angles were used as a three-dimensional feature vector. When obtaining haptic information, the robot grasped the target object 10 times and obtained 10 feature vectors. The feature vectors were transformed into BoF representations using k-means clustering with $k = 5$ in the same way as for the other information types. The index of the haptic modality was defined as m^h.

5.1.4. Multimodal Information as BoF Representations

In summary, a robot could obtain multimodal information from four modalities for perception. The dimensions of the BoFs were set to 25, 25, 25, and 5 for m^v, m^{as}, m^{ah}, and m^h, respectively. The dimension of each BoF corresponds to the number of clusters for k-means clustering. The numbers of clusters, i.e., the sizes of the dictionaries, were empirically determined on the basis of a preliminary experiment on multimodal categorization. All of the training datasets were used to train the dictionaries. The histograms of the feature vectors, i.e., the BoFs, were resampled to make their counts $N_j^{m^v} = 100, N_j^{m^{as}} = 80, N_j^{m^{ah}} = 130$, and $N_j^{m^h} = 30$. The weight of each modality w^m was set to 1. The formation of multimodal object categories itself is out of the scope of this paper. Therefore, the constants were empirically determined so that the robot could form object categories that are similar to human participants. The number of samples K in the Monte Carlo approximation for estimating IG was set to $K = 5,000$. The constant K was determined empirically. The effect of K will be examined in the experiment as well (see **Figure 11**).

5.1.5. Target Objects

For the target objects, 17 types of commodities were prepared for the experiment shown in **Figure 4**. An object was provided for obtaining a training data, i.e., data for object categorization, and another object was provided for obtaining test data, i.e., data for active perception, for each type of objects. Each index on the right-hand side of the figure indicates the index of each object. The hardness of the balls, the striking sounds of the cups, and the sounds made while shaking the bottles were different depending on the object categories. Therefore, ground-truth categorization could not be achieved using visual information alone.

5.2. Procedure

The experimental procedure was as follows. First, the robot formed object categories through multimodal categorization in an unsupervised manner. An experimenter placed each object

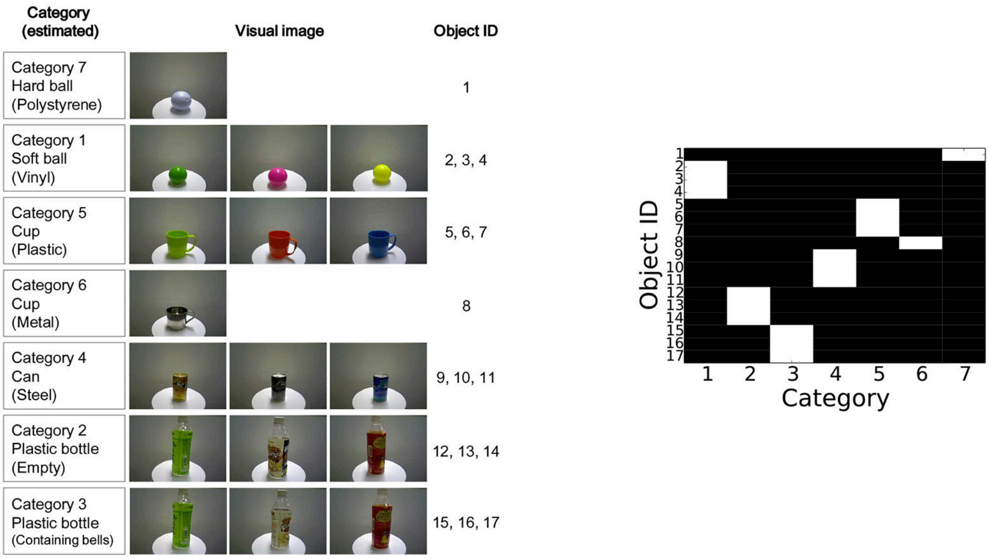


FIGURE 4 | (Left) target objects used in the experiment and **(Right)** categorization results obtained in the experiment.

in front of the robot one by one. In this training phase, two objects for each type of objects were provided. The robot looked at the object to obtain visual features, grasped it to obtain haptic features, shook it to obtain auditory shaking features, and hit it to obtain the auditory striking features. After obtaining the multimodal information of the objects as a training data set, the MHDP was trained using a Gibbs sampler. The results of multimodal categorization are shown in **Figure 4**. The category that has the highest posterior probability for each object is shown in white. These results show that the robot can form multimodal object categories using MHDP, as described in Nakamura et al. (2011b). After the robot had formed object categories, we fixed the latent variables for the training data set[3].

Second, an experimental procedure for active perception was conducted. An experimenter placed an object in front of the robot. The robot observed the object using its camera, obtained visual information, and set $\mathbf{m}_{\mathbf{o}j} = \{m^v\}$. An object was provided for each type of objects shown in **Figure 4** to the robot one by one. Therefore, 17 objects were used for evaluating each active perception strategy. The sequential action selection and object recognition were performed once per an object. At each step of the sequential action selection, Gibbs sampler for MHDP was performed and it updated its latent variables, i.e., recognition state, of the MHDP. The robot then determined its next set of actions for recognizing the target object using its active perception strategy shown in Algorithms 1 and 2.

5.3. Results

5.3.1. Selecting the Next Action

First, we describe results for the first single action selection after obtaining visual information. In this experiment, the robot had three choices for its next action, i.e., m^{as}, m^{ah}, and m^h. To evaluate the results of active perception, we used $\mathrm{KL}\left(P(k|X_j^{\mathbf{M}}), P(k|X_j^{\mathbf{A}\cup\mathbf{m}_{\mathbf{o}j}})\right)$, i.e., the distance between the posterior distribution over the object categories k in the final recognition state and that in the next recognition state as an evaluation criterion on behalf of $\mathrm{KL}\left(P(\mathbf{z}_j|X_j^{\mathbf{M}}), P(\mathbf{z}_j|X_j^{\mathbf{A}\cup\mathbf{m}_{\mathbf{o}j}})\right)$, which is the original evaluation criterion in (4). The computational cost for numerical evaluation of $\mathrm{KL}\left(P(\mathbf{z}_j|X_j^{\mathbf{M}}), P(\mathbf{z}_j|X_j^{\mathbf{A}\cup\mathbf{m}_{\mathbf{o}j}})\right)$ using a Monte Carlo method is too high because $\mathbf{z}_j = \{\{k_{jt}\}_{1\le t\le T_j}, \{t_{jn}^m\}_{m\in\mathbf{M},1\le n\le N_j^m}\}$ has so many variables and a posterior distributions over $\mathbf{z}_j$ is very complex.

Figure 5 (Top) shows samples of the KL divergence between the posterior probabilities of the category after obtaining the information from all modalities and after obtaining only visual information.

With regard to some objects, e.g., objects 6 and 7, the figure shows samples of that visual information seems to be sufficient for the robot to recognize the objects as compared the other objects[4]. However, with regard to many objects, visual information alone could not lead the recognition state to the final state. However, it could be reached using the information of all modalities. **Figure 5** (Middle) shows samples of IG_m calculated using the visual information for each action. **Figure 5** (Bottom) shows the KL divergence between the final recognition state and the posterior probability estimated after obtaining visual information and the information of each selected action. We observe that an action with a higher value of IG_m tended to further reduce the KL divergence, as Theorem 1

[3] The collected datasets for this experiment can be found in GitHub: https://github.com/tanichu/data-active-perception-hmdp

[4] Note that currently we don't have a good criteria of KL divergence to determine whether performing further actions are necessary or not.

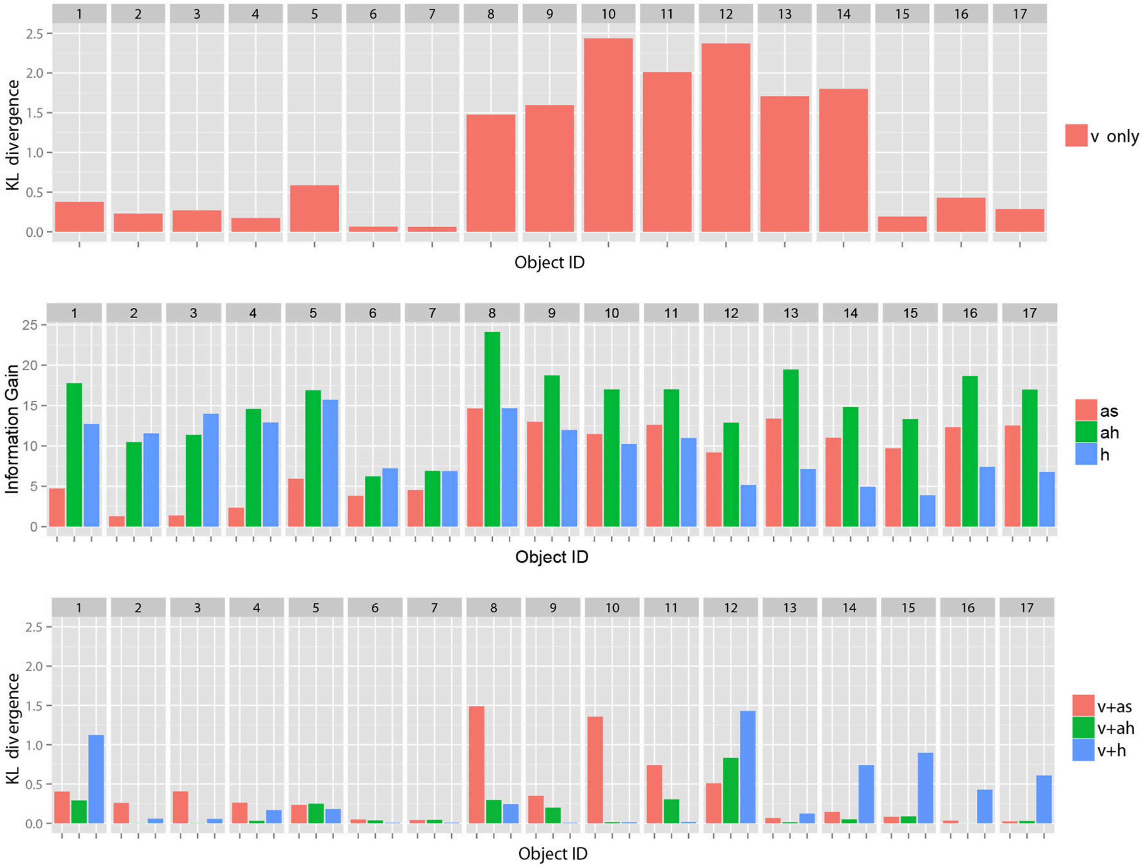


FIGURE 5 | (Top) Samples of KL divergence between the final recognition state and the posterior probability estimated after obtaining only visual information, **(Middle)** samples of estimated IG_m for each object based on visual information (v), and **(Bottom)** samples of KL divergence between the final recognition state and the posterior probability estimated after obtaining only visual information and each selected action where as, ah, h represent represent auditory information obtained by shaking an object, one by hitting an object and haptic information, respectively. Our theory of multimodal active perception suggests that the action with the highest information gain (shown in the middle) tends to lead its initial recognition state (whose KL divergence from the final recognition state is shown at the top) to a recognition state whose KL divergence from the final recognition state (shown at the bottom) is the smallest. These figures suggest the probabilistic relationships were satisfied as a whole.

suggests. **Figure 6** shows the average KL divergence for the final recognition state after executing an action selected by the IG_m criterion. Actions IG.min, IG.mid, and IG.max denote actions that have the minimum, middle, and maximum values of IG_m, respectively. These results show that IG.max clearly reduced the uncertainty of the target objects.

The precision of category recognition after an action execution is summarized in **Table 1**. Basically, a category recognition result is obtained as the posterior distribution (3) in the MHDP. The category with the highest posterior probability is considered to be the recognition result for illustrative purposes in **Table 1**. Obtaining information by executing IG.max almost always increased recognition performance.

Examples of changes in the posterior distribution are shown in **Figure 7** (Left, Right) for objects 8 ("metal cup") and 12 ("empty plastic bottle"), respectively. The robot could not clearly recognize the category of object 8 after obtaining visual information. Action IG_m in **Figure 5** shows that m^{ah} was IG.max for the 8th object. **Figure 7** (Left) shows that m^{ah} reduced the uncertainty and allowed the robot to correctly recognize the object, as evidenced by category 6, a metal cup. This means that the robot noticed that the target object was a metal cup by hitting it and listening to its metallic sound. The metal cup did not make a sound when the robot shook it. Therefore, the IG for m^{as} was small. As **Figure 7** (Right) shows, the robot first recognized the 12th object as a plastic bottle containing bells with high probability and as an empty plastic bottle with a low probability. **Figure 5** shows that the IG_m criterion suggested m^{ah} as the first alternative and m^{as} as the second alternative. **Figure 7** (Right) shows that m^{as} and m^{ah} could determine that the target object was an empty plastic bottle, but m^{h} could not.

As humans, we would expect to differentiate an empty bottle from a bottle containing bells by shaking or hitting the bottle, and differentiate a metal cup from a plastic cup by hitting it. The proposed active perception method constructively reproduced this behavior in a robotic system

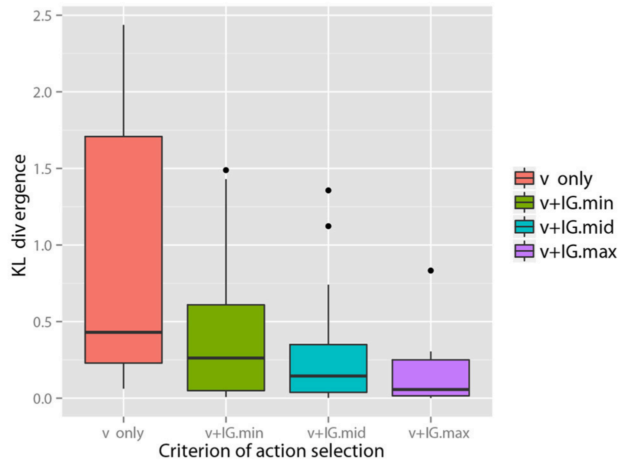


FIGURE 6 | Reduction in the KL divergence by executing an action selected on the basis of the IG_m maximization criterion. The KL divergences between the recognition state after executing the second action and the final recognition state are calculated for all objects and shown with box plot. This shows that an action with more information brings the recognition of its state closer to the final recognition state.

TABLE 1 | Number of successfully recognized objects.

v only	v+IG.min	v+IG.mid	v+IG.max	Full information
8/17	11/17	15/17	16/17	17/17

using an unsupervised multimodal machine learning approach.

5.3.2. Selecting the Next Set of Multiple Actions

We evaluated the greedy and lazy greedy algorithms for active perception sequential decision making. The KL divergence from the final state for all target objects is averaged at each step and shown in **Figure 8**. For each condition, the KL divergence gradually decreased and reached almost zero. However, the rate of decrease notably differed. As the theory of submodular optimization suggests, the greedy algorithm was shown to be a better solution on average and slightly worse than the best case (Nemhauser et al., 1978). The best and worst cases were selected after all types of sequential actions had been performed. The “average” is the average of the KL divergence obtained by all possible types of sequential actions. The results for the lazy greedy algorithm were almost same as those of the greedy algorithm, as Minoux (1978) suggested.

The sequential behaviors of IG_m were observed to determine if their behaviors were consistent with our theories. For example, the changes in IG_m at each step as the robot sequentially selected its action to perform on object 10 using the greedy algorithm is shown in **Figure 9**. Theorem 2 shows that the IG is a submodular function. This predicts that IG_m decreases monotonically when a new action is executed in active perception. When the robot obtained only visual information (v only in **Figure 9**), all values of IG_m were still large. After m^{ah} was executed on the basis of the greedy algorithm, $\mathrm{IG}_{m^{ah}}$ became zero. At the same time, $\mathrm{IG}_{m^{as}}$ and IG_{m^h} decreased. In the same way, all values of IG_m gradually decreased monotonically.

Figure 10 shows the time series of the posterior probability of the category for object 10 during sequential active perception. Using only visual information, the robot misclassified the target object as a plastic bottle containing bells (category 3). The action sequence in reverse order did not allow the robot to recognize the object as a steel can at its first step and change its recognition state to an empty plastic bottle (category 4). After the second action, i.e., grasping (m^h), the robot recognized the object as a steel can. In contrast, the greedy algorithm could determine that the target object was in category 4, i.e., steel can, with its first action.

The effect of the number of samples K for the Monte Carlo approximation was observed. **Figure 11** shows the relation between K and the standard deviation of the estimated IG_m for the 15th object for each action after obtaining a visual image. This figure shows that estimation error gradually decreases when K increases. Roughly speaking, $K \geq 1,000$ seems to be required for an appropriate estimate of IG_m in our experimental setting. Evaluation of IG_m required less than 1 second, which is far shorter than the time required for action execution by a robot. This means that our method can be used in a real-time manner.

These empirical results show that the proposed method for active perception allowed a robot to select appropriate actions sequentially to recognize an object in the real-world environment and in a real-time manner. It was shown that the theoretical results were supported, even in the real-world environment.

6. EXPERIMENT 2: SYNTHETIC DATA

In experiment 1, the numbers of classes, actions, and modalities as well as the size of dataset were limited. In addition, it was difficult to control the robotic experimental settings so as to check some interesting theoretical properties of our proposed method. Therefore, we performed a supplemental experiment, Experiment 2, using synthetic data comprising 21 object types, 63 objects, and 20 actions, i.e., modalities.

First, we checked the validity of our active perception method when the number of types of actions increases. Second, we checked how the method worked when two classes were assigned to the same object. Although the MHDP can categorize an object into two or more categories in a probabilistic manner, each object was classified into a single category in the previous experiment.

6.1. Conditions

A synthetic dataset was generated using the generative model that the MHDP assumes (see **Figure 2**). We prepared 21 virtual object classes, and three objects were generated from each object class, i.e., we obtained 63 objects in total. Among the object classes, 14 object classes are “pure,” and seven object classes are “mixed.” For each pure object class, a multinomial distribution was drawn from the Dirichlet distribution corresponding to each modality. We set the number of modalities $M = 20$. The hyperparameters of the Dirichlet distributions of the modalities were set to $\alpha_0^m =$

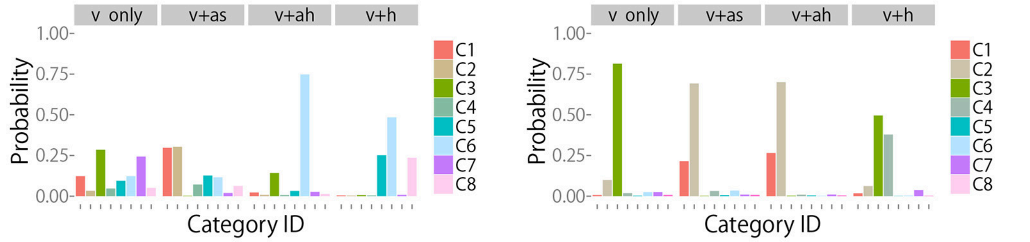


FIGURE 7 | (Left) Posterior probability of the category for object 8 after executing each action. These results show that the action with the highest information gain, i.e., *ah*, allowed the robot to efficiently estimate that the true object category was "metal cup". **(Right)** Posterior probability of the category for object 12 after executing each action. These results show that the actions with the highest and second highest information gain, i.e., *ah* and *as*, allowed the robot to efficiently estimate that the true object category was "empty plastic bottle".

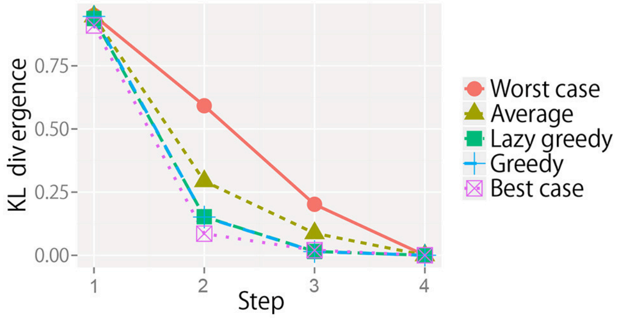


FIGURE 8 | KL divergence from the final state at each step for each sequential action selection procedure. Note that the line of the lazy greedy algorithm is overlapped by that of the greedy algorithm.

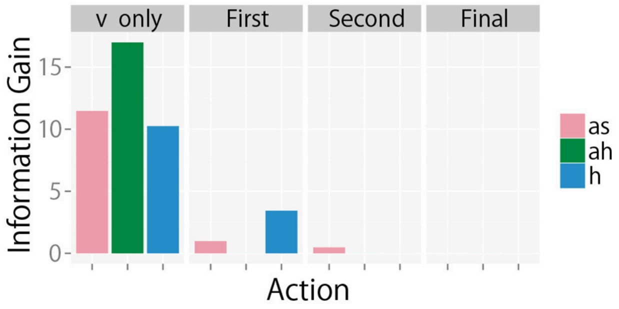


FIGURE 9 | IG_m at each step for object 10 when the greedy algorithm is used.

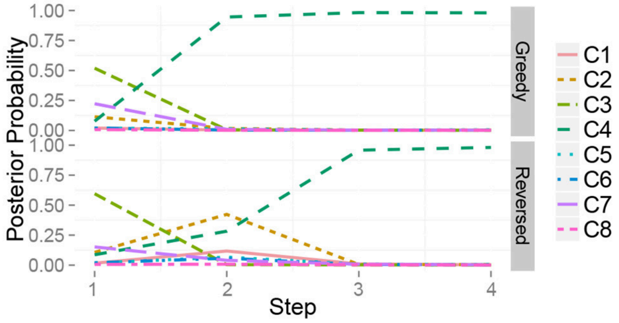


FIGURE 10 | Time series of the posterior probability of the category for object 10 during sequential action selection based on **(top)** the greedy algorithm, i.e., $m^{ah} \to m^{h} \to m^{as}$, and **(bottom)** its reverse order , i.e., $m^{as} \to m^{h} \to m^{ah}$.

$0.4(m-1)$ for $m > 1$. For $m = 1$, we set $\alpha_0^1 = 10$. For each mixed object class, a multinomial distribution for each modality was prepared by mixing the distributions of the two pure object classes. Specifically, the multinomial distribution for the i-th mixed object was obtained by averaging those of the $(2i-1)$-th and the $2i$-th object classes. The observations for each modality of each object were drawn from the multinomial distributions corresponding to the object's class. The count of the BoFs for each modality was set to 20. Finally, 42 pure virtual objects and 21 mixed virtual objects were generated.

The experiment was performed almost in the same way as experiment 1. First, multimodal categorization was performed for the 63 virtual objects, and 14 categories were successfully formed in an unsupervised manner. The posterior distributions over the object categories are shown in **Figure 12**. Generally speaking, mixed objects were categorized into two or more classes. After categorization, a virtual robot was asked to recognize all of the target objects using the proposed active perception method.

6.2. Results

We compared the greedy, lazy greedy, and random algorithms for the active perception sequential decision making process. The random algorithm is a baseline method that determines the next action randomly from the remaining actions that have not been taken. In other words, the random algorithm is the case in which a robot does not employ any active perception algorithms.

The KL divergence from the final state for all target objects is averaged at each step and shown in **Figure 13**. For each condition, the KL divergence gradually decreased and reached almost zero. However, the rate of decrease was different. The greedy and lazy greedy algorithms were clearly shown to be better solutions on average than the random algorithm. In contrast with experiment 1, the best and worst cases could not practically be calculated because of the prohibitive computational cost.

Interestingly, the lazy greedy algorithm has almost the same performance as the greedy algorithm, as the theory suggests, although the laziness reduced the computational cost in reality.

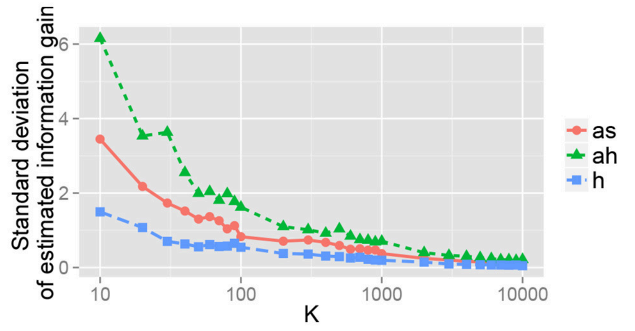


FIGURE 11 | Standard deviation of the estimated information gain IG_m for the 15th object. For each K, 100 values of the estimated information gain IG_m were obtained, and their standard deviation is shown.

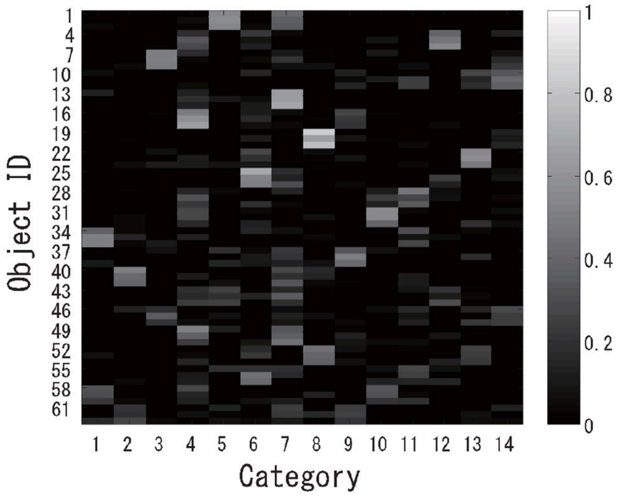


FIGURE 12 | Categorization results for the posterior probability distributions for each object.

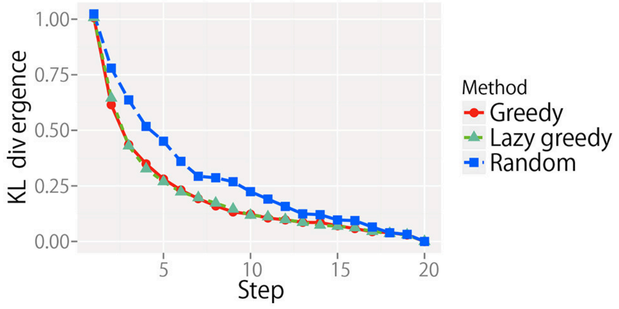


FIGURE 13 | KL divergence from the final state at each step for each sequential action selection procedure.

The number of times the robot evaluated IG_m to determine the action sequences for all executable counts of actions $L = 1, 2, \ldots, M$ is summarized for each method. The number of times the lazy greedy algorithm was required for each target object was 71.7 ($SD = 5.2$) on average, and that of the greedy algorithm was 190. Theoretically, the greedy and lazy greedy algorithms require $O(M^2)$ evaluations. Practically, the number of re-evaluations needed by the lazy greedy algorithm is quite small. In contrast, the brute-force algorithm requires $O(2^M)$ evaluations, i.e., far more evaluations of IG are required.

Next, a case in which two classes were assigned to the same object was investigated. The target dataset contained "mixed" objects. The results also imply that our method works well even when two classes are assigned to the same object. This is because our theory is completely derived on the basis of the probabilistic generative model, i.e., the MHDP. We show a typical result. **Figure 14** shows the time series of the posterior probability of the category for object 51, i.e., one of the mixed objects, during sequential active perception. This shows that the greedy and lazy greedy algorithms quickly categorized the target object into two categories "correctly." Our formulation assumes the categorization result to be a posterior distribution. Therefore, this type of probabilistic case can be treated naturally.

7. CONCLUSION AND DISCUSSION

In this paper, we described an MHDP-based active perception method for robotic multimodal object category recognition. We formulated a new active perception method on the basis of the MHDP (Nakamura et al., 2011b) .

First, we proposed an action selection method based on the IG criterion and showed that IG is an optimal criterion for active perception from the viewpoint of reducing the expected

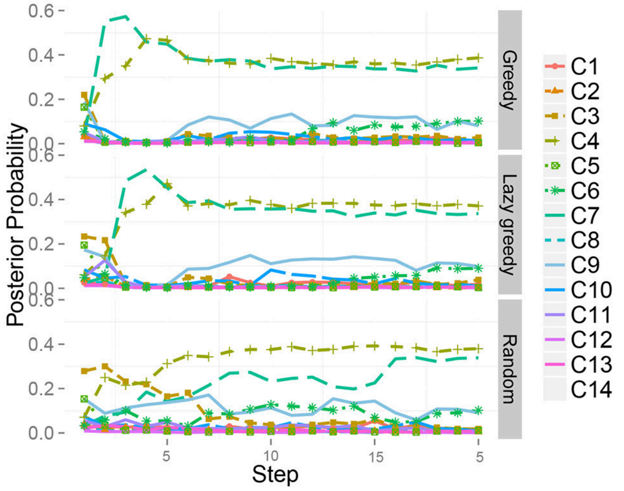


FIGURE 14 | Time series of the posterior probability of the category for object 51 during sequential action selection based on **(Top)** the greedy algorithm, **(Middle)** the lazy greedy algorithm, and **(Bottom)** the random selection procedure.

KL divergence between the final and current recognition states. Second, we proved that the IG has a submodular property and reduced the sequential active perception problem to a submodular maximization problem. Third, we derived a Monte Carlo approximation method for evaluating IG efficiently and made the action selection method executable. Given the theoretical results, we proposed to use the greedy and lazy greedy algorithms for selecting a set of actions for active perception. It is important to note that all of the three theoretical contributions mentioned above were naturally derived from the characteristics of the MHDP. These contributions are clearly a result of the theoretical soundness of the MHDP. In this sense, our theorems reveal a new advantage of the MHDP that other several heuristic multimodal object categorization methods do not have.

To evaluate the proposed methods empirically, we conducted experiments using an upper-torso humanoid robot and a synthetic dataset. Our results showed that the method enables the robot to actively select actions and recognize target objects quickly and accurately.

One of the most interesting points of this paper is that not only object categories but also an action selection for object recognition can be formed in an unsupervised manner. From the viewpoint of cognitive developmental robotics, providing an unsupervised learning model for bridging the development between perceptual and action systems is meaningful for shedding a new light on the computational understanding of cognitive development (Asada et al., 2009; Cangelosi and Schlesinger, 2015). It is believed that the coupling of action and perception is important for an embodied cognitive system (Pfeifer and Scheier, 2001).

The advantage of this paper compared with the related works in robotics is that our action selection method for multimodal category recognition has a clear theoretical basis and is tightly connected to the computational model for multimodal object categorization, i.e., MHDP. The theoretical basis gives the method preferable characteristics, i.e., theoretical guarantee.

However, note that the theoretical guarantee is satisfied only when IG is correctly estimated. We assumed that outcome of each action is deterministic and fully observable when we apply the theory of submodular optimization to active perception in multimodal categorization. However, observations X^m and IG are measured somehow probabilistically because of real-world uncertainty and Monte Carlo approximation. For example, IG is approximately estimated at each step of the greedy and lazy greedy algorithms. Theoretically, given this approximation in evaluating the objective being maximized, the $(1 - 1/e)$ bound no longer holds. Streeter et al. proposed to introduce an additional penalty based on a function approximation (Streeter and Golovin, 2009). Golovin et al. extended submodularity to adaptive submodularity to consider stochastic property (Golovin and Krause, 2011). Though we discussed the proposed method from the viewpoint of submodular optimization, this algorithm can be regarded as a version of the sequential information maximization, more specifically (Chen et al., 2015). Extending our idea by referring the adaptive submodularity and/or the sequential information maximization, and update our method is our future challenge.

We assumed that each action requires same cost, and tried to reduce the number of actions in active perception, i.e., to maximize the performance of perception with the fixed number of actions. However, practically, each action, e.g., shake, hit and look at, requires different duration and different energy. Therefore, practical cost is not always the number of actions, but total cost of actions. Zhang et al. (2017) tried to deal with this problem in the context of multimodal object identification. This problem leads us a knapsack problem-like formulation. This type of submodular optimization has been studied by many researchers (Streeter and Golovin, 2009; Zhou et al., 2013). Our method will be able to be extended in the similar way.

In addition to active perception, active "learning/exploration" for multimodal categorization is also an important research topic. It takes a longer time for a robot to gather multimodal information to form multimodal object categories from a massive number of daily objects than it does to recognize a new object. If a robot can notice that "the object is obviously a sample of learned category," the robot need not obtain knowledge about object categories from such an object. In contrast, if a target object appears to be completely new to the robot, the robot should carefully interact with the object to obtain multimodal information from the object. Such a scenario will be achieved by developing an active "learning/exploration" method for multimodal categorization. It is likely that such a method will be able to be obtained by extending our proposed active perception method.

Considering more complex categorization scenario is our future challenge. For example, Schenck et al. (2014) is dealing with the more complex categorization scenario, i.e., 36 plastic containers with identical shape and 3 colors, 4 types of contents, and 3 different amounts of those contents. In this paper, we used MHDP which assumes an object is classified into a single object category and infers the posterior distribution over categories. When we consider human cognition, we can find that object categories have more complex characteristics. For example, object categories have a hierarchical structure, an object is categorized into several classes, and they have different modality-dependency based on the types of categories. Unsupervised machine learning methods for such complex categorization problem have proposed by several researchers based on hierarchical Bayesian models (Griffiths and Ghahramani, 2006; Ando et al., 2013; Nakamura et al., 2015). Theoretically, the main assumption we used was that the MHDP is a hierarchical Bayesian model and action selection is corresponding to obtaining an observation which is a probabilistic variable on the leaf node of its graphical model. Therefore, by applying the same idea to the more complex categorization methods, we will be able to extend our theory to more complex categorization problems. This is on of our future works.

Another challenge lies in feature representation for multimodal categorization. The MHDP assumed that observations are given as bag-of-features representations. However, there are many kinds of feature representations for visual, auditory and haptic information. In particular, the feature extraction capability of deep neural networks is gathering attention, recently. Theoretically, our main theorems do not depend on the type of emission distributions, i.e., bag-of-features representations. It is likely that the same approach can be used even when a multimodal categorization method uses different feature representations, e.g., the features in the last hidden layer of a pre-trained deep neural network. This extension is also a part of our future challenges.

In addition, the MHDP model treated in this paper assumed that an action for perception is related to only one modality, e.g., grasping only corresponds to m^h. However, in reality, when we interact with an object with a specific action, e.g., grasping, shaking, or hitting, we obtain rich information related to various modalities. For example, when we shake a box to obtain auditory information, we also unwittingly obtain haptic information and information about its weight. The tight linkage between the modality information and an action is a type of approximation taken in this research. An extension of our model and the MHDP to a model that can treat actions that are related to various modalities is also a task for our future work.

AUTHOR CONTRIBUTIONS

The main theory was developed by TaT. The experiments were conceived by RY. The data were analyzed by RY and ToT with help of TaT. The manuscript was written by TaT.

FUNDING

This research was partially supported by Tateishi Science and Technology Foundation, and JST, CREST. This was also partially supported by a Grant-in-Aid for Scientific Research on Innovative Areas (16H06569) and a Grant-in-Aid for Young Scientists (B) (24700233) funded by the Ministry of Education, Culture, Sports, Science, and Technology.

ACKNOWLEDGMENTS

The authors would like to thank undergraduate student Takuya Takeshita and graduate student Hajime Fukuda of Ritsumeikan University, who helped us develop the experimental instruments for obtaining our preliminary results.

REFERENCES

Ando, Y., Nakamura, T., Araki, T., and Nagai, T. (2013). "Formation of hierarchical object concept using hierarchical latent dirichlet allocation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo), 2272–2279.

Araki, T., Nakamura, T., Nagai, T., Nagasaka, S., Taniguchi, T., and Iwahashi, N. (2012). "Online learning of concepts and words using multimodal LDA and hierarchical Pitman-Yor Language Model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Algarve), 1623–1630.

Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., et al. (2009). Cognitive Developmental Robotics: A Survey. *IEEE Trans. Auton. Mental Develop.* 1, 12–34. doi: 10.1109/TAMD.2009.2021702

Barsalou, L. W. (1999). Perceptual symbol systems. *Behav. Brain Sci.* 22, 1–16.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022.

Borotschnig, H., Paletta, L., Prantl, M., and Pinz, A. (2000). Appearance-based active object recognition. *Image Vision Comput.* 18, 715–727. doi: 10.1016/S0262-8856(99)00075-X

Burgard, W., Fox, D., and Thrun, S. (1997). "Active obile robot localization," in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)* (Nagoya), 1346–1352.

Cangelosi, A., and Schlesinger, M. (2015). *Developmental Robotics*. Cambridge, MA: The MIT press.

Celikkanat, H., Orhan, G., Pugeault, N., Guerin, F., Erol, S., and Kalkan, S. (2014). "Learning and Using Context on a Humanoid Robot Using Latent Dirichlet Allocation," in *Joint IEEE International Conferences on Development and Learning and Epigenetic Robotics (ICDL-Epirob)* (Genoa), 201–207.

Chen, Y., Hassani, S. H., Karbasi, A., and Krause, A. (2015). "Sequential information maximization: When is greedy near-optimal?" in *Conference on Learning Theory* (Paris), 338–363.

Cohn, D. A., Ghahramani, Z., and Jordan, M. I. (1996). Active learning with statistical models. *J. Artif. Intell. Res.* 4, 129–145.

Correa, J., and Soto, A. (2009). Active Visual Perception for Mobile Robot Localization. *J. Intell. Robot. Sys.* 58, 339–354. doi: 10.1007/s10846-009-9348-4

Denzler, J., and Brown, C. M. (2002). Information Theoretic Sensor Data Selection for Active Object Recognition and State Estimation. *IEEE Trans. Patt. Anal. Mach. Intell.* 24, 1–13. doi: 10.1109/34.982896

Dutta Roy, S., Chaudhury, S., and Banerjee, S. (2004). Active recognition through next view planning: a survey. *Patt. Recogn.* 37, 429–446. doi: 10.1016/j.patcog.2003.01.002

Eidenberger, R., and Scharinger, J. (2010). "Active perception and scene modeling by planning with probabilistic 6D object poses," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Taipei), 1036–1043.

Ferreira, J., Lobo, J., Bessiere, P., Castelo-Branco, M., and Dias, J. (2013). A Bayesian framework for active artificial perception. *IEEE Trans. Cyber.* 43, 699–711. doi: 10.1109/TSMCB.2012.2214477

Fishel, J. A. and Loeb, G. E. (2012). Bayesian exploration for intelligent identification of textures. *Front. Neurorobot.* 6, 1–20. doi: 10.3389/fnbot.2012.00004

Golovin, D., and Krause, A. (2011). Adaptive submodularity: theory and applications in active learning and stochastic optimization. *J. Artif. Intell. Res.* 42, 427–486. doi: 10.1613/jair.3278

Gouko, M., Kobayashi, Y., and Kim, C. H. (2013). "Online exploratory behavior acquisition of mobile robot based on reinforcement learning," in *26th International Conference on Industrial Engineering and Other Applications of Applied Intelligence Systems, IEA/AIE 2013* (Amsterdam), 272–281.

Griffith, S., Sinapov, J., Sukhoy, V., and Stoytchev, A. (2012). A behavior-grounded approach to forming object categories: Separating containers from noncontainers. *IEEE Trans. Auton. Mental Develop.* 4, 54–69. doi: 10.1109/TAMD.2011.2157504

Griffiths, T. L., and Ghahramani, Z. (2006). "Infinite latent feature models and the indian buffet process," in *Advances in Neural Information Processing Systems 2006* (Vancouver, BC), 475–482.

Harnad, S. (1990). The symbol grounding problem. *Phys. D* 42, 335–346.

Hogman, V., Bjorkman, M., and Kragic, D. (2013). "Interactive object classification using sensorimotor contingencies," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Tokyo), 2799–2805.

Ivaldi, S., Nguyen, S. M., Lyubova, N., Droniou, A., Padois, V., Filliat, D., et al. (2014). Object learning through active exploration. *IEEE Trans. Auton. Mental Develop.* 6, 56–72. doi: 10.1109/TAMD.2013.2280614

Iwahashi, N., Sugiura, K., Taguchi, R., Nagai, T., and Taniguchi, T. (2010). "Robots that learn to communicate: a developmental approach to personally and physically situated human-robot conversations," in *Dialog with Robots Papers from the AAAI Fall Symposium* (Palo Alto, CA), 38–43.

Ji, S., and Carin, L. (2006). Cost-Sensitive Feature Acquisition and Classification. *Patt. Recogn.* 40, 1474–1485. doi: 10.1016/j.patcog.2006.11.008

Kemp, C., Chang, K. M., and Lombardi, L. (2010). Category and feature identification. *Acta Psychol.* 133, 216–233. doi: 10.1016/j.actpsy.2009.11.012

Krainin, M., Curless, B., and Fox, D. (2011). "Autonomous generation of complete 3D object models using next best view manipulation planning," in *IEEE International Conference on Robotics and Automation* (Shanghai), 5031–5037.

Krause, A., and Guestrin, C. E. (2005). "Near-optimal nonmyopic alue of information in graphical models," in *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence* (Edinburgh).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge, UK: Cambridge University Press.

Minoux, M. (1978). "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*, ed J. Stoer (Berlin: Springer), 234–243.

Muslea, I., Minton, S., and Knoblock, C. A. (2006). Active learning with multiple views. *J. Art. Intell. Res.* 27, 203–233. doi: 10.1613/jair.2005

Nakamura, T., Ando, Y., Nagai, T., and Kaneko, M. (2015). "Concept formation by robots using an infinite mixture of models," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Hamburg), 4593–4599.

Nakamura, T., Nagai, T., Funakoshi, K., Nagasaka, S., Taniguchi, T., and Iwahashi, N. (2014). "Mutual learning of an object oncept and language model based on MLDA and NPYLM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'14)* (Chicago, IL), 600–607.

Nakamura, T., Nagai, T., and Iwahashi, N. (2007). "Multimodal object categorization by a robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, CA), 2415–2420.

Nakamura, T., Nagai, T., and Iwahashi, N. (2009). "Grounding of word meanings in multimodal concepts using LDA," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (St. Louis, MO), 3943–3948.

Nakamura, T., Nagai, T., and Iwahashi, N. (2011a). "Bag of multimodal LDA models for concept formation," in *IEEE International Conference on Robotics and Automation* (Shanghai), 6233–6238.

Nakamura, T., Nagai, T., and Iwahashi, N. (2011b). "Multimodal categorization by hierarchical dirichlet process," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Francisco, CA), 1520–1525.

Natale, L., Metta, G., and Sandini, G. (2004). "Learning haptic representation of objects," in *International Conference of Intelligent Manipulation and Grasping* (Genoa).

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions-I. *Math. Program.* 14, 265–294.

Pape, L., Oddo, C. M., Controzzi, M., Cipriani, C., Förster, A., Carrozza, M. C., et al. (2012). Learning tactile skills through curious exploration. *Front. Neurorobot.* 6:6. doi: 10.3389/fnbot.2012.00006

Pfeifer, R., and Scheier, C. (2001). *Understanding Intelligence*. A Bradford Book. Cambridge, MA: MIT Press.

Rebguns, A., Ford, D., and Fasel, I. (2011). "InfoMax control for acoustic exploration of objects by a mobile robot," in *AAAI11 Workshop on Lifelong Learning* (San Francisco, CA), 22–28.

Roy, D. K., and Pentland, A. P. (2002). Learning words from sights and sounds: a computational model. *Cogn. Sci.* 26, 113–146. doi: 10.1207/s15516709cog2601_4

Roy, N., and Thrun, S. (1999). "Coastal navigation with mobile robots," in *Advances in Neural Processing Systems 12*. Cambridge, MA: The MIT Press.

Russo, D., and Van Roy, B. (2016). An information-theoretic analysis of thompson sampling. *J. Mach. Learn. Res.* 17, 2442–2471. Available online at: http://jmlr.org/papers/v17/14-087.html

Saegusa, R., Natale, L., Metta, G., and Sandini, G. (2011). "Cognitive Robotics - Active Perception of the Self and Others," in *The 4th International Conference on Human System Interactions (HSI)* (Yokohama), 419–426.

Schenck, C., Sinapov, J., Johnston, D., and Stoytchev, A. (2014). Which object fits best? solving matrix completion tasks with a humanoid robot. *IEEE Trans. Auton. Mental Develop.* 6, 226–240. doi: 10.1109/TAMD.2014.2325822

Schneider, A., Sturm, J., Stachniss, C., Reisert, M., Burkhardt, H., and Burgard, W. (2009). "Object identification with tactile sensors using bag-of-features," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (St. Louis, MO), 243–248.

Settles, B. (2012). Active learning. *Synth. Lect. Artif. Intell. Mach. Learn.* 6, 1–114. doi: 10.2200/S00429ED1V01Y201207AIM018

Sinapov, J., Schenck, C., Staley, K., Sukhoy, V., and Stoytchev, A. (2014). Grounding semantic categories in behavioral interactions: experiments with 100 objects. *Robot. Auton. Sys.* 62, 632–645. doi: 10.1016/j.robot.2012.10.007

Sinapov, J., and Stoytchev, A. (2011). "Object category recognition by a humanoid robot using behavior-Grounded Relational Learning," in *IEEE International Conference on Robotics and Automation (ICRA)* (Shanghai), 184–190.

Stachniss, C., Grisetti, G., and Burgard, W. (2005). Information gain-based exploration using rao-blackwellized particle filters. in *Robotics Science and Systems (RSS)* (Cambridge, MA).

Streeter, M., and Golovin, D. (2009). "An online algorithm for maximizing submodular functions," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 1577–1584.

Sushkov, O. O., and Sammut, C. (2012). "Active robot learning of object properties," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Algarve: IEEE), 2621–2628.

Taniguchi, T., Nagai, T., Nakamura, T., Iwahashi, N., Ogata, T., and Asoh, H. (2016). Symbol emergence in robotics: a survey. *Adv. Robot.* 30, 706–728. doi: 10.1080/01691864.2016.1164622

Teh, Y., Jordan, M., Beal, M., and Blei, D. (2006). Hierarchical Dirichlet processes. *J. Am. Stat. Assoc.* 101, 1566–1581. doi: 10.1198/016214506000000302

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2005). "Sharing clusters among related groups: Hierarchical dirichlet processes," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 1385–1392.

Tuci, E., Massera, G., and Nolfi, S. (2010). Active categorical perception of object shapes in a simulated anthropomorphic robotic arm. *IEEE Trans. Evol. Comput.* 14, 885–899. doi: 10.1109/TEVC.2010.2046174

van Hoof, H., Kroemer, O., Ben Amor, H., and Peters, J. (2012). "Maximally informative interaction learning for scene exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Algarve), 5152–5158.

Velez, J., Hemann, G., Huang, A. S., Posner, I., and Roy, N. (2012). Modelling observation correlations for active exploration and robust object detection. *J. Artif. Intell. Res.* 44, 423–453. doi: 10.1613/jair.3516

Zhang, S., Sinapov, J., Wei, S., and Stone, P. (2017). "Robot behavioral exploration and multimodal perception using pomdps," in *AAAI 2017 Spring Symposium on Interactive Multisensory Object Perception for Embodied Agents* (Palo Alto, CA).

Zhou, J., Ross, S., Yue, Y., Dey, D., and Bagnell, J. A. (2013). "Knapsack constrained contextual submodular list prediction with application to multi-document summarization," *ICML 2013 Workshop on Inferning: Interactions between Inference and Learning* (Atlanta).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX A: PROOF OF THE OPTIMALITY OF THE PROPOSED ACTIVE PERCEPTION STRATEGY

In this appendix, we show that the proposed active perception strategy, which maximizes the expected KL divergence between the current state and the posterior distribution of $\mathbf{z}_j$ after a selected set of actions, minimizes the expected KL divergence between the next and final states.

$$
\begin{aligned}
\mathbf{A}_j^* &= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}}{\operatorname{argmin}} \mathbb{E}_{X_j^{\mathbf{M}\backslash\mathbf{m}_{\mathbf{o}j}}|X_j^{\mathbf{m}_{\mathbf{o}j}}}\left[\mathrm{KL}\left(P(\mathbf{z}_j|X_j^{\mathbf{M}}), P(\mathbf{z}_j|X_j^{\mathbf{A}\cup\mathbf{m}_{\mathbf{o}j}})\right)\right] \\
&= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}}{\operatorname{argmin}} \sum_{X_j^{\mathbf{M}\backslash\mathbf{m}_{\mathbf{o}j}}} \sum_{\mathbf{z}_j} \Big[P(X_j^{\mathbf{M}\backslash\mathbf{m}_{\mathbf{o}j}}|X_j^{\mathbf{m}_{\mathbf{o}j}})P(\mathbf{z}_j|X_j^{\mathbf{M}}) \\
&\quad \log\frac{P(\mathbf{z}_j|X_j^{\mathbf{M}})}{P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}, X_j^{\mathbf{A}})}\Big] \qquad \text{(A1)}
\end{aligned}
$$

The numerator inside of the log function does not depend on **A**. Therefore, the term related to the numerator can be deleted. In addition, by negating the remaining term, we obtain

$$
\begin{aligned}
\text{(A1)} &= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}}{\operatorname{argmax}} \sum_{X_j^{\mathbf{M}\backslash\mathbf{m}_{\mathbf{o}j}}} \sum_{\mathbf{z}_j}[P(X_j^{\mathbf{M}\backslash\mathbf{m}_{\mathbf{o}j}}|X_j^{\mathbf{m}_{\mathbf{o}j}})P(\mathbf{z}_j|X_j^{\mathbf{M}}) \\
&\quad \log P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}, X_j^{\mathbf{A}})] \\
&= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}}{\operatorname{argmax}} \sum_{X_j^{\mathbf{M}\backslash\mathbf{m}_{\mathbf{o}j}}} \sum_{\mathbf{z}_j}[P(\mathbf{z}_j, X_j^{\mathbf{M}\backslash\mathbf{m}_{\mathbf{o}j}}|X_j^{\mathbf{m}_{\mathbf{o}j}}) \\
&\quad \log P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}, X_j^{\mathbf{A}})]. \qquad \text{(A2)}
\end{aligned}
$$

By marginalizing $X_j^{\mathbf{M}\backslash(\mathbf{m}_{\mathbf{o}j}\cup\mathbf{A})}$ from (A2), we obtain

$$
\begin{aligned}
\text{(A2)} &= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}}{\operatorname{argmax}} \sum_{X_j^{\mathbf{A}}} \sum_{\mathbf{z}_j} P(\mathbf{z}_j, X_j^{\mathbf{A}}|X_j^{\mathbf{m}_{\mathbf{o}j}}) \log P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}, X_j^{\mathbf{A}}) \\
&= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}}{\operatorname{argmax}}\Big[\Big(\sum_{X_j^{\mathbf{A}}} \sum_{\mathbf{z}_j} P(\mathbf{z}_j, X_j^{\mathbf{A}}|X_j^{\mathbf{m}_{\mathbf{o}j}}) \log P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}}}, X_j^{\mathbf{A}})\Big) \\
&\quad \times \underbrace{\Big(-\sum_{\mathbf{z}_j} P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}) \log P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}})\Big)}_{\textit{constant w.r.t. } \mathbf{A}}\Big] \\
&= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}}{\operatorname{argmax}}\Big[\sum_{X_j^{\mathbf{A}}} \sum_{\mathbf{z}_j}[P(X_j^{\mathbf{A}}|X_j^{\mathbf{m}_{\mathbf{o}j}})P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}, X_j^{\mathbf{A}}) \\
&\quad \log P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}, X_j^{\mathbf{A}})] - \sum_{X_j^{\mathbf{A}}} \sum_{\mathbf{z}_j}[P(X_j^{\mathbf{A}}|X_j^{\mathbf{m}_{\mathbf{o}j}})P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}, X_j^{\mathbf{A}}) \\
&\quad \log P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}})]\Big] \\
&= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}}{\operatorname{argmax}} \sum_{X_j^{\mathbf{A}}} \sum_{\mathbf{z}_j}[P(X_j^{\mathbf{A}}|X_j^{\mathbf{m}_{\mathbf{o}j}})\, \mathrm{KL}\left(P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}}, X_j^{\mathbf{A}}),\right. \\
&\quad \left.P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}})\right)] \\
&= \underset{\mathbf{A}\in\mathbf{F}_L^{\mathbf{m}_{\mathbf{o}j}}}{\operatorname{argmax}} \mathbb{E}_{X_j^{\mathbf{A}}|X_j^{\mathbf{m}_{\mathbf{o}j}}}[\mathrm{KL}\left(P(\mathbf{z}_j|X_j^{\mathbf{A}\cup\mathbf{m}_{\mathbf{o}j}}), P(\mathbf{z}_j|X_j^{\mathbf{m}_{\mathbf{o}j}})\right)].
\end{aligned}
$$

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 08 June 2018
doi: 10.3389/fnbot.2018.00026

Affordance Equivalences in Robotics: A Formalism

Mihai Andries[1*†], *Ricardo Omar Chavez-Garcia*[2†], *Raja Chatila*[3], *Alessandro Giusti*[2] and *Luca Maria Gambardella*[2]

[1] Institute for Systems and Robotics (ISR-Lisboa), Instituto Superior Técnico, Lisbon, Portugal, [2] Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, USI-SUPSI, Lugano, Switzerland, [3] Institut des Systèmes Intelligents et de Robotique, Sorbonne Université, Centre National de la Recherche Scientifique, Paris, France

Automatic knowledge grounding is still an open problem in cognitive robotics. Recent research in developmental robotics suggests that a robot's interaction with its environment is a valuable source for collecting such knowledge about the effects of robot's actions. A useful concept for this process is that of an affordance, defined as a relationship between an actor, an action performed by this actor, an object on which the action is performed, and the resulting effect. This paper proposes a formalism for defining and identifying affordance equivalence. By comparing the elements of two affordances, we can identify equivalences between affordances, and thus acquire grounded knowledge for the robot. This is useful when changes occur in the set of actions or objects available to the robot, allowing to find alternative paths to reach goals. In the experimental validation phase we verify if the recorded interaction data is coherent with the identified affordance equivalences. This is done by querying a Bayesian Network that serves as container for the collected interaction data, and verifying that both affordances considered equivalent yield the same effect with a high probability.

Keywords: affordance, learning, cognitive robotics, symbol grounding, affordance equivalence

Edited by:
Tadahiro Taniguchi,
Ritsumeikan University, Japan

Reviewed by:
Ashley Kleinhans,
Ford Motor Company (United States),
United States
Lola Cañamero,
University of Hertfordshire,
United Kingdom

****Correspondence:***
Mihai Andries
mandries@isr.tecnico.ulisboa.pt

†These authors have contributed equally to this work.

Received: *30 September 2017*
Accepted: *16 May 2018*
Published: *08 June 2018*

Citation:
Andries M, Chavez-Garcia RO, Chatila R, Giusti A and Gambardella LM (2018) Affordance Equivalences in Robotics: A Formalism. Front. Neurorobot. 12:26. doi: 10.3389/fnbot.2018.00026

1. INTRODUCTION

Symbolic grounding of robot knowledge consists in creating relationships between the symbolic concepts used by algorithms controlling the robot and the physical concepts to which they correspond (Harnad, 1990). An *affordance* is a concept that allows collection of grounded knowledge. The notion of *affordance* was introduced by Gibson (1977), and refers to the action opportunities provided by the environment. In the context of robotics, an affordance is a relationship between an actor (i.e., robot), an action performed by the actor, an object on which this action is performed, and the observed effect.

A robot able to discover and learn the affordances of an environment can autonomously adapt to it. Moreover, a robot that can detect equivalences between affordances can quickly compute alternative plans for reaching a desired goal, which is useful when some actions or objects suddenly become unavailable.

In this paper, we introduce a method for identifying affordances that generate equivalent effects (see examples in **Figures 1**, **2**). We define a (comparison) operator that allows robots to identify equivalence relationships between affordances by analysing their constituent elements (i.e., actors, objects, actions).

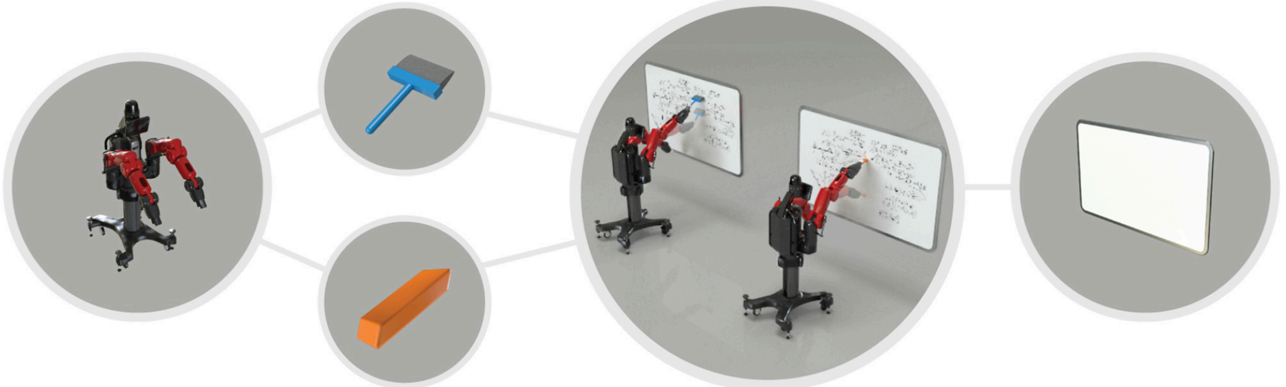


FIGURE 1 | Example of equivalence between two objects for cleaning a whiteboard: a wiper and an eraser. The robot affords to clean the white board by wiping it either with a wiper or an eraser.

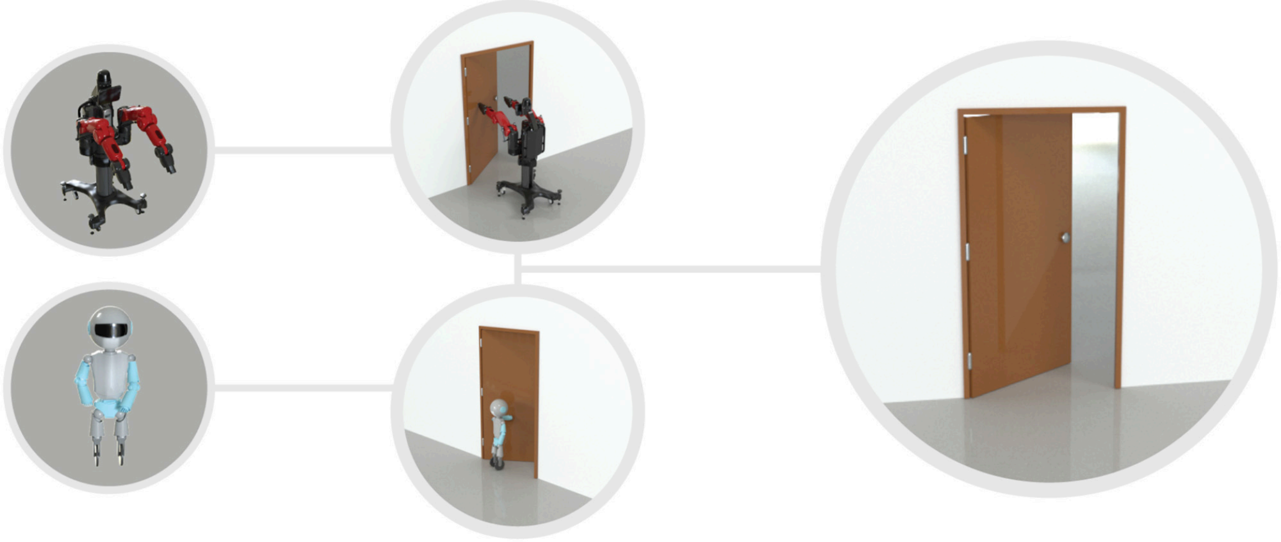


FIGURE 2 | Example of equivalence between different actors and their actions for opening a door. A door can be opened by any robot that can interact with the door.

1.1. Affordance Discovery and Learning

All methods proposed in the literature for affordance learning are similar in viewing an interaction as being composed of three components: an action, a target object, and a resulting effect. Different methods were proposed to infer the expected effect, given knowledge about the action and target object.

Several papers approached affordance learning as learning to predict object motion after interaction. For this purpose, Krüger et al. (2011) employed a feedforward neural network with backpropagation which learned so-called *object-action complexes*; Hermans et al. (2013) used Support Vector Machines (SVM) with kernels; while Kopicki et al. (2017) employed Locally Weighted Projection Regression (LWPR) with Kernel Density Estimation and a mixture of experts. Ridge et al. (2009) first used a Self-Organising Map and clustering in the *effect space* to classify objects by their effect, and then trained a SVM which identified to which cluster an object belongs using its feature-vector description.

Other papers addressed affordance learning from the perspective of object grasping. Stoytchev (2005) employed detection of invariants to learn object grasping affordances. Ugur et al. (2012) used SVMs to study the traversability affordance of a robot for grasping. Katz et al. (2014) used linear SVM to learn to perceive object affordances for autonomous pile manipulation. More details on the use of affordances for object manipulation can be found in the dissertation of Hermans (2014).

Some works followed a supervised training approach, providing hand-labeled datasets which mapped objects images (2D or RGB-D) to their affordances. Myers et al. (2015) learned affordances from local shape and geometry primitives using Superpixel-based Hierarchical Matching Pursuit (S-HMP), and Structured Random Forests (SRF). Image regions (from RGB-D frames) with pre-selected properties were tagged with specific affordance labels. For instance, a surface region with high convexity was labeled as *containable* (or a variation of it). Varadarajan and Vincze (2012) proposed an Affordance Network for providing affordance knowledge ontologies for common household articles, intended to be used for object recognition and manipulation. An overview of machine learning approaches for detecting affordances of tools in 3D visual data is available in the thesis of Ciocodeica (2016).

Another approach for learning affordances uses Bayesian Networks. Montesano et al. (2008) and Moldovan et al. (2012) employed a graphical model approach for learning affordances, using a Bayesian Network which represents objects/actions/effects as random variables, and which encodes relations between them as dependency links. The structure of this network is learned based on the data of robot's interaction with the world and on *a priori* information related to the dependency of some variables. Once learned, affordances encoded in this way can (1) predict the effect of an action applied to a given object, (2) infer which action on a given object generated an observed effect, and (3) identify which object generates the desired effect when given a specific action.

Yet another popular method for supervised affordance learning uses Deep Learning techniques. For instance, Nguyen et al. (2016) trained a convolutional neural network to identify object affordances in RGB-D images, employing a dataset of object images labeled pixelwise with their corresponding affordances. A similar approach using a deep convolutional neural network was taken by Srikantha and Gall (2016).

Recent comprehensive overviews of affordance learning techniques are available in the dissertation of Moldovan (2015), and in reviews by Jamone et al. (2016), Min et al. (2016), and Zech et al. (2017).

We argue that once affordances are learned, we can find relations between affordances by considering the effects they generate. One of these relations is equivalence, i.e., when two different affordances specify corresponding actions on objects that generate the same effect.

1.2. Affordance Equivalence

Affordance equivalence was studied by Şahin et al. (2007), who considered relationships between single elements of an affordance. Thus, it was possible to identify objects or actions that are equivalent with respect to an affordance when they generate the same effect. Griffith et al. (2012) employed clustering to identify classes of objects that have similar functional properties. Montesano et al. (2008) and Jain and Inamura (2013) treated affordance equivalence from a probabilistic point of view, where, in the context of imitation learning, the robot searches for the combination of action and effect that maximises their similarity to the demonstrated action on an object. Boularias et al. (2015) discovered through reinforcement learning the graspability affordance over objects with different shapes, and indirectly showed equivalence of the grasp action.

Developing this line of thought, we propose a probabilistic method to identify which *combinations of affordance elements* generate equivalent effects. We first present in section 2 the affordance formalization employed, and based on that we then list in section 2.4 all the possible types of affordance equivalences.

Since the purpose of this study is to identify equivalences between affordances that were already recorded by the robot, we are not seeking to explain how to record these affordances. In this paper we employed the graphical model approach for learning affordances proposed by Montesano et al. (2008). In addition, we rely purely on perception-interaction data, without using *a priori* information (Chavez-Garcia et al., 2016b). To facilitate the experimental setup, we used pre-defined sensorial and motor capabilities for our robots.

The remainder of this paper is organized as follows. In section 2, we introduce our formalization of affordance elements, and define the equivalence relationship in section 2.4. A series of experiments on the discovery of equivalences between affordances is detailed in section 3, together with the obtained results. We conclude and present opportunities for future work in section 4.

2. METHODOLOGY: AFFORDANCE FORMALIZATION

In this section, we present the affordance formalism employed throughout the paper. We follow the definition proposed by Ugur et al. (2011), that we enrich by including the actor performing the action into the affordance tuple *(object, action, effect)*. The inclusion of the actor into the affordance allows robots to record affordances specific to their body morphologies. Although we will not focus on this aspect in this paper, it is possible to generalize this knowledge through a change of affordance perspective from robot joint space to object task space (more about this in section 2.1.2).

We define an affordance as follows. Let G be the set of actors in the environment, O the set of objects, A the set of actions, and E the set of observable effects. Hence, when an *actor* applies an *action* on an *object*, generating an *effect*, the corresponding affordance is defined as a tuple:

$$\alpha = (actor, object, action, effect),\ \text{for}\ actor \in G, object \in O,\\ action \in A,\ \text{and}\ effect \in E, \tag{1}$$

and can be graphically represented as shown in **Figure 3**. From actor perspective, it interacts with the environment (the object) and discovers the affordances. From object perspective, affordances are properties of objects which can be perceived by actors, and which are available to actors with specific capabilities. We can also consider observers, who learn by perceiving other actors' affordance acquisition process.

The way in which affordance elements are defined influences the operations that can be performed with affordances. Since we aim to establish equivalence relationships between affordances, we will analyse the definitions of the following affordance elements: actions (from actor and object perspectives), objects (as perceived by robot's feature detectors), and effects (seen as a description of the environment).

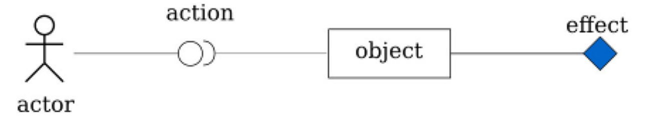


FIGURE 3 | A graphical representation of an affordance. An object accepts any action that fits its interface (shown on object's left), and produces the specified effect (shown on object's right). Any actor capable of performing the expected action on this object can produce the described effect.

2.1. How Are Actions Defined?

Actions can be defined (1) relative to actors, by describing the body control sequence during the execution of an action in joint space; or (2) relative to objects, by describing the consequences of actions on the objects in operational space. We refer to *object perspective* when the actions are defined in the operational/task space, making their definition independent of the actor executing them. We refer to *actor perspective* when the actions are defined in the joint space of the actor, making them dependent of the actor executing them.

This statement comes from the different perspectives obtained from the affordance definition in Equation (1): *actor* and *object* perspective.

2.1.1. Actions Described Relative to Actors

Actions are here described relative to actors and their morphology. They are defined with respect to their control variables in joint space (i.e., velocity, acceleration, jerk), indexed by time τ:

$$action : \{Q, \dot{Q}, \ddot{Q}\}_\tau \tag{2}$$

As the action is described with respect to the actor morphology and capabilities, comparing two actions requires comparing both the actors performing the actions, and the actions themselves. When the actors are identical, the action comparison is straightforward. However, when there is a difference between actors' morphologies (and their motor capabilities), the straightforward comparison of actions is not possible and a common frame of reference for such comparison is needed.

2.1.2. Actions Described Relative to Objects

When actions are described relative to objects, they represent an action generalisation from the *joint space* of a particular actor (where actions are defined on the actor) to the *operational space* of any actor (where actions are defined on the object).

Thus, when actions are described relative to objects, the *actor* can be omitted from the affordance tuple, to indicate that any actor which has the required motor capabilities is able to generate the action which causes this effect. In addition, the action employed in this representation is defined in *operational space* (and not in *joint space* as before). Hence, dropping the actor from the equation, we can rewrite Equation (1) as:

$$\alpha = (object, action, effect), \text{ for } object \in O, action \in A_o, \text{ and } effect \in E \tag{3}$$

where A_o is the set of all actions in operational space, applicable to object o.

While affordances defined from actor perspective (in joint space, e.g., joint forces to apply) allow to learn using robot's motor and perceptual capabilities, affordances defined from object perspective (in task space, e.g., forces applied on the object) allow to generalise this knowledge.

2.2. How Are Objects Defined?

If an actor has the feature detectors $p_1, \ldots, p_n$ corresponding to its perception capacities (such as hue, shape, size), then an object is defined as:

$$object = \{p_1, \ldots, p_n\}, \tag{4}$$

where each feature detector can be seen as function on a perceptual unit (e.g., a salient segment from a visual perception process).

2.3. How Are Effects Defined?

We suppose that an actor g has a set ξ of q effect detectors, that are able to detect changes in the world after an action $a \in A_g$ is applied. For example, when an actor executes action *push* on an object, the object-displacement-effect detector would be a function that computes the difference between two measurements of the object position taken before and after the interaction. Another effect can be the difference in the feedback force measured in the end effector before and after the interaction. Formally, effects are a set of q salient changes in the world ω (i.e., in the target object, the actor, or the environment), detected by robot's effect detectors ξ:

$$effect = \{\xi_1(\omega), \ldots, \xi_q(\omega)\} \tag{5}$$

2.4. Affordance Equivalence Operator

In this section, we introduce the concept of affordance equivalence, based on the formalization presented earlier in section 2. We provide truth tables for two different affordance comparison operators: one for the case where actions are defined in actor joint space, and one for the case where actions are defined in object task space. For each case, we explore the possible types of affordance equivalence.

We have defined an affordance as a tuple of type $(actor, object, action_{joint\_space}, effect)$ when the action is defined relative to the actor, or as a tuple of type $(object, action_{operational\_space}, effect)$ when the action is defined relative to the object. Let us now define the truth table for an operator for comparing affordances (one for the actor perspective, and one for the object perspective) and identifying equivalence relationships between them.

We consider equivalent two affordances that generate equivalent effects. To know when two effects are equivalent, an effect-comparison function is required. We define an equivalence function $f(e_a, e_b)$ that yields true if two effects values e_a and e_b are similar in a common frame (e.g., distances for position values, similarity in color models, vector distances for force values). We detect affordance equivalence by (1) feeding the continuous (non-discretised) data on the measured effects to the Bayesian

Network (BN) structure learning algorithm, and then (2) querying the BN over an observed effect to obtain the empirical decision on effect equivalence. Whenever two affordances generate equivalent effects, it is possible to find which affordance elements cause this equivalence. We distinguish several cases of affordance equivalence, depending on the elements which differ in two equivalent affordances, which are detailed below.

2.4.1. Equivalence Between Affordance With Actions Defined Relative to Actors

The comparison cases for affordances with actions described relative to actors are shown in **Table 1**. The 2^4 cases of comparison between the elements of two affordances stem from all the possible (binary) equivalence combinations between the elements. In each case we compare the four components and establish if the elements of affordances are equivalent.

Since actions are defined here relative to the actors, actors with different morphologies cannot perform the same action defined in joint space, because their joint spaces are different. This renders inconsistent cases in which different actors perform the same action: lines (3), (4), (7), and (8) in **Table 1**. This leaves us with five cases of equivalence in **Table 1**, where:

- If different actors using different actions on different objects generate an equivalent effect, then we have *(actor, action, object) equivalence*
- If different actors using different actions on the same object generate an equivalent effect, then we have *(actor, action) equivalence*
- If the same actor using different actions on different objects generates an equivalent effect, then we have *(object, action) equivalence*
- If the same actor using the same action on different objects generates an equivalent effect, then we have *object equivalence*
- If the same actor using different actions on the same object generates an equivalent effect, then we have *action equivalence.*

We assume that the environment is a deterministic system: each time the same actor applies the same action on the same object, it will generate an equivalent effect. Therefore, generating a different effect with the same actor, action, and object is impossible, due to determinism.

Both the effect equivalence and non-equivalence cases provide information about the relationship between two affordances. The affordance equivalence concept is empirically validated in section 3.

2.4.2. Equivalence Between Affordances With Actions Defined Relative to Objects

The comparison cases for affordances with actions described relative to objects are shown in **Table 2**. There are 2^3 cases of comparison, corresponding to the total number of possible (binary) equivalence cases between the elements of a pair of affordances. In this case, three types of equivalence exist:

- If different actions on different objects generate the same effect, then it is (*object*, *action*) equivalence;
- If same action on different objects generates the same effect, then it is *object* equivalence;
- If different actions on same object generate the same effect, then it is *action* equivalence.

3. EXPERIMENTS AND RESULTS: AFFORDANCE EQUIVALENCE

We designed experiments that would confirm the capability of our affordance representation to detect equivalences and non-equivalences between learned affordances. We employed a Bayesian Network structure-learning approach presented in (Chavez-Garcia et al., 2016a) to describe and learn affordances as relations between random variables (affordance elements). Then we analyse how the learned affordances relate to each case of equivalence presented in **Table 2**.

3.1. Pre-defined Actions

We assume that an agent is equipped, since its conception, with motor and perceptual capabilities that we called *pre-defined*. However, we do not limit the agent's capabilities to the pre-defined set, as through learning the agent may acquire new capabilities. In our scenario, we employed three robotic actors of different morphologies, each with its pre-defined actions:

1. Baxter$_{\text{gripper}}$: the Baxter robot's left arm (7 DOF) equipped with a gripper, with actions:
    - Push (moving with constant velocity without closing the gripper)
    - Pull (closing the gripper and moving with constant velocity)
    - Wipe (closing the gripper and pressing downwards while moving)
    - Move aside (closing the gripper and moving aside)
2. Baxter$_{\text{nogripper}}$: the Baxter robot's right arm with no gripper, with action:
    - Poke (moving forwards with constant acceleration)
3. Katana arm with no gripper (5 d.o.f.), with action:
    - Side push (moving aside with constant velocity)

The actors and their pre-defined sets of actions (motor capabilities) are shown in **Figure 4**.

3.2. Pre-defined Perceptual Capabilities

Our visual perception process takes raw RGB-D data of an observed scene to oversegment the point cloud into a supervoxel representation. This 3D oversegmentation technique is based on a flow-constrained local iterative clustering which uses color and geometric features from the point cloud (Papon et al., 2013). Strict partial connectivity between voxels guarantees that supervoxels cannot flow across disjoint boundaries in 3D space. Supervoxels are then grouped to obtain object clusters that are used for extracting features and manipulation. **Figure 5** illustrates the visual perception process. The objects employed were objects of daily use: toys that can

TABLE 1 | Comparison of two affordances, when actions are described with respect to actors.

#	Actors	Objects	Actions	Effects	Conclusion
1	different	different	different	different	*(actor, object, action)* non-equivalence
2	different	different	different	equivalent	***(actor, object, action)* equivalence**
3	different	different	same	different	*(actor, object)* non-equivalence
4	different	different	same	equivalent	***(actor, object)* equivalence**
5	different	same	different	different	*(actor, action)* non-equivalence
6	different	same	different	equivalent	***(actor, action)* equivalence**
7	different	same	same	different	*actor* non-equivalence
8	different	same	same	equivalent	***actor* equivalence**
9	same	different	different	different	*(object, action)* non-equivalence
10	same	different	different	equivalent	***(object, action)* equivalence**
11	same	different	same	different	*object* non-equivalence
12	same	different	same	equivalent	***object* equivalence**
13	same	same	different	different	*action* non-equivalence
14	same	same	different	equivalent	***action* equivalence**
15	same	same	same	different	impossible in deterministic systems
16	same	same	same	equivalent	**due to determinism**

Equivalence cases between affordances are presented in even rows. Inconsistencies are underlined in red. The types of affordance equivalence are shown in bold letters.

TABLE 2 | Comparison of two affordances, when actions are described with respect to objects.

#	Objects	Actions	Effects	Conclusion
1	different	different	different	*(object, action)* non-equivalence
2	different	different	equivalent	***(object, action)* equivalence**
3	different	same	different	*object* non-equivalence
4	different	same	equivalent	***object* equivalence**
5	same	different	different	*action* non-equivalence
6	same	different	equivalent	***action* equivalence**
7	same	same	different	impossible in deterministic systems
8	same	same	equivalent	**due to determinism**

Equivalence cases between affordances are presented in even rows. The types of affordance equivalence are shown in bold letters.

be assembled, markers, and dusters. The objects were selected so as to be large enough to allow easy segmentation and manipulation.

3.3. Pre-defined Effect Detectors

We used custom hand-written effect detectors for the experimental use-cases, although our experimental architecture allows for an automatic effect detector. An effect detector quantifies the change, if present, in one property of the environment or the actor. For this series of experiments, we developed the following effect detectors: color change in a 2D image (HSV hue) for an object or a region of interest; object's position change (translation only); and the end-effector position. **Figure 6** illustrates the detected effects when *wipe* action is performed. In our previous work we covered changes in joint torques, distance between finger grippers and object speed.

3.4. Affordance Learning

Affordance elements E (effects), O (objects) and A (actions) are represented as random variables of a Bayesian Network (BN) $\mathcal{B}$. First, in each actor interaction we record the values (discretized) for the random variables representing the objects (section 3.2), actions (section 3.1), and effects (section 3.3). The problem of discovering the relations between E, O, and A can be then translated to finding dependencies between the variables in $\mathcal{B}$, i.e., $P(\mathcal{B}|\mathcal{D})$ learning the structure of the corresponding network $\mathcal{B}$ from data $\mathcal{D}$. Thus, affordances are described by the conditional dependencies between variables in $\mathcal{B}$.

We implemented an information-compression score to estimate how well a Bayesian Network structure describes data $\mathcal{D}$ (Chavez-Garcia et al., 2016b). Our score is based on the Minimum Description Length (MDL) score:

$$MDL(\mathcal{B}|\mathcal{D}) = LL(\mathcal{B}|\mathcal{D}) - |\mathcal{B}|\frac{logN}{2}, \tag{6}$$

where the first term measures (by applying a log-likelihood score Suzuki, 2017) how many bits are needed to describe data $\mathcal{D}$ based on the probability distribution $P(\mathcal{B})$. The second term counts the number of bits needed to encode $\mathcal{B}$, where $\frac{log(N)}{2}$ bits are used for each parameter in the BN. We consider $\frac{log(N)}{2}$ as factor that penalizes structures with larger number of parameters. For a BN's structure $\mathcal{B}$, its score is then defined as the posterior probability given the data $\mathcal{D}$.

We implemented a search-based structure learning algorithm based on the hill-climbing technique, as we did in our previous work. As inputs, this algorithm takes values for the variables in E, O, and A obtained from robot's interaction. This procedure estimates the parameters of the local probability density functions (pdfs) given a Bayesian Network structure. Typically, this is a

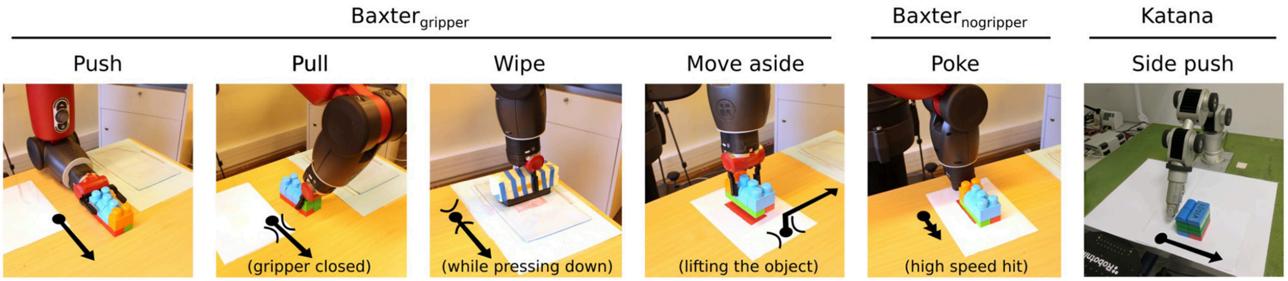


FIGURE 4 | Set of pre-defined actions for three actors: Baxter$_{gripper}$ equipped with a 7 d.o.f. arm, and an electrical gripper attached to it, Baxter$_{nogripper}$ equipped only with a 7 d.o.f. arm, and Katana 5 d.o.f. arm without gripper. Poke is the only pre-defined action of actor Baxter$_{nogripper}$, and side push the only pre-defined action of Katana. The arrows show the direction of the manipulator movement. The arcs show the position of the gripper with respect to the object, while the black bullet represents the object.

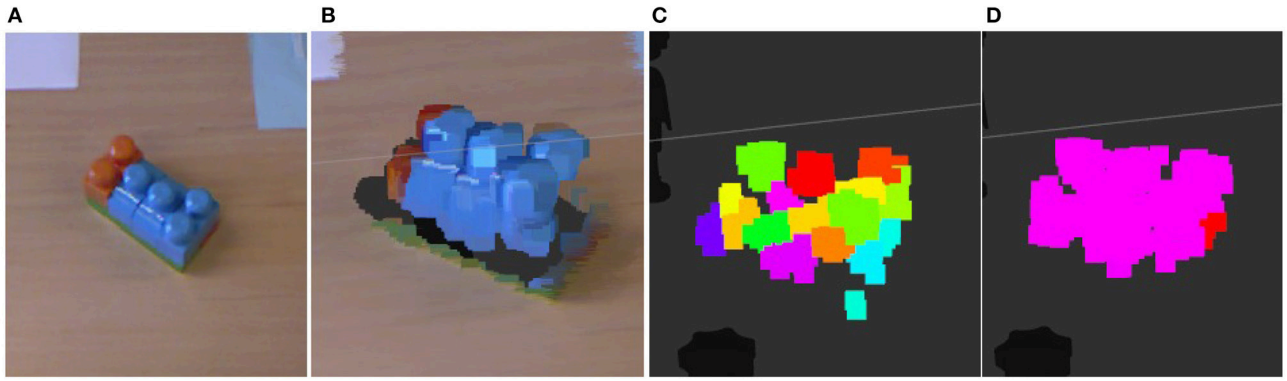


FIGURE 5 | An example of the visual perception process output. From left to right: **(A)** reference image **(B)** RGB cloud of points of the scene **(C)** supervoxel extraction **(D)** clusterization of supervoxels. For visual perception we use a Microsoft Kinect sensor that captures RGB-D data.

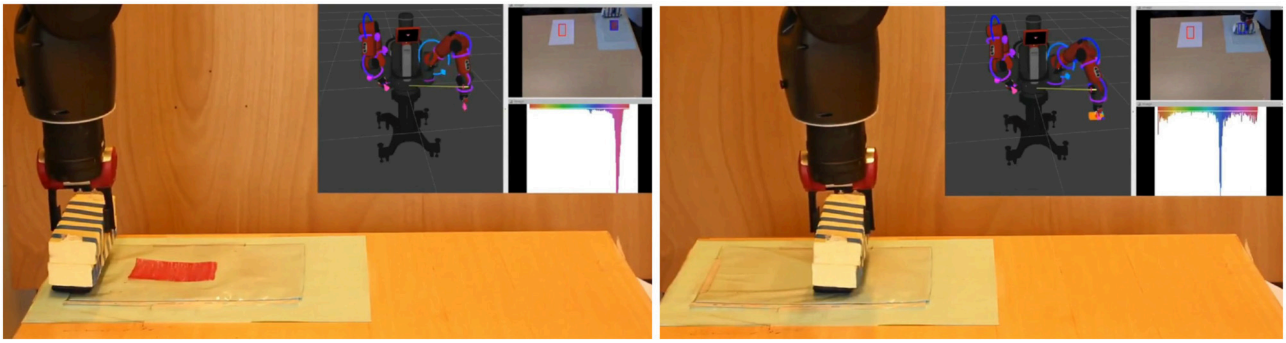


FIGURE 6 | Example of captured effects when performing the action *wipe* on the object *duster*. Left figure shows the spatial (pose) and perceptual (color) state of the *duster*, and the *surface*. After *wipe* action is performed, the effects on position and in hue are detected: *duster* has changed position but not color, *surface* has changed color but not position. Although for this experiment we do not use the force in the joints, we are also capturing these changes.

maximum-likelihood estimation of the probability entries from the data set, which, for multinomial local pdfs, consists of counting the number of tuples that fall into each table entry of each multinomial probability table in the BN. The algorithm's main loop consists of attempting every possible single-edge addition, removal, or reversal, making the network that increases the score the most the current candidate, and iterating. The process stops when there is no single-edge change that increases the score. There is no guarantee that this algorithm will settle at a global maximum, but there are techniques to increase its reaching possibilities (we use simulated annealing).

By using the BN framework, we are capable of displaying relationships between affordance elements. The directed nature of its structure allows us to approximate cause-effects relationships. It also handles uncertainty through the established probability theory. In addition to direct dependencies, we can represent indirect causation.

3.4.1. Detection of Affordance Equivalence

Equivalence between two affordances can be identified by comparing their ability to consistently reproduce the same effect e, judging by the cumulated experimental evidence. The precise type of equivalence between two affordances, which tells which affordance elements' values are equivalent, can be identified by probabilistic inference on the learned BN. Inference allows to identify which *(actor, object, action)* configurations are more likely to generate the same effect. In practice, this inference is calculated through executing queries to the Bayesian Network, which allow to compute the probability of an event (in our case: the probability of an effect having a value between some given bounds) given the provided evidence data.

Queries have the following form: $P(proposition|evidence)$ where *proposition* represents the query on some variable x, and *evidence* represents the available information for the affordance elements, e.g., the identity of the actor, the description of the action, and the description of the object. In the example of the robot pushing an object, the following query allows to compute the probability of the object displacement falling between certain bounds:

$$P\big((position > lower\ bound) \text{ and } (position < upper\ bound) \mid \\ actor = Baxter, action = push, object = block\big) \tag{7}$$

After querying the learned BN with the corresponding elements from **Tables 1**, **2** as evidence, if two *(actor, object, action)* configurations have probabilities of generating an effect that are higher than an arbitrary threshold, then we consider both affordances equivalent:

$$\begin{aligned} &\textbf{if } P(e|actor_1, object_1, action_1) > \theta \\ &\textbf{and } P(e|actor_2, object_2, action_2) > \theta \\ &\textbf{then } (actor_1, object_1, action_1) \equiv (actor_2, object_2, action_2) \end{aligned} \tag{8}$$

For our experiments, we empirically established the equivalence threshold $\theta = 0.85$. The aforementioned querying process connects the learning and reasoning steps, and according to the current goal of an actor, it allows for an empirical threshold selection or an adaptive mechanism.

3.5. Experimental Results

As shown in **Table 1**, affordances composed of 4 elements (actor, object, action, effect), which have their actions defined from the actor perspective, have five cases of equivalence (see **Figure 7** for some illustrated examples). We have selected three of them to demonstrate the use of the affordance equivalence operator: (object) equivalence, (action) equivalence, and (actor, action) equivalence. In **Figure 7** they correspond to the settings (a), (b), and (c). These experiments are detailed below. For a video demonstration of these experiments, please see the Supplementary Material section at the end of this document.

3.5.1. The (Actor, Action) Equivalence

This experiment consisted in discovering the equivalence between (actor, action) tuples. The goal was to identify configurations that are equivalent in their ability of uncovering a region of interest (a red mark on the table) by moving the object occluding it from robot's camera view (in the case of the Baxter — a toy with features color: blue and shape: box; in the case of the Katana actor — a box with the same perceptual features). In our representation, two objects with the same perceptual features are considered the same. Actor Baxter$_{\text{gripper}}$ is equipped with a gripper and can perform action *move_aside*. Actor Baxter$_{\text{nogripper}}$ does not have a gripper and can only perform action *poke*. Actor Katana does not have a gripper and can only perform *side push* action.

The Bayesian Network structure was learned using data from 15 interactions using each (actor, action) tuple (**Figure 8**). Variables object_shape and object_color represent the object features, variable color_mark captures the presence or absence of a colored mark. Queries performed on the BN suggested that the effect of revealing the red mark is consistently recreated when moving the object toy, with a probability of 0.98 for the action move_aside done by the hand with a gripper, 0.97 for the action poke done by the hand with no gripper, and 0.94 for the action side_push done by the Katana arm on the box object. The probabilities are based on the total number of trials verifying these relationships. Since these affordances consistently recreate equivalent effects while having some equal elements (same toy object for Baxter$_{\text{gripper}}$ and Baxter$_{\text{nogripper}}$, and a similar object for Katana), this points that affordance elements that differ between configurations are in fact equivalent in their ability to generate the effect of revealing the red mark, i.e., the tuples (Baxter$_{\text{nogripper}}$, poke), (Baxter$_{\text{gripper}}$, move_aside) and (Katana, side_push). Source code of the experimental setup for the Katana actor is available at https://romarcg@bitbucket.org/romarcg/katana_docker.git.

3.5.2. The (Object) Equivalence

The experiment consisted in determining the equivalence between two visually different whiteboard dusters: duster$_{\text{blue}}$ and duster$_{\text{orange}}$. Actor Baxter$_{\text{gripper}}$ applies the same action wipe to remove a red marker trace from a blue colored surface, as shown in **Figures 4**, **6**. For distinguishing the clean blue colored surface from the surface dirtied with the red marker, the robot's predefined effect detector measured the effect on the hue extracted from an HSV histogram.

The robot performed 25 trials of the *wipe* action with each duster, and the obtained data was subsequently used to learn the Bayesian Network structure (see **Figure 8B**). Objects are represented in the same way as in section 3.5.1. The effect capturing the change in the wiped area is described by the variable color_effect. Queries revealed that the wipe action cleans the red marker trace from the blue colored surface with a probability of 0.95 in both cases. Since the observed effects were equivalent, and the actor and action were the same, the objects duster$_{\text{blue}}$ and duster$_{\text{orange}}$ are considered equivalent in their ability to reach this effect.

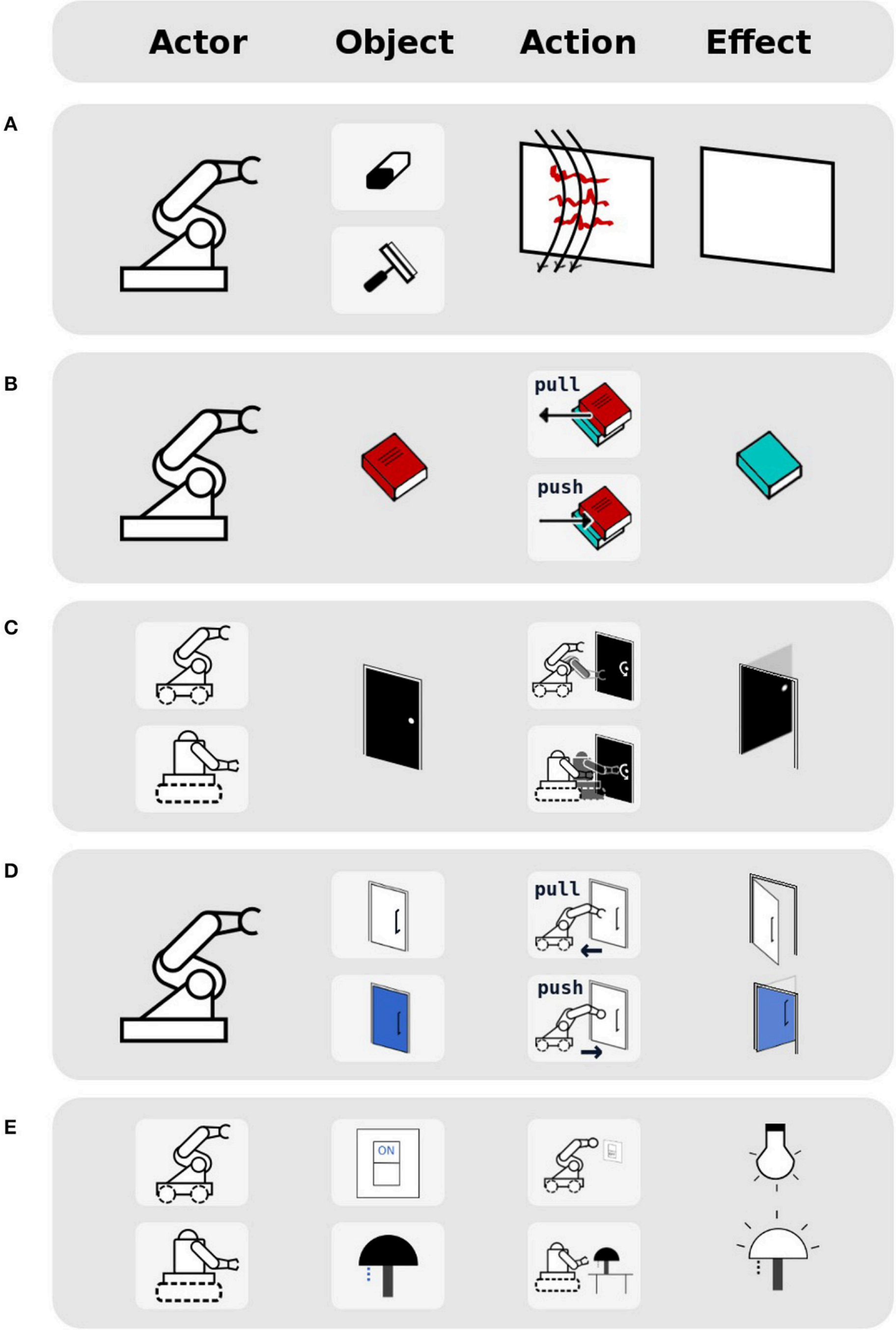


FIGURE 7 | Illustrated examples for each of the five types of affordance equivalences, from the actor perspective, when affordances are represented as *(actor, object, action, effect)* tuples: **(A)** A robot can use two different objects (wiper/eraser) to obtain the same effect of obtaining a clean whiteboard when performing wipe action. **(B)** A robot can perform two different actions (push/pull) to obtain the same effect of revealing a book underneath. **(C)** Two different robots can perform two different actions on the same object to obtain the same effect of opening a door. **(D)** A robot can perform two different actions (pull/push) on two different objects (door handle/door) in order to obtain the same effect of opening those doors. **(E)** Two robots can apply two different actions on two different objects (light switch, lamp) to obtain the effect of turning on the light.

3.5.3. The (Action) Equivalence

In this experiment we analysed equivalence between the actions of an actor. This experiment consisted in placing the same object *toy* into a desired location using two different actions *push* and *pull* of the actor Baxter$_{\text{gripper}}$. The robot performed 30 trials using each of the push and pull actions. **Figure 8C** shows the learned BN for (action) equivalence. The arrival of the object (described as in previous experiments) to the desired position is described

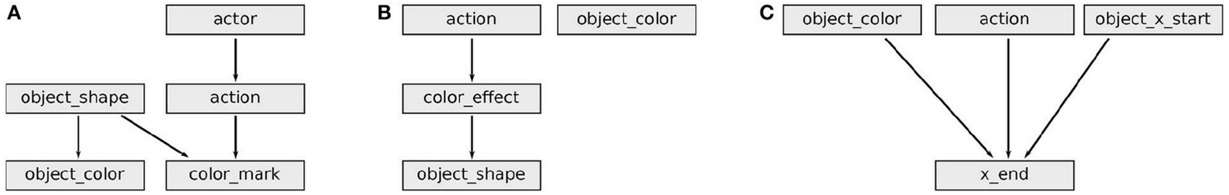


FIGURE 8 | Learned Bayesian networks for the experiments. **(A) (actor, action) equivalence** between Baxter and Katana actors using different movements to reveal a colored region of interest (section 3.5.1). The BN shows the dependence between the chosen actors and actions and the revelation of the colored region of interest. **(B) (object) equivalence** between two dusters of different colors that clean a whiteboard (section 3.5.2). The BN shows the irrelevance of object_color feature for the wiping affordance. **(C) (action) equivalence** between push/pull actions (section 3.5.3). The BN shows the relation between the chosen action and the final displacement of the object (feature x_end).

by the effect variable x_end (only the x component of the 3D position was measured). The target location to which we aim to push/pull the object is at x coordinate 0.72 ± $0.02m$. Variable object_x_start is an object feature representing the object initial position. According to the BN that processed the obtained data, there was a 0.97 probability to *pull* the object to the desired location, and a 0.89 chance to do so by *pushing* it. With all the rest being equal (the actor, object, and effect are the same), and since both actions have a high probability of generating the given effect, these *push* and *pull* actions can be considered equivalent for placing the object *toy* in a desired location.

4. CONCLUSIONS AND FUTURE WORK

We have presented a formalization for affordances with respect to their elements, and the equivalence operator for comparing two affordances from the actor and object perspective. We performed Bayesian Network structure learning to capture affordances as sensorimotor representations based on the observed experimental data. We analysed and validated experimentally the affordance equivalence operator, demonstrating how to extract information on the tuples of actors, actions and objects by comparing two affordances and determining if such tuples are equivalent.

In practice, the learned affordance equivalences can be interchangeably used when some objects or actions become unavailable. In a multi-robot setting, these equivalences can allow an ambient intelligence (an Artificial Intelligence system controlling an environment) to select the appropriate robot for using an affordance to reach a desired effect.

4.1. Future Work

Our future work will focus on the domain of transfer learning. We plan to implement a transformation between the affordances learned by specific robots (in their own joint space) to affordances applicable to objects and defined in their operational space. This will generalise the affordances learned and perceivable by a robot with a specific body schema, making them perceivable (and potentially available) to robots with any type of body schema (morphology).

We are already working on an automatic method for generating 3D object-descriptors. This would allow us to remove human bias from the way in which the robot observes and analyses its environment. By using an auto-encoder (a type of artificial neural network) that trains on appropriate datasets, it can automatically adapt to changes in objects that the robot interacts with.

Work is also underway on representing robot actions in a continuous space (e.g., using a vector representation of torque forces, or Dynamic Movement Primitives), which would be an improvement from today's discrete representation of actions (e.g., move, push, pull).

Ultimately, we intend to define an *algebra of affordances* detailing all the operations that are possible on affordances, and which would encompass operators such as affordance equivalence, affordance chaining (Ugur et al., 2011), and other operators that are still to explore.

AUTHOR CONTRIBUTIONS

Literature review by MA and RC-G. Methodology and theoretical developments by MA, RC-G, and RC. Experiment design and implementation by MA and RC-G. Analysis of the experimental results by MA, RC-G, RC, AG, and LG. Document writing and illustrations by MA, RC-G, RC, AG, and LG.

ACKNOWLEDGMENTS

This work was funded by ANR RoboErgoSum project under reference ANR-12-CORD-0030, and by Laboratory of Excellence SMART (ANR-11-LABX-65) supported by French State funds managed by the ANR - Investissements d'Avenir programme ANR-11-IDEX-0004-02.

We kindly thank Hugo Simão for his help with the 3D renderings used for illustrating this work. Credit for the 3D models of the Baxter robot used in **Figures 1**, **2**, **6** goes to Rethink Robotics. Credit for the robot model used in the bottom part of **Figure 2** goes to Dushyant Chourasia (https://grabcad.com/library/robot-242).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2018.00026/full#supplementary-material

REFERENCES

Boularias, A., Bagnell, J. A., and Stentz, A. (2015). "Learning to manipulate unknown objects in clutter by reinforcement," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence Learning* (Austin, TX), 1336–1342.

Chavez-Garcia, R. O., Andries, M., Luce-Vayrac, P., and Chatila, R. (2016a). "Discovering and manipulating affordances," in *International Symposium on Experimental Robotics (ISER)* (Tokyo).

Chavez-Garcia, R. O., Luce-Vayrac, P., and Chatila, R. (2016b). "Discovering affordances through perception and manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Daejeon).

Ciocodeica, S. (2016). *A Machine Learning Approach for Affordance Detection of Tools in 3D Visual Data*. Bachelor's thesis, University of Aberdeen.

Gibson, J. (1977). "The theory of affordances," in *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, eds R. Shaw and J. Bransford (Hoboken, NJ: John Wiley & Sons Inc.), 67–82.

Griffith, S., Sinapov, J., Sukhoy, V., and Stoytchev, A. (2012). "A behavior-grounded approach to forming object categories: separating containers from noncontainers," in *IEEE Transactions on Autonomous Mental Development*, 54–69.

Harnad, S. (1990). The symbol grounding problem. *Phys. D Nonlinear Phenomena* 42, 335–346.

Hermans, T., Li, F., Rehg, J. M., and Bobick, A. F. (2013). "Learning contact locations for pushing and orienting unknown objects," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (Atlanta, GA), 435–442.

Hermans, T. R. (2014). *Representing and Learning Affordance-Based Behaviors.* Ph.D. thesis, Georgia Institute of Technology.

Jain, R., and Inamura, T. (2013). Bayesian learning of tool affordances based on generalization of functional feature to estimate effects of unseen tools. *Artif. Life Robot.* 18, 95–103. doi: 10.1007/s10015-013-0105-1

Jamone, L., Ugur, E., Cangelosi, A., Fadiga, L., Bernardino, A., Piater, J., et al., (2016). "Affordances in psychology, neuroscience and robotics: a survey," in *IEEE Transactions on Cognitive and Developmental Systems.*

Katz, D., Venkatraman, A., Kazemi, M., Bagnell, J. A., and Stentz, A. (2014). Perceiving, learning, and exploiting object affordances for autonomous pile manipulation. *Auton. Robots* 37, 369–382. doi: 10.1007/s10514-014-9407-y

Kopicki, M., Zurek, S., Stolkin, R., Moerwald, T., and Wyatt, J. L. (2017). Learning modular and transferable forward models of the motions of push manipulated objects. *Auton. Robots* 41, 1061–1082. doi: 10.1007/s10514-016-9571-3

Krüger, N., Geib, C., Piater, J., Petrick, R., Steedman, M., Wörgötter, F., et al. (2011). Object-action complexes: grounded abstractions of sensory motor processes. *Robot. Auton. Syst.* 59, 740–757. doi: 10.1016/j.robot.2011.05.009

Min, H., Yi, C., Luo, R., Zhu, J., and Bi, S. (2016). "Affordance research in developmental robotics: a survey," in *IEEE Transactions on Cognitive and Developmental Systems*, 237–255.

Moldovan, B. (2015). *Relational Affordances and Their Applications*. Ph.D. thesis, KU Leuven.

Moldovan, B., Moreno, P., van Otterlo, M., Santos-Victor, J., and Raedt, L. D. (2012). "Learning relational affordance models for robots in multi-object manipulation tasks," in *2012 IEEE International Conference on Robotics and Automation* (Saint Paul, MN), 4373–4378.

Montesano, L., Lopes, M., Bernardino, A., and Santos-Victor, J. (2008). Learning object affordances: from sensory - Motor coordination to imitation. *IEEE Trans. Robot.* 24, 15–26. doi: 10.1109/TRO.2007.914848

Myers, A., Teo, C. L., Fermüller, C., and Aloimonos, Y. (2015). "Affordance detection of tool parts from geometric features," in *IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA), 1374–1381.

Nguyen, A., Kanoulas, D., Caldwell, D. G., and Tsagarakis, N. G. (2016). "Detecting object affordances with convolutional neural networks," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on IEEE* (Daejeon), 2765–2770.

Papon, J., Abramov, A., Schoeler, M., and Wörgötter, F. (2013). "Voxel cloud connectivity segmentation - Supervoxels for point clouds," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Portland, OR), 2027–2034.

Ridge, B., Skočaj, D., and Leonardis, A. (2009). *Unsupervised Learning of Basic Object Affordances from Object Properties.* PRIP, Vienna University of Technology.

Şahin, E., Çakmak, M., Doğar, M. R., Uğur, E., and Üçoluk, G. (2007). To afford or not to afford: a new formalization of affordances toward affordance-based robot control. *Adapt. Behav.* 15, 447–472. doi: 10.1177/1059712307084689

Srikantha, A., and Gall, J. (2016). Weakly supervised learning of affordances. *arXiv:1605.02964.*

Stoytchev, A. (2005). "Toward learning the binding affordances of objects: a behavior-grounded approach," in *Proceedings of AAAI Symposium on Developmental Robotics* (Palo Alto, CA), 17–22.

Suzuki, J. (2017). A theoretical analysis of the bdeu scores in bayesian network structure learning. *Behaviormetrika* 44, 97–116. doi: 10.1007/s41237-016-0006-4

Ugur, E., Şahin, E., and Oztop, E. (2012). "Self-discovery of motor primitives and learning grasp affordances," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura), 3260–3267.

Ugur, E., Şahin, E., and Oztop, E. (2011). "Unsupervised learning of object affordances for planning in a mobile manipulation platform," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on IEEE* (Shanghai), 4312–4317.

Varadarajan, K. M. and Vincze, M. (2012). "Afrob: the affordance network ontology for robots," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vilamoura), 1343–1350.

Zech, P., Haller, S., Lakani, S. R., Ridge, B., Ugur, E., and Piater, J. (2017). Computational models of affordance in robotics: a taxonomy and systematic classification. *Adapt. Behav.* 25, 235–271. doi: 10.1177/1059712317726357

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 26 June 2018
doi: 10.3389/fnbot.2018.00025

SERKET: An Architecture for Connecting Stochastic Models to Realize a Large-Scale Cognitive Model

Tomoaki Nakamura[1]*, *Takayuki Nagai*[1] *and Tadahiro Taniguchi*[2]

[1] *Department of Mechanical Engineering and Intelligent Systems, University of Electro-Communications, Tokyo, Japan,*
[2] *Department of Information Science and Engineering, Ritsumeikan University, Shiga, Japan*

To realize human-like robot intelligence, a large-scale cognitive architecture is required for robots to understand their environment through a variety of sensors with which they are equipped. In this paper, we propose a novel framework named Serket that enables the construction of a large-scale generative model and its inferences easily by connecting sub-modules to allow the robots to acquire various capabilities through interaction with their environment and others. We consider that large-scale cognitive models can be constructed by connecting smaller fundamental models hierarchically while maintaining their programmatic independence. Moreover, the connected modules are dependent on each other and their parameters must be optimized as a whole. Conventionally, the equations for parameter estimation have to be derived and implemented depending on the models. However, it has become harder to derive and implement equations of large-scale models. Thus, in this paper, we propose a parameter estimation method that communicates the minimum parameters between various modules while maintaining their programmatic independence. Therefore, Serket makes it easy to construct large-scale models and estimate their parameters via the connection of modules. Experimental results demonstrated that the model can be constructed by connecting modules, the parameters can be optimized as a whole, and they are comparable with the original models that we have proposed.

Keywords: cognitive models, probabilistic generative models, symbol emergence in robotics, concept formation, unsupervised learning

OPEN ACCESS

Edited by:
Quan Zou,
UnitedHealth Group, United States

Reviewed by:
Eric Chen,
Thomas Jefferson University,
United States
Yanan Sun,
Booz Allen Hamilton, United States

****Correspondence:***
Tomoaki Nakamura
tnakmaura@uec.ac.jp

Received: *30 November 2017*
Accepted: *14 May 2018*
Published: *26 June 2018*

Citation:
Nakamura T, Nagai T and Taniguchi T (2018) SERKET: An Architecture for Connecting Stochastic Models to Realize a Large-Scale Cognitive Model. Front. Neurorobot. 12:25. doi: 10.3389/fnbot.2018.00025

1. INTRODUCTION

To realize human-like robot intelligence, a large-scale cognitive architecture is required for robots to understand their environment through a variety of sensors with which they are equipped. In this paper, we propose a novel framework that enables the construction of a large-scale generative model and its inferences easily by connecting sub-modules in order for robots to acquire various capabilities through interactions with their environment and others. We consider it important for robots to understand the real world by learning from their environment and others, and have proposed a method that enables robots to acquire concepts and language (Nakamura et al., 2014; Attamimi et al., 2016; Nishihara et al., 2017; Taniguchi et al., 2017) based on the clustering of multimodal information that they obtain. These proposed models are based on Bayesian models with complex structures, and we derived and implemented the parameter estimation equations. If we realize a model that enables robots to learn more complicated capabilities, we have to construct a more complicated model, and derive and implement equations for parameter estimation. However, it is difficult to construct higher-level cognitive models by leveraging this approach. Alternatively, these models can be interpreted as a composition of more fundamental Bayesian models. In this paper, we develop a large-scale cognitive model by connecting the Bayesian models and propose an architecture named Serket (Symbol Emergence in Robotics tool KIT[1]), which enables the easier construction of such models.

In the field of cognitive science, cognitive architectures (Laird, 2008; Anderson, 2009) have been proposed to implement human cognitive mechanisms by describing human perception, judgment, and decision-making. However, complex machine learning algorithms have not yet been introduced, which makes it difficult to implement our proposed models. Serket makes it possible to implement more complex models by connecting modules.

One approach to develop a large-scale cognitive model is the use of probabilistic programming languages (PPLs), which make it easy to construct Bayesian models (Patil et al., 2010; Goodman et al., 2012; Wood et al., 2014; Carpenter et al., 2016; Tran et al., 2016). PPLs can construct Bayesian models by defining the dependencies between random variables, and the parameters are automatically estimated without having to derive the equations for them. By using PPLs, it is easy to construct relatively small-scale models, such as a Gaussian mixture model and latent Dirichlet allocation, but it is still difficult to model multimodal sensory information, such as images and speech obtained by the robots. Because of this, we implemented models for concept and language acquisition, which are relatively large-scale models, as standalone models without PPLs. However, we consider the approach where an entire model is implemented by itself has limitations if it is constructed as a large-scale model.

Large-scale cognitive models can be constructed by connecting smaller fundamental models hierarchically; in fact, our proposed models have such a structure. In the proposed novel architecture Serket, large-scale models were constructed by hierarchically connecting smaller-scale Bayesian models (hereafter, each one is referred to as a *module*) while maintaining their programmatic independence. The connected modules are dependent on each other, and parameters must be optimized as a whole. When models are constructed by themselves, the parameter estimation equations have to be derived and implemented depending on the models. However, in this paper, we propose a method for parameter estimation by communicating the minimum parameters between various modules while maintaining their programmatic independence. Therefore, Serket makes it easy to construct large-scale models and estimate their parameters by connecting modules.

In this paper, we propose the Serket framework and implement models that we proposed by leveraging this framework. Experimental results demonstrated that the model can be constructed by connecting modules, the parameters can be optimized as a whole, and they are comparable with original models that we have proposed.

2. BACKGROUND

2.1. Symbol Emergence in Robotics

Recently, it has been said that artificial intelligence is superior to human intelligence in the area of supervised learning, as typified by deep learning as far as certain specific tasks (He et al., 2015; Silver et al., 2017). However, we believe that it is difficult to realize human-like intelligence only via supervised learning because all supervised labels cannot be obtained for all the sensory information of robots. To this end, we believe that it is also important for robots to understand the real environment by structuring their own sensory information in an unsupervised manner. We consider such a learning process as a symbol emergence system (Taniguchi et al., 2016a).

The symbol emergence system is based on the genetic epistemology proposed by Piaget (Piaget and Duckworth, 1970). In genetic epistemology, humans organize symbol systems in a bottom-up manner through interaction with the environment. **Figure 1** presents an overview of the symbol emergence system. The symbols are self-organized from sensory information obtained through interactions with the environment. However, it can be difficult for robots to communicate with others using symbols learned only in a bottom-up manner, because the sensory information cannot be shared directly with others and the meaning of symbols differs depending on the individual. To communicate with others, the meanings of symbols must be transformed into common meanings among individuals through their interactions. This is considered as a top-down effect from symbols to individuals' organization of them. Thus, in the symbol emergence system, the symbols emerge through loops of top-down and bottom-up effects. In the symbol emergence in robotics, symbols include not only linguistic symbols but also various types of knowledge self-organized by robots. Therefore, symbol emergence in robotics covers a wide range of research topics, such as concept formation (Nakamura et al., 2007), language acquisition (Taniguchi et al., 2016b, 2017; Nishihara et al., 2017), learning of interactions (Taniguchi et al., 2010), learning of body schemes (Mimura et al., 2017), and learning of motor skills and segmentation of time-series data (Taniguchi et al., 2011; Nakamura et al., 2016).

We have proposed models that enable robots to acquire concepts and language by considering its learning process as a symbol emergence system. The robots form concepts in a bottom-up manner, and acquire word meanings by connecting words and concepts. Simultaneously, words are shared with others, and their meanings are changed through communication with others. Therefore, such words affect concept formation in a top-down manner, and concepts are changed. Thus, we have

[1] Symbol emergence in robotics focuses on the real and noisy environment, and the *e* in Serk*e*t represents a false recognition obtained through learning in such an environment.

considered that robots can acquire concepts and word meanings through loops of bottom-up and top-down effects.

2.2. Existing Cognitive Architecture

There have been many attempts to develop intelligent systems. In the field of cognitive science, cognitive architectures (Laird, 2008; Anderson, 2009) have been proposed to implement humans cognitive mechanisms by describing human perception, judgment, and decision-making. As mentioned earlier, it is important to consider how to model the multimodal sensory information obtained by robots. However, this is still difficult to achieve with these cognitive architectures. To construct more complex models, some frameworks have been proposed in the field of machine learning.

Frameworks of deep neural networks (DNNs) such as TensorFlow (Abadi et al., 2016), Keras (Chollet , 2015), and Chainer (Tokui et al., 2015) have been developed. These frameworks make it possible to construct DNN models and estimate their parameters easily. These frameworks are one of the reasons why DNNs have been widely used for several years.

Alternatively, PPLs that make it easy to construct Bayesian models have also been proposed (Patil et al., 2010; Goodman et al., 2012; Wood et al., 2014; Carpenter et al., 2016; Tran et al., 2016). The advantages of PPLs are that they can construct Bayesian models by defining the dependencies between random variables, and the parameters are automatically estimated without deriving equations for them. By using PPLs, relatively small-scale models, such as the Gaussian mixture model and latent Dirichlet allocation (LDA), can be constructed easily. However, it is still difficult to model multimodal sensory information, such as images and speech obtained by the robots. We believe that a framework by which a large-scale probabilistic generative model can be more easily constructed is required to model the multimodal information of the robot.

2.3. Cognitive Architecture Based on Probabilistic Generative Model

We believe that cognitive models make it possible to predict an output Y against an input X. For example, as shown in **Figure 2**, an object label Y is predicted from a sensor input X via object recognition. It is through the understanding of word meanings that the semantic content Y are predicted from speech signal X. In other words, the problem can be defined as how to model $P(Y|X)$, where the prediction is realized by

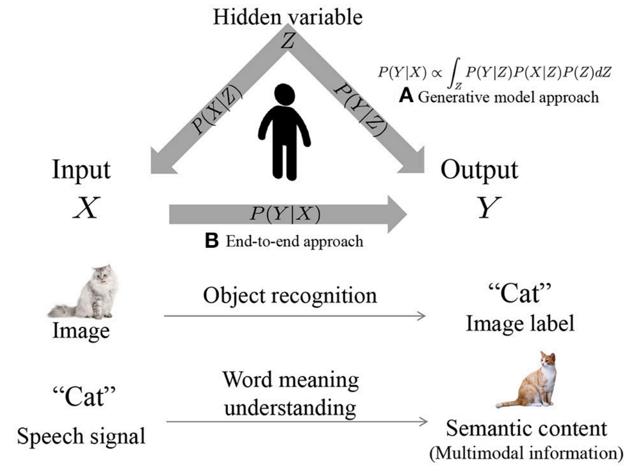


FIGURE 2 | Overview of cognitive model by **(A)** probabilistic generative model and **(B)** end-to-end learning.

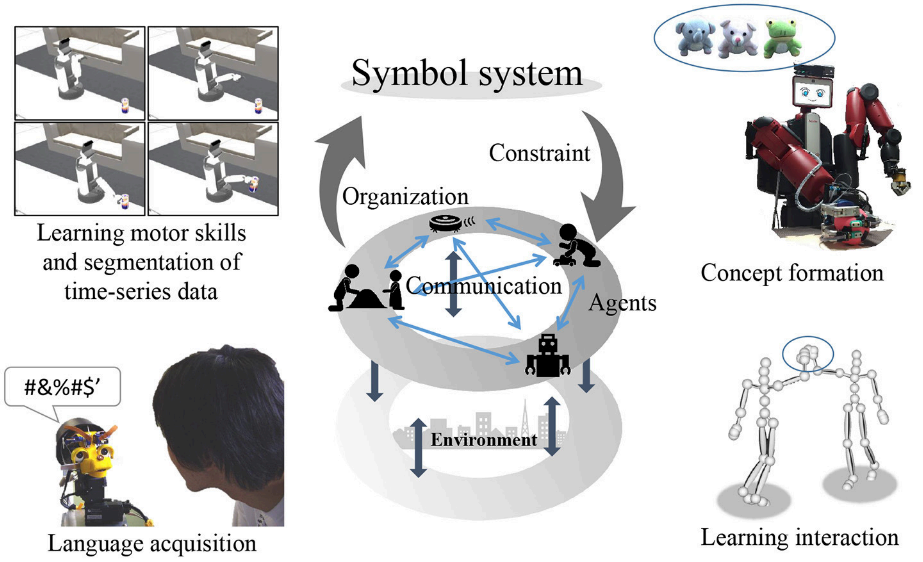


FIGURE 1 | Symbol emergence system.

$\text{argmax}_Y P(Y|X)$. DNNs model relationships between an input X and output Y directly by an end-to-end approach (**Figure 2B**). Alternatively, we considered developing these cognitive models by leveraging Bayesian models, where X and Y are treated as random variables, and the relationships between them are represented by a latent variable Z (**Figure 2A**). Therefore, in Bayesian models, the prediction of output Y from input X is computed as follows:

$$P(Y|X) \propto P(Y,X) \tag{1}$$

$$= \int_Z P(Y|Z)P(X|Z)P(Z)dZ. \tag{2}$$

This is multimodal latent Dirichlet allocation (MLDA) (Blei and Jordan, 2003; Nakamura et al., 2009; Putthividhy et al., 2010), the details of which are described in the Appendix. However, MLDA is based on the important assumption that the observed variables X and Y are conditionally independent against latent variable Z. Here, we consider models where assumptions are made about multiple observations without distinguishing between input and output. **Figure 3A** displays the generalized model, where the right side of Equation (1) corresponds to the following equation, and a part of the observations can be predicted from other observations.

$$P(\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots) = \int_z P(z)\Pi_n P(\boldsymbol{o}_n|z)dz. \tag{3}$$

As mentioned earlier, it is assumed that all observations $\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots$ are conditionally independent against z. This assumption is often used to deal with multimodal data (Blei and Jordan, 2003; Wang et al., 2009; Putthividhy et al., 2010; Françoise et al., 2013) because modeling all dependencies makes parameter estimation difficult.

Considering the modeling of various sensor data as observations $\boldsymbol{o}_1, \boldsymbol{o}_2, \cdots$, it is not always true for all the observations to satisfy the conditionally independent assumption. In general, the information surrounding us has a hierarchical structure. Hence, a hierarchical model can be used to avoid this difficulty (Attamimi et al., 2016). Furthermore, latent variables, such as concepts, are generally related to each other, and such relationships can be represented by hierarchical models. **Figure 3B** represents a hierarchical version of **Figure 3A** and can be thought of as generalization of the cognitive architecture based on a probabilistic generative model. It should be noted that the structure can be designed manually (Attamimi et al., 2016) and/or found autonomously by using a structure learning method (Margaritis, 2003), which is beyond the scope

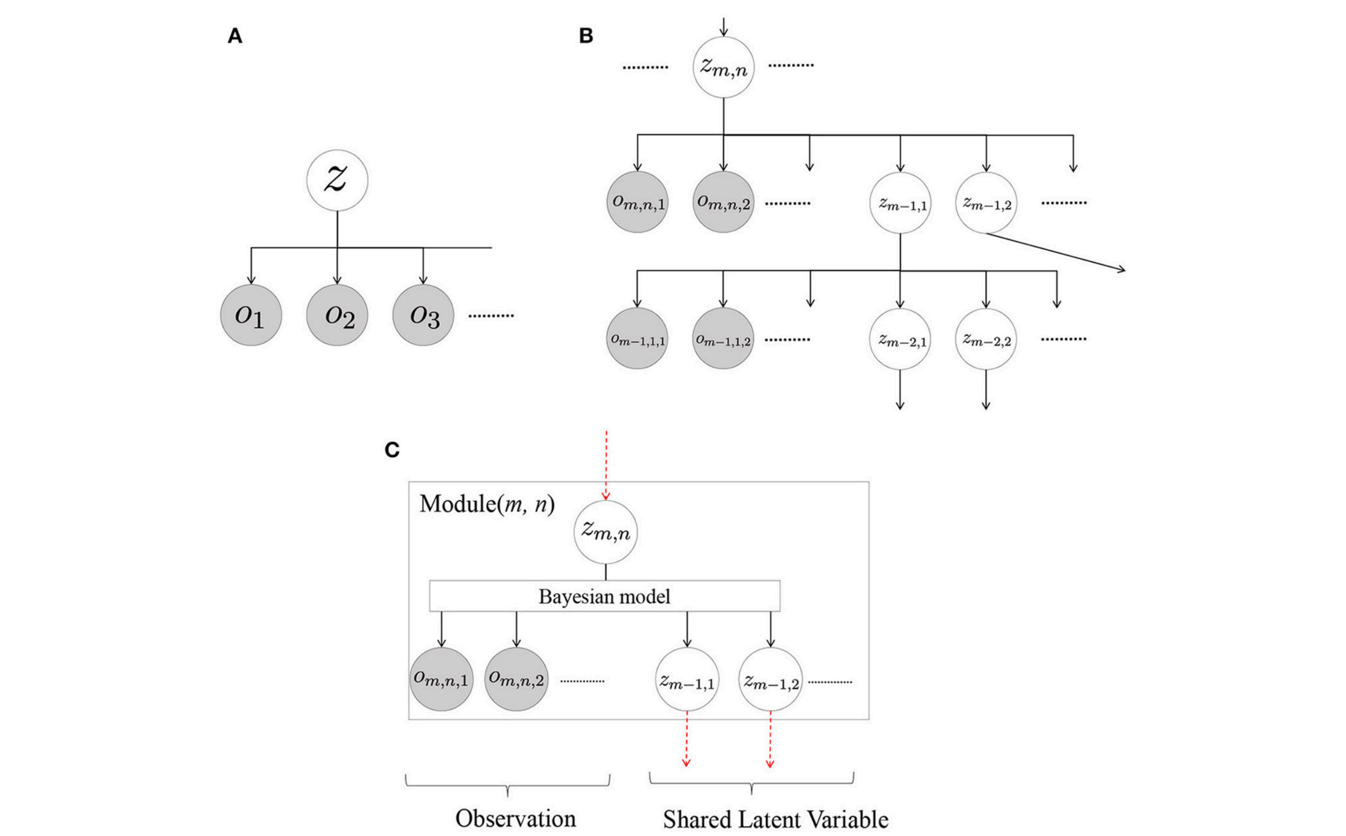


FIGURE 3 | Generalized hierarchical cognitive model: **(A)** single-layer model, **(B)** multilayered model by hierarchicalization of single-layer models, and **(C)** generalized form of a module in Serket.

of this paper. In this hierarchized model, $o_{*,*}$ are observations and $z_{*,*}$ are latent variables, and the right side of Equation (1) corresponds to the following equation:

$$P(\boldsymbol{O}|z_{M,1}, z_{M,2}, \cdots) = \prod_{m}^{M} \prod_{n}^{\bar{N}_m} \int_{z_{m,n}} P(z_{m,n}) \prod_{i}^{N_m} P(\boldsymbol{o}_{m,n,i}|z_{m,n}) \prod_{n'}^{\bar{N}_{m-1}} P(z_{m-1,n'}|z_{m,n}) dz_{m,n}, \tag{4}$$

where $\boldsymbol{O}$ is the set of all observations, M is the number of the hierarchy, and N_m and $\bar{N}_m$ denote the number of observations and latent variables in the m-th hierarchy, respectively. In this model, it is not difficult to analytically derive equations to estimate the parameters if the number of the hierarchy is not large. However, it is more difficult to derive them if the number of the hierarchy increases. To estimate the parameters of the hierarchical model, we propose Serket, which is an architecture that renders it possible to estimate the parameters by dividing them into even hierarchies.

From the viewpoint of hierarchical models, many studies have proposed models that capture the hierarchical nature of the data (Li and McCallum, 2006; Blei et al., 2010; Ghahramani et al., 2010; Ando et al., 2013; Nguyen et al., 2014). On the other hand, Serket models the hierarchical structure of modalities. For such hierarchical models, methods based on LDA (Li et al., 2011; Yang et al., 2014) have been proposed, and we have also proposed multilayered MLDA (Attamimi et al., 2016). These models are the simplest examples constructed by Serket. In this paper, we construct these models by dividing them into smaller modules.

2.4. Cognitive Models

In the past, studies on how the relationships between multimodal information are modeled have been conducted (Roy and Pentland, 2002; Wermter et al., 2004; Ridge et al., 2010; Ogata et al., 2010; Lallee and Dominey, 2013; Zhang et al., 2017). Neural networks were used in these studies, which made inferences based on observed information possible by learning multimodal information, such as words, visual information, and a robot's motions. As mentioned earlier, these are some examples of the cognitive models that we defined.

There are also studies in which manifold learning was used for modeling a robot's multimodal information (Mangin and Oudeyer, 2013; Yuruten et al., 2013; Mangin et al., 2015; Chen and Filliat, 2015). These studies used manifold learning such as non-negative matrix factorization, in which multimodal information is represented by low-dimensional hidden parameters. We consider this as another approach to constructing cognitive models, in which the information is inferred through hidden parameters.

Recently, DNNs have made notable advances in many areas such as object recognition (He et al., 2015), object detection (Redmon et al., 2016), speech recognition (Amodei et al., 2016), sentence generation (Vinyals et al., 2015), machine translation (Sutskever et al., 2014), and visual question answering (Wu et al., 2016). In these studies, end-to-end learning was used, which made it possible to infer information from other information. Therefore, these are also considered part of the cognitive model defined in this paper. However, as mentioned in section 2.1, we believe that it is important for robots to understand the real environment by structuring their own sensory information in an unsupervised manner.

To develop a cognitive model where robots learn autonomously, our group proposed several models for concept formation (Nakamura et al., 2007), language acquisition (Taniguchi et al., 2016b, 2017; Nishihara et al., 2017), learning of interactions (Taniguchi et al., 2010), learning of body schemes (Mimura et al., 2017), learning motor skills, and segmentation of time series data (Taniguchi et al., 2011; Nakamura et al., 2016). Although all of these are targets of Serket, we focused on concept formation in this paper. We define concepts as categories into which the sensory information is classified, and propose various concept models. These are implementations of the aforementioned hierarchical model. **Figure 4A** displays one of our proposed models. This is the simplest form of the hierarchical model, where z^O and z^M denote an object and a motion concept, respectively, and their relationship is represented by z (Attamimi et al., 2016). Therefore, in this model, z represents objects and possible motions against them, which are considered as their usage, and observations become conditionally independent by introducing the latent variables z^O and z^M.

In these Bayesian models, the latent variables shown as the white nodes z, z^O, and z^M in **Figure 4A** can be learned from the observations shown as gray nodes in an unsupervised manner. Moreover, these latent variables are not determined independently but optimized as a whole by depending on each other. Although it seems that this model has a complex structure and that it is difficult to estimate the parameters and determine the latent variables, this model can be divided into smaller components, each of which is an MLDA model. The models shown in **Figures 4B,C** can also be divided into smaller components despite their complex structure. Similar to these models, it is possible to develop larger models by combining smaller models as modules. In this paper, we propose a novel architecture Serket to develop larger models by combining modules.

In the proposed architecture, the parameters of each module are not learned independently but learned based on their dependence on each other. To implement such learning, it is important to share latent variables between modules. For example, z^O and z^M are shared between two MLDAs in the model, respectively, as shown in **Figure 4A**. The shared latent variables were not determined independently but determined depending on each other. Serket makes it possible for each module to maintain its independence as a program as well as be learned as a whole through the shared latent variables.

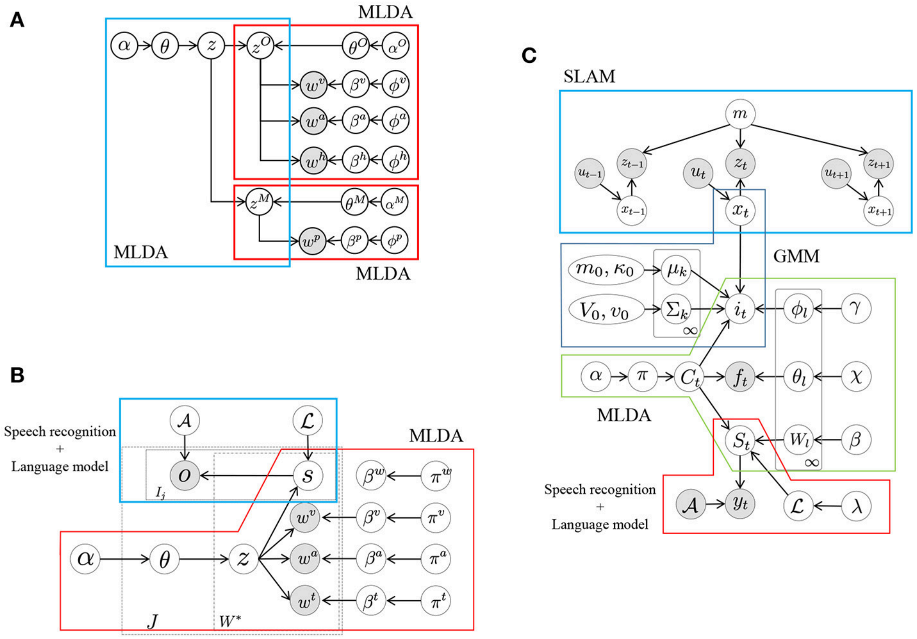


FIGURE 4 | Graphical models for concept formation: **(A)** model for hierarchical concept (Attamimi et al., 2016) constructed with multimodal latent Dirichlet allocations (MLDAs), **(B)** model for object concept and language acquisition (Nakamura et al., 2014; Nishihara et al., 2017) constructed with MLDAs and speech recognition, and **(C)** model for location concept and language acquisition (Taniguchi et al., 2017) constructed with simultaneous localization and mapping (SLAM), Gaussian mixture model (GMM), MLDA, and speech recognition.

3. SERKET

3.1. Composing Cognitive Sub-modules

Figure 3C displays the generalized form of the module assumed in Serket. In this figure, we omit the detailed parameters for generalization because we assume that any type of models can be the modules of Serket. Each module has multiple shared latent variables $z_{m-1,*}$ and observations $\boldsymbol{o}_{m,n,*}$, which are assumed to be generated from latent variable $z_{m,n}$ of a higher level. Modules with no shared latent variable or observations are also included in the generalized model. Moreover, the modules can have any internal structure as long as they have shared latent, observation, and higher-level latent variables. Based on this module, a larger model can be constructed by connecting the latent variables of module$(m-1,1)$, module$(m-1,2)$, $\cdots$ recursively. In the Serket architecture, each module must satisfy the following requirements:

1. In each module with shared latent variables, the probability that latent variables are generated can be computed as

$$P(z_{m-1,i}|z_{m,n},\boldsymbol{o}_{m,n,1},\boldsymbol{o}_{m,n,2},\cdots,\boldsymbol{z}_{m-1}). \qquad (5)$$

2. The module can send the following probability by leveraging one of the methods explained in the next section:

$$P(z_{m-1,i}|z_{m,n},\boldsymbol{o}_{m,n,1},\boldsymbol{o}_{m,n,2},\cdots,\boldsymbol{z}_{m-1}). \qquad (6)$$

3. The module can determine $z_{m,n}$ by using the following probability sent from module $(m+1,j)$ by one of the methods explained in the next section:

$$P(z_{m,n}|z_{m+1,j},\boldsymbol{o}_{m+1,j,1},\boldsymbol{o}_{m+1,j,2},\cdots,\boldsymbol{z}_{m}). \qquad (7)$$

4. Terminal modules have no shared latent variables and only have observations.

In Serket, the modules affecting each other and the shared latent variables are determined by their communication with each other. Methods to determine the latent variables are classified into two types depending on their nature. One is the case that they are discrete and finite, and another is the case that they are continuous or infinite.

3.2. Inference of Composed Models

In this section, we explain the parameter inference methods used for the composed models. We focus on the batch algorithm for parameter inference, which makes it easy to implement each module. Therefore, real-time application is beyond the scope of this paper although we would like to realize it in the future. One of the inference methods used to estimate the parameters of complex models is based on variational Bayesian (VB) approximation (Minka and Lafferty, 2002; Blei et al., 2003; Kim et al., 2013). However, a VB-based approach requires derivation against latent variables, and it is difficult to implement derivation in independent modules. To this end, we employed a sampling-based method because of its simpler implementation.

In this section, we utilize three approaches according to the nature of the latent variables.

3.2.1. Message Passing Approach

First, we consider the case when the latent variables are discrete and finite. For example, in the model shown in **Figure 4A**, the shared latent variable z^O was generated from a multinomial distribution, which is represented by finite dimensional parameters. Here, we consider the estimation of the latent variables according to the simplified model shown in **Figure 5A**. In module 2, the shared latent variable z_1 was generated from z_2; and in module 1, the observation o was generated from z_1. The latent variable z_1 is shared in modules 1 and 2, and determined by the effect on these two modules as follows:

$$z_1 \sim P(z_1|\boldsymbol{o}, z_2) \tag{8}$$

$$\propto P(z_1|\boldsymbol{o})P(z_1|z_2). \tag{9}$$

In this equation, $P(\boldsymbol{o}|z_1)$ and and $P(z_1|z_2)$ can be computed in modules 1 and 2, respectively. We assumed that the latent variable is discrete and finite, and $P(z_1|z_2)$ is a multinomial distribution that can be represented by a finite-dimensional parameter whose dimension ranges from the number of elements of z_1. Therefore, $P(z_1|z_2)$ can be sent from module 2 to module 1. Moreover, $P(z_1|z_2)$ can be learned in module 2 by using $P(z_1|\boldsymbol{o})$ sent from module 1, which is also a multinomial distribution. The parameters of these distributions can be easily sent and received, and the shared latent variable can be determined by the following procedure:

1. In module 1, $P(z_1|\boldsymbol{o})$ is computed.
2. $P(z_1|\boldsymbol{o})$ is sent to module 2.
3. In module 2, the probability distribution $P(z_1|z_2)$, which represents the relationships between z_1 and z_2, is estimated using $P(z_1|\boldsymbol{o})$.
4. $P(z_1|z_2)$ is sent to module 1.
5. In module 1, the latent variable z_1 is estimated using Equation (9), and the parameters of $P(\boldsymbol{o}|z_1)$ are updated.

Thus, in the case when the latent variable is infinite and discrete, the modules are learned by sending and receiving the parameters

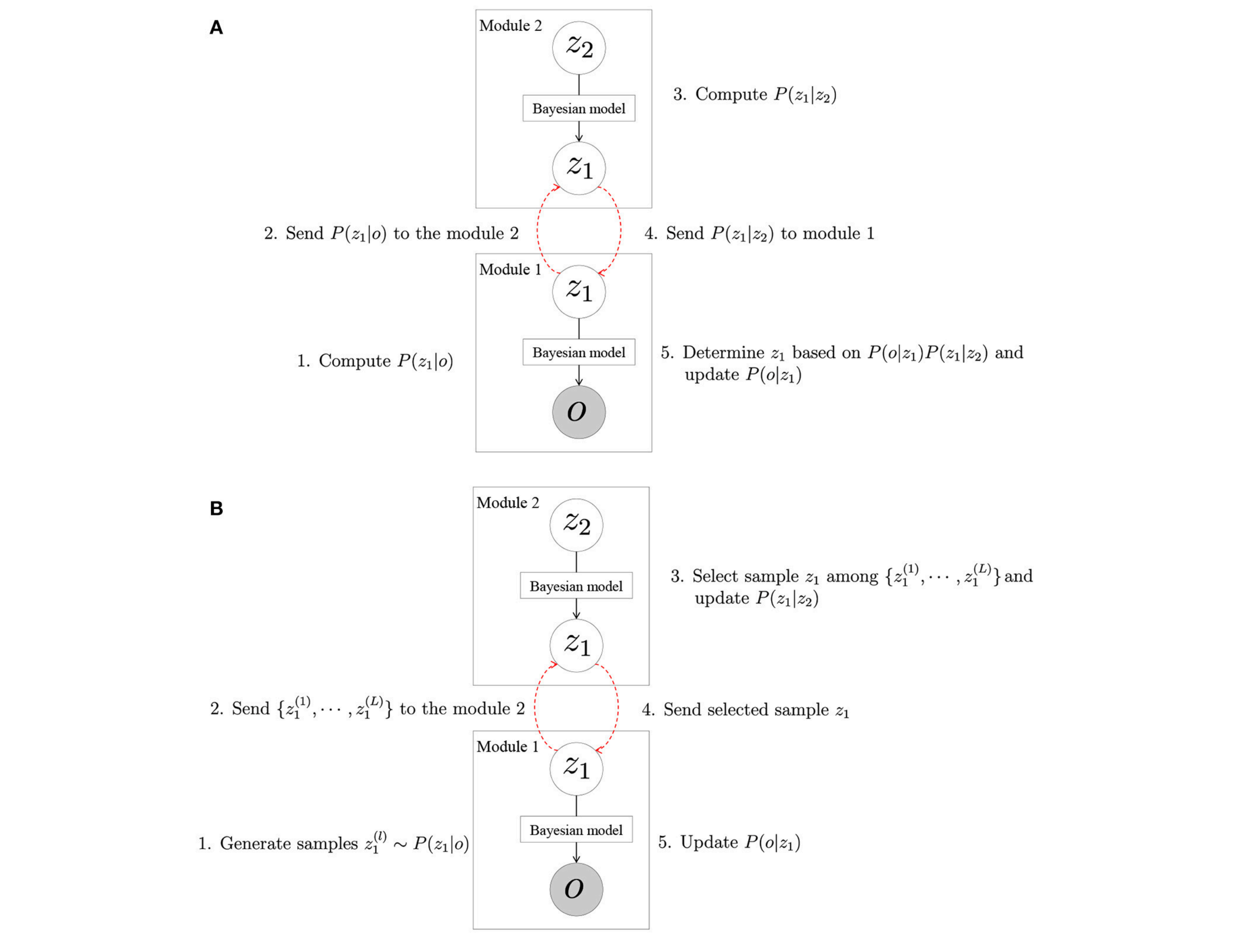

A
Module 2
z_2
Bayesian model
z_1
3. Compute $P(z_1|z_2)$
2. Send $P(z_1|o)$ to the module 2
4. Send $P(z_1|z_2)$ to module 1
Module 1
z_1
Bayesian model
o
1. Compute $P(z_1|o)$
5. Determine z_1 based on $P(o|z_1)P(z_1|z_2)$ and update $P(o|z_1)$
B
Module 2
z_2
Bayesian model
z_1
3. Select sample z_1 among $\{z_1^{(1)}, \cdots, z_1^{(L)}\}$ and update $P(z_1|z_2)$
2. Send $\{z_1^{(1)}, \cdots, z_1^{(L)}\}$ to the module 2
4. Send selected sample z_1
Module 1
z_1
Bayesian model
o
1. Generate samples $z_1^{(l)} \sim P(z_1|o)$
5. Update $P(o|z_1)$

FIGURE 5 | Connecting two modules by **(A)** MP approach and **(B)** SIR approach.

of a multinomial distribution of z_1. We call this the message passing (MP) approach because the model parameters can be optimized by communicating the message.

3.2.2. Sampling Importance Resampling Approach

In the previous section, the latent variable was determined by communicating the parameters of the multinomial distributions if the latent variables are discrete and finite. Otherwise, it can be difficult to communicate the parameters. For example, the number of parameters becomes infinite if the possible values of the latent variables are infinite patterns. In the case of a complex probability distribution, it is difficult to represent it by a small number of parameters. In such cases, the model parameters are learned by approximation using sampling importance resampling (SIR). We also consider parameter estimation using the simplified model shown in **Figure 5B**. Here, the latent variable z_1 is shared, and its possible value is either an infinite pattern or continuous. Similar to the previous section, the latent variable is determined if the following equation can be computed:

$$\begin{aligned} z_1 &\sim P(z_1|\boldsymbol{o}, z_2) && (10) \\ &\propto P(z_1|\boldsymbol{o})P(z_1|z_2). && (11) \end{aligned}$$

However, when the value of z_1 is infinite or continuous, module 2 cannot send $P(z_1|z_2)$ to module 1. Therefore, $P(z_1|\boldsymbol{o})$ is first approximated by L samples $\{z^{(l)} : l = 1, \cdots, L\}$:

$$z_1^{(l)} \sim P(z_1|\boldsymbol{o}). \quad (12)$$

This approximation is equivalent to approximating $P(z_1|o)$ by the following $\tilde{P}(z_1|\boldsymbol{o})$:

$$P(z_1|\boldsymbol{o}) \approx \tilde{P}(z_1|\boldsymbol{o}) = \frac{1}{L}\sum_{l}^{L} \delta(z_1, z_1^{(l)}), \quad (13)$$

where $\delta(a, b)$ represents a delta function, which is 1 if $a = b$, and 0 otherwise. The generated samples are sent from module 1 to module 2, and a latent variable is selected among them based on $P(z_1|z_2)$:

$$z_1 \sim P(z_1 \in \{z_1^{(1)}, \cdots, z_1^{(L)}\}|z_2). \quad (14)$$

This procedure is equivalent to sampling from the following distribution, which is an approximation of Equation (11):

$$z_1 \sim P(z_1|z_2)\tilde{P}(z_1|\boldsymbol{o}). \quad (15)$$

Thus, the parameters of each module can be updated by the determined latent variables.

3.2.3. Other Approaches

We have presented two methods but these are not the only ones available for parameter estimation. There are other applicable methods to estimate parameters. For example, one of the applicable methods is the Metropolis-Hastings (MH) approach. In the MH approach, samples are generated from a proposal distribution $Q(z|z^*)$, where z^* and z represent the current value and generated value of latent variables, respectively. Then, they are accepted according to the acceptance probability $A(z, z^*)$:

$$\begin{aligned} A(z, z^*) &= \min(1, \alpha) && (16) \\ \alpha &= \frac{P(z^*)Q(z|z^*)}{P(z)Q(z^*|z)}, && (17) \end{aligned}$$

where $P(z)$ represents the target distribution from which the samples are generated.

The model parameters in **Figure 5** can be estimated by considering $P(z_1|\boldsymbol{o})$ and $P(z_1|z_2, \boldsymbol{o})$ as the proposal distribution and target distribution, respectively. $P(z_1|z_2, \boldsymbol{o})$ can be transformed into

$$P(z_1|z_2, \boldsymbol{o}) \propto P(z_1|\boldsymbol{o})P(z_1|z_2)P(z_2). \quad (18)$$

Therefore, α in Equation (16) becomes

$$\begin{aligned} \alpha &= \frac{P(z^*)Q(z|z^*)}{P(z)Q(z^*|z)} = \frac{P(z_1^*|z_2, \boldsymbol{o})}{P(z_1|z_2, \boldsymbol{o})} \cdot \frac{P(z_1|\boldsymbol{o})}{P(z_1^*|\boldsymbol{o})} && (19) \\ &= \frac{P(z_1^*|\boldsymbol{o})P(z_1^*|z_2)P(z_2)}{P(z_1|\boldsymbol{o})P(z_1|z_2)P(z_2)} \cdot \frac{P(z_1|\boldsymbol{o})}{P(z_1^*|\boldsymbol{o})} = \frac{P(z_1^*|z_2)}{P(z_1|z_2)}, && (20) \end{aligned}$$

Hence, the proposal distribution $P(z_1|\boldsymbol{o})$ can be computed in module 1, and the acceptance distribution can be computed in module 2. By using this approach, the parameters can be estimated while maintaining programmatic independence. The proposed value is sent to module 2, and module 2 determines whether it is accepted or not. Then, the parameters are updated according to the accepted values.

Thus, various approaches can be utilized for parameter estimation, and it should be discussed which methods are most suitable. However, we will leave this for a future discussion because of limited space.

4. EXAMPLE 1: MULTILAYERED MLDA

First, we show that a more complex model, mMLDA, can be constructed by combining the simpler models based on Serket. By using the mMLDA, the object categories, motion categories, and integrated categories representing the relationships between them were formed from the visual, auditory, haptic, and motion information obtained by the robot. The information obtained by the robot is detailed in Appendix 2. We compared it with the original mMLDA and an independent model, where the object and motion categories were learned independently. The original mMLDA has an upper-bound performance because any approximation is not used in it. Therefore, the purpose of this experiment is to show that Serket implementation has a comparable performance with the original mMLDA.

4.1. Implementation Based on Serket

The mMLDA shown in **Figure 4A** can be constructed using the MP approach. This model can be divided into to three MLDAs. In the lower-level MLDAs, object categories z^O can be formed from multimodal information $\boldsymbol{w}^v$, $\boldsymbol{w}^a$, and $\boldsymbol{w}^h$ obtained from the objects, and motion categories z^M can be formed from joint angles obtained by observing a human's motion. Details of the information are explained in the Appendix. Moreover, in the higher-level MLDA, integrated categories z that represent the relationships between objects and motions can be formed by considering z^O and z^M as observations. In this model, latent variables z^O and z^M are shared; therefore, the whole model parameters are optimized in a mutually affecting manner. **Figure 6** shows the mMLDA represented by three MLDAs.

First, in the two MLDAs shown in **Figures 6A,B**, the probabilities $P(z_j^O|\boldsymbol{w}_j^v, \boldsymbol{w}_j^a, \boldsymbol{w}_j^h)$ and $P(z_j^M|\boldsymbol{w}_j^p)$ that the object and motion category of the multimodal information in the j-th data become z_j^O and z_j^M, respectively, can be computed using Gibbs sampling. These probabilities are represented by finite and discrete parameters, which can be sent to the integrated concept model shown in **Figure 6C**, where $\hat{z}_j^O$ and $\hat{z}_j^M$ can be treated as observed variables using these probabilities.

$$\hat{z}_{jn}^O \sim P(z_j^O|\boldsymbol{w}_j^v, \boldsymbol{w}_j^a, \boldsymbol{w}_j^h), \tag{21}$$

$$\hat{z}_{jn}^M \sim P(z_j^M|\boldsymbol{w}_j^p). \tag{22}$$

where $\boldsymbol{w}_j^v, \boldsymbol{w}_j^a, \boldsymbol{w}_j^h$, and $\boldsymbol{w}_j^p$ represent the visual information, auditory information, haptic information, and joint angles of the human's motion, respectively, which are included in the j-th data.

Thus, in the integrated concept model, category z can be formed in an unsupervised manner. Next, the values of the shared latent variables are inferred stochastically using a learned integrated concept model:

$$P(z^O|\hat{z}_j^M, \hat{z}_j^O) = \sum_z P(z^O|z)P(z|\hat{z}_j^m, \hat{z}_j^o), \tag{23}$$

$$P(z^M|\hat{z}_j^M, \hat{z}_j^O) = \sum_z P(z^M|z)P(z|\hat{z}_j^m, \hat{z}_j^o). \tag{24}$$

These probabilities are also represented by finite and discrete parameters, which can be communicated using the MP approach. These parameters are sent to an object concept model and motion concept model, respectively, where the latent variables assigned to the modality information $m \in \{v, a, h, p\}$ of concept $C \in \{O, M\}$ are determined using Gibbs sampling.

$$z_{jmn}^C \sim P(z^C|\boldsymbol{W}^m, \boldsymbol{Z}_{-jmn})P(z^C|\hat{\boldsymbol{z}}_j^M, \hat{\boldsymbol{z}}_j^O), \tag{25}$$

where $\boldsymbol{W}^m$ represents all the information of modality m, and $\boldsymbol{Z}_{-jmn}$ represents a set of latent variables, except for the latent variable assigned to the information of modality m of the j-th observation. Whereas the latent variables were sampled from $P(z^C|\boldsymbol{W}^m, \boldsymbol{Z}_{-jmn})$ in the normal MLDA, they were also sampled using $P(z^C|\hat{\boldsymbol{z}}_j^M, \hat{\boldsymbol{z}}_j^O)$. Therefore, all the latent variables were learned in a complementary manner. From the sampled variables, the parameters of $P(z_j^o|\boldsymbol{w}_j^v, \boldsymbol{w}_j^a, \boldsymbol{w}_j^h)$ and $P(z_j^m|\boldsymbol{w}_j^m)$ were updated, and Equations (21–25) were iterated until they converged.

Figure 7 shows the pseudocode of mMLDA and the corresponding graphical model. The model on the left in **Figure 7** can be constructed by connecting the latent variables based on Serket. Although the part framed by the red rectangle was implemented in the experiment, it can be easily extended to the model shown in this figure.

4.2. Experimental Results

Figure 8A shows a confusion matrix of classification by the model, where the object and motion categories were learned independently, and the vertical and horizontal axes represent the correct category index and the category index to which each object was classified, respectively. The accuracies were 98 and 72%. One can see that the motion categories can be formed by the independent model almost correctly. However, the object categories could not be formed correctly compared to the motion categories. On the other hand, **Figure 8B** shows the results of using mMLDA implemented based on Serket, and the categories were learned in a

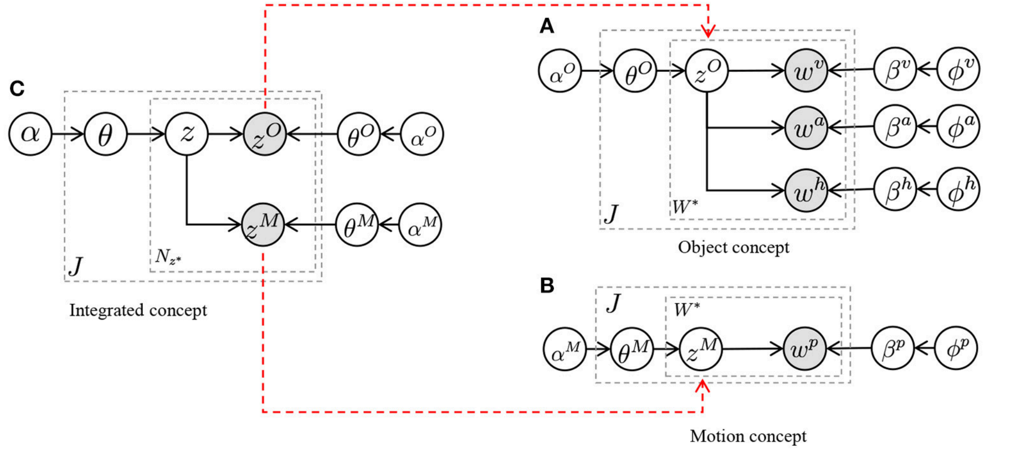
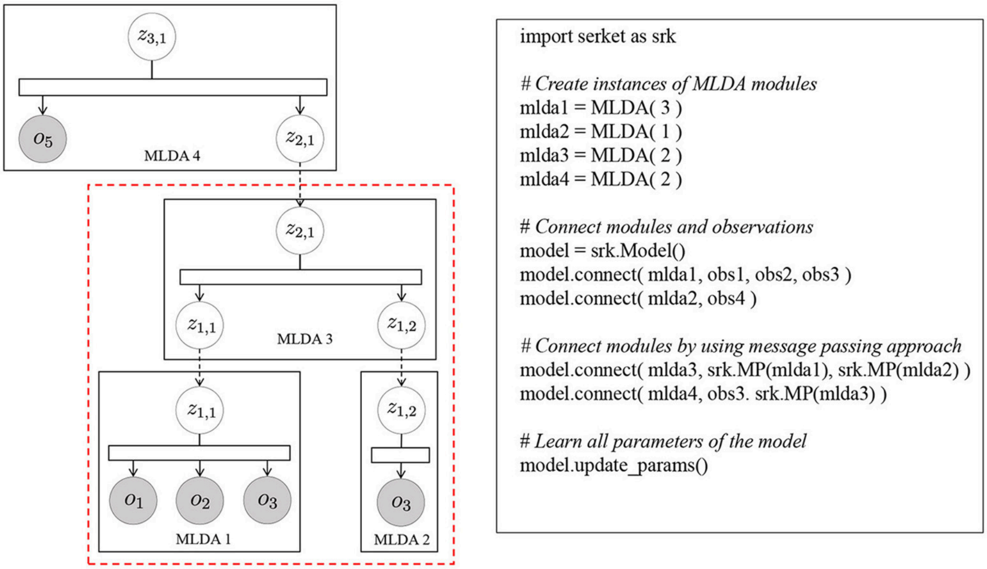


FIGURE 6 | Implementation of mMLDA by connecting three MLDAs. The dashed arrows denote the conditional dependencies represented by Serket. **(A)** Object concept, **(B)** motion concept, and **(C)** integrated concept.



```
import serket as srk

# Create instances of MLDA modules
mlda1 = MLDA( 3 )
mlda2 = MLDA( 1 )
mlda3 = MLDA( 2 )
mlda4 = MLDA( 2 )

# Connect modules and observations
model = srk.Model()
model.connect( mlda1, obs1, obs2, obs3 )
model.connect( mlda2, obs4 )

# Connect modules by using message passing approach
model.connect( mlda3, srk.MP(mlda1), srk.MP(mlda2) )
model.connect( mlda4, obs3. srk.MP(mlda3) )

# Learn all parameters of the model
model.update_params()
```

FIGURE 7 | Pseudo code of mMLDA.

complementary manner. The classification accuracies were 100% and 94%. The motion that could not be classified correctly by the independent model was classified correctly. Moreover, the object classification accuracy improved by 22% owing to the effects of motion categories. In the independent model, category five (shampoos) objects were classified as category seven because of their visual similarity. On the other hand, in the mMLDA based on Serket, they were misclassified as category three (dressings) because the same motion (pouring) was performed with these objects. Also, the rattles (category 10) were misclassified because the rattles (category 10) and soft toys (category nine) had a similar appearance and the same motion (throwing) was performed with them. However, other objects were classified correctly, and this fact indicates that mutual learning was realized by Serket.

Furthermore, we conducted an experiment to investigate the efficiency of the original mMLDA which was not divided into modules. The results in **Figure 8C** show that the accuracies of the classification of objects and motions were 100 and 94%, respectively, although misclassified objects differed from that of the Serket implementation of mMLDA because of sampling. One can see that mMLDA implementation based on Serket is comparable with the original mMLDA.

Table 1 shows the computation time of mMLDA implemented by each method. The Independent model was fastest because the parameters of two MLDAs were independently learned. Serket implementation was slower than the independent model but faster than the original mMLDA. In the original MLDA, all the observations were used for parameter estimation of the integrated concept model. On the other hand, in the Serket implementation, this was approximated and only the parameters sent from lower-level MLDA in Equations (21, 22) were used for parameter estimation of the integrated concept models. Thus, the Serket implementation is faster than the original mMLDA.

4.3. Deeper Model

In the original mMLDA, the structure of the model was fixed, and we derived the equations to estimate its parameters and then implemented them. However, by using Serket, we can flexibly change the structure of the model without deriving the equations for the parameter estimation. As one example, we changed the structure of mMLDA and constructed a deeper model as shown in **Figure 9**. To confirm that the parameters can be learned by using Serket, we generated training data by using the following generative process:

$$z_{5,1} \sim P(z|\theta_5) \tag{26}$$

$$\boldsymbol{o}_5 \sim P(\boldsymbol{o}|\phi_{z_{5,1}}) \tag{27}$$

for $m = 4$ to 1:

$$z_{m,1} \sim P(z|z_{m+1,1}, \boldsymbol{\theta}_m) \tag{28}$$

$$\boldsymbol{o}_m \sim P(\boldsymbol{o}|\boldsymbol{\phi}_{z_{m,1}}) \tag{29}$$

where m denote the index of hierarchies, and the number of categories of all modules was 10. $\boldsymbol{\theta}_m$ and $\boldsymbol{\phi}_z$ were randomly generated, and we used uniform distribution as $P(z|\theta_5)$. This generative process was repeated 50 times, and 250 observations were made. The parameters were estimated by classifying these 250 observations through a Serket implementation and independent model. **Table 2** shows the classification accuracies in each hierarchy. We can see that the Serket implementation outperformed the

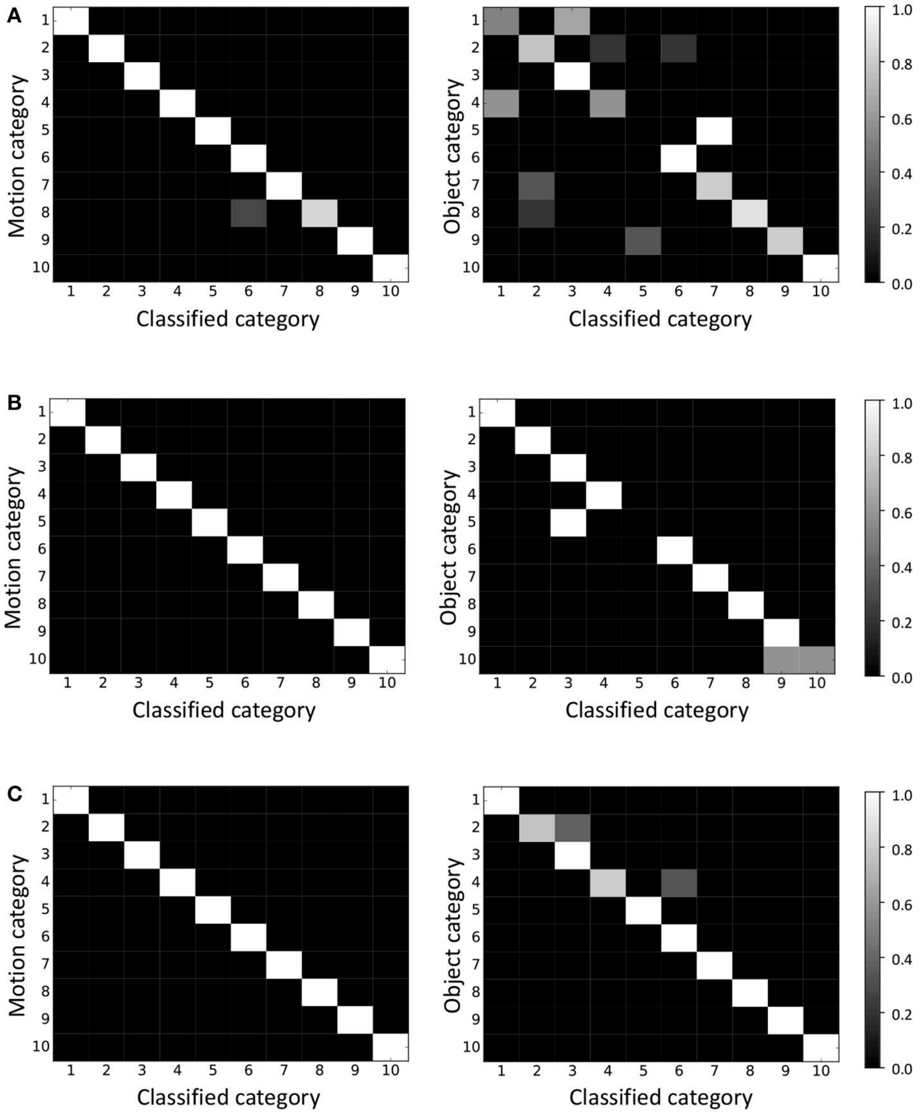


FIGURE 8 | Classification results of motion and object by **(A)** independent model, **(B)** Serket implementation, and **(C)** original model. The classification accuracies for motions and objects were **(A)** 98 and 72%, **(B)** 100 and 94%, and **(C)** 100 and 94%, respectively.

TABLE 1 | Computational time of mMLDA.

Methods	Time (seconds)
Independent model	1.77
Serket implementation	21.4
Original model	64.1

independent model because the parameters were optimized as a whole by using an MP approach. Usually, the equations for parameter estimation must be derived for each model individually; deriving them for a more complicated model is difficult. However, Serket makes it possible to construct a complicated model flexibly and to estimate the parameters easily.

5. EXAMPLE 2: MUTUAL LEARNING OF CONCEPT MODEL AND LANGUAGE MODEL

In Nakamura et al. (2014) and Nishihara et al. (2017), we proposed a model for the mutual learning of concepts and the language model shown in **Figure 4B**; its parameters were estimated by dividing the models into smaller parts. In this section, we show that this model can be constructed by Serket. To learn the model, the visual, auditory, and haptic information obtained by the robot and teaching utterances given by a human user were used. The details are explained in Appendix 2. As in the previous experiment, the original model has upper-bound performance. Therefore, the purpose of this experiment is also to show that Serket implementation has comparable performance with the original model.

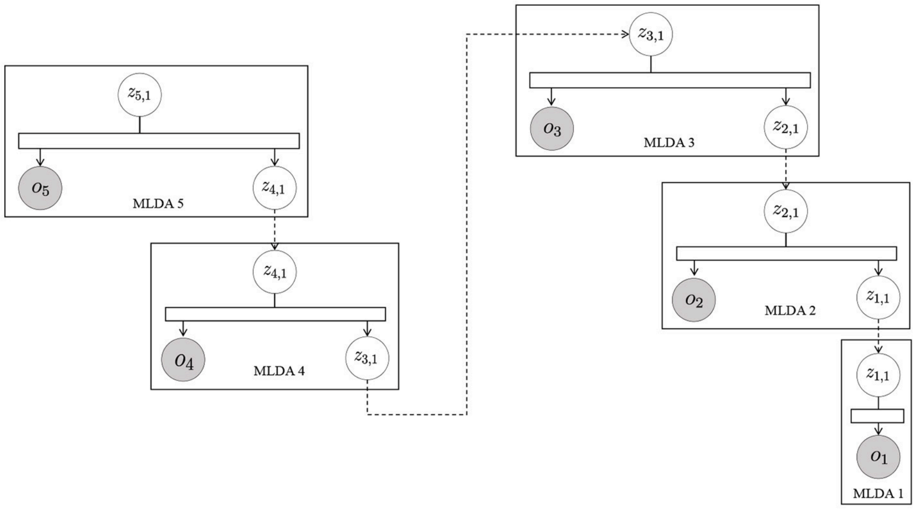


FIGURE 9 | mMLDA that has five hierarchies.

TABLE 2 | Classification accuracies of mMLDA having five hierarchies.

Methods	$z_{1,1}$ (%)	$z_{2,1}$(%)	$z_{3,1}$(%)	$z_{4,1}$(%)	$z_{5,1}$(%)	Average
Independent model	70.0	66.0	74.0	76.0	66.0	70.4
Serket implementation	100	90.0	100	100	100	98.0

5.1. Implementation Based on Serket

Here, we reconsider the mutual learning model based on Serket. The model shown in **Figure 4B** is a one where the speech recognition part and the MLDA that represents the object concepts are connected, and can be divided as shown in **Figure 10**. The MLDA makes it possible to form object categories by classifying the visual, auditory, and haptic information obtained, as shown in the Appendix 2. In addition, the words in the recognized strings of a user's utterances to teach object features are also classified in the model shown in **Figure 10**. Through this categorization of multimodal information and teaching utterance, the words and multimodal information are connected stochastically, which enables the robot to infer the sensory information represented by the words. However, the robot cannot obtain the recognized strings directly; it can only obtain continuous speech. Therefore, in the model shown in **Figure 10**, the words s which are in the recognized strings are treated as latent variables and connected to the model for speech recognition. The parameter $\mathcal{L}$ of the language model is also a latent variable, and is learned from the recognized strings of continuous speech $\boldsymbol{o}$ using the nested Pitman–Yor language model (NPYLM) (Mochihashi et al., 2009). Furthermore, it is an important point of this model that the MLDA and speech recognition model are connected through the words s, which makes it possible to learn them in a complementary manner. That is, the speech is not only recognized based on the similarity of $\boldsymbol{o}$ but is accurately recognized by utilizing the inferred words s from the multimodal information perceived by the robot.

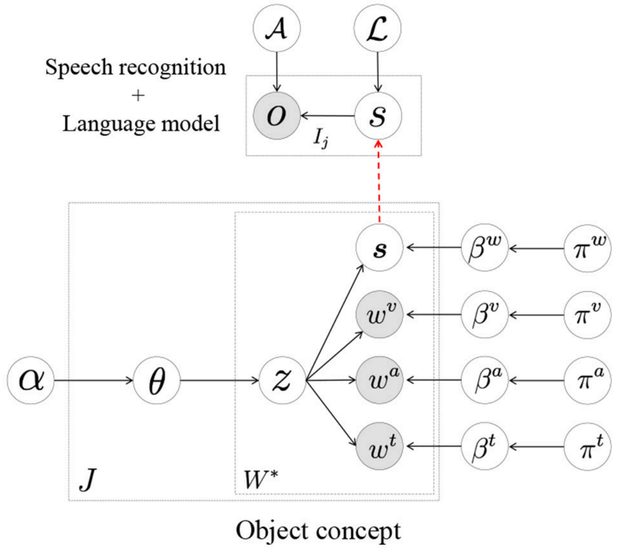


FIGURE 10 | Mutual learning model of concepts and language model.

First, as the initial parameter of $\mathcal{L}$, we used the language model where all phonemes were generated with equal probabilities. The MP approach can be used if all teaching utterances $\boldsymbol{O}$ are recognized by using a language model whose parameter is $\mathcal{L}$ and the probability $P(\boldsymbol{S}|\boldsymbol{O}, \mathcal{A}, L)$ that the word sequences $\boldsymbol{S}$ are generated can be computed. However, it is actually difficult to compute the probabilities for all possible word segmentation patterns of all possible recognized strings. Therefore, we approximated this probability distribution using the SIR approach. The L-best speech recognition results were utilized as samples because it is difficult to compute the probabilities for all possible recognized strings. $\boldsymbol{s}_j^{(l)}$ represents the l-th recognized string of a teaching utterance given the j-th object.

By applying the NPYLM and segmenting them into words, the word sequences $\boldsymbol{S} = \{\boldsymbol{s}_j^{(l)} | 1 \leq l \leq L, 1 \leq j \leq J\}$ can be obtained.

$$\boldsymbol{S} \sim P(\boldsymbol{S}|\boldsymbol{S}', \mathcal{L}). \tag{30}$$

These generated samples are sent to the MLDA module, and the samples that are likely to represent multimodal information are sampled based on the MLDA whose current parameter is $\boldsymbol{\Theta}$:

$$\hat{\boldsymbol{s}}_j \sim P(\boldsymbol{s}_j^{(l)} | \boldsymbol{w}_j^v, \boldsymbol{w}_j^a, \boldsymbol{w}_j^t, \boldsymbol{\Theta}). \tag{31}$$

The selected samples $\hat{\boldsymbol{s}}_j$ are considered as words that can represent multimodal information. Then, the MLDA parameters are updated using a set of these words $\hat{\boldsymbol{S}} = \{\hat{\boldsymbol{s}}_j | 1 \leq j \leq J\}$ and a set of multimodal information $\boldsymbol{W}^v, \boldsymbol{W}^a, \boldsymbol{W}^t$ by utilizing Gibbs sampling.

$$\boldsymbol{\Theta} = \text{argmax}\, P(\hat{\boldsymbol{S}}, \boldsymbol{W}^v, \boldsymbol{W}^a, \boldsymbol{W}^t | \boldsymbol{\Theta}). \tag{32}$$

Moreover, $\hat{\boldsymbol{S}}$ is sent to the speech recognition model, and the parameter $\mathcal{L}$ of the language model is updated.

$$\mathcal{L} = \text{argmax}\, P(\hat{\boldsymbol{S}} | \hat{\boldsymbol{S}}', \mathcal{L}), \tag{33}$$

where $\hat{\boldsymbol{S}}'$ denotes strings obtained by connecting words in $\hat{\boldsymbol{S}}$. The parameters of the whole model can be optimized by iteration through the following process: the sampling words using Equation (30), the resampling words using Equation (31), and the updating parameters using Equations (32, 33).

Figure 11 displays the pseudocode and the corresponding graphical model. In this model, one of modules is MLDA with three observations and one shared latent variable connected to the speech recognition module. o_1, o_2, and o_3 represent multimodal information obtained by the sensors on the robot, and o_4, which is an observation of the speech recognition model, represents the utterances given by the human user. Although the parameter estimation of the original model proposed in Nakamura et al. (2014) and Nishihara et al. (2017) is very complicated, it can be briefly described by connecting the modules based on Serket.

5.2. Experimental Results

We conducted an experiment where the concepts were formed using the aforementioned model to demonstrate the validity of Serket. We compared the following three methods.

(a) A method where speech recognition results $\boldsymbol{S}_0'$ of teaching utterances with maximum likelihoods are segmented into words by the applied NPYLM, and the words obtained are used for concept formation.

(b) A method where the concepts and language model are learned by a mutual learning model implemented based on Serket. (Proposed method)

(c) A method where the concepts and language model are learned by a mutual learning model implemented without Serket proposed in (Nakamura et al., 2014). (Original method)

In method (a), the following equation was used instead of Equation (30), and the parameter $\mathcal{L}$ of the language model was not updated:

$$\boldsymbol{S}_0 \sim P(\boldsymbol{S}|\boldsymbol{S}_0', \mathcal{L}). \tag{34}$$

Alternatively, method (b) was implemented by Serket, and the concepts and language model were learned mutually through the shared latent variable $\boldsymbol{s}$.

Table 3i shows the speech recognition accuracies of each method. In method (a), the language model was not updated; therefore, the accuracy is equal to phoneme recognition. In contrast, in method (b), the accuracy is higher than that of method (a) by updating the language model from the words sampled by MLDA.

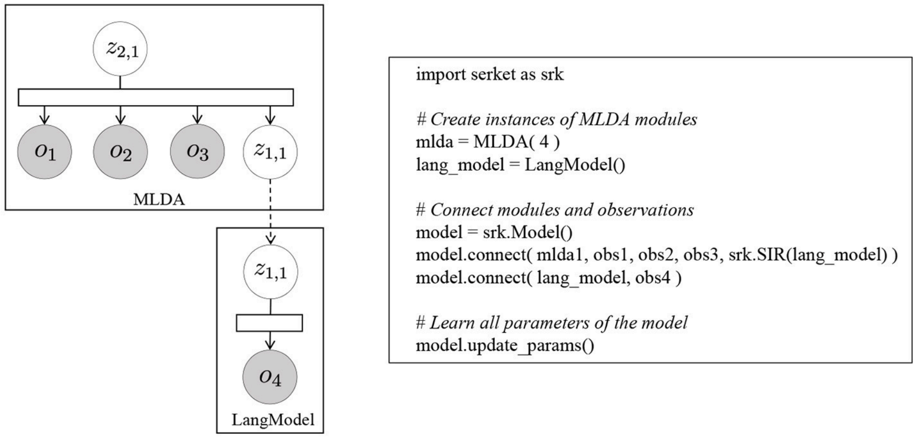
Table 3ii shows the accuracies of word segmentation. Segmentation points were evaluated, as shown in **Table 4**, by applying dynamic-programming matching to find the correspondence between the correct and estimated segmentation. This table shows a case where the correct segmentation of a correctly recognized string "ABCD" is "A/BC/D," and the recognized string "AACD" is segmented into "A/A/CD." ("/" represents the cut points between each word.) The points that were correctly estimated (**Table 4b**), as cut points were evaluated as true positive (TP), and those that were incorrectly estimated (**Table 4d**) were evaluated as false positive (FP). Similarly, the points that were erroneously estimated as not cut points (**Table 4f**) were evaluated as false negative (FN). From the evaluation of the cut points, the precision, recall, and F-measure are computed as follows.

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \tag{35}$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \tag{36}$$

$$F = \frac{2RP}{R + P}, \tag{37}$$

where N_{TP}, N_{FP}, and N_{FN} denote the number of points evaluated as TP, FP, and FN, respectively. Comparing the precision of methods (a) and (b) in **Table 3ii**, one can see that it increases according to Serket. This is because more correct words can be selected among the samples generated by the speech recognition module. Alternatively, the recall of method (b) decreases because some functional words (e.g., "is" and "of") are connected with other words such as "bottleof." However, the precision of method (b) is higher, and its F-measure is greater than 0.11. Therefore, method (b), which was implemented based on Serket, outperformed method (a). **Table 3iii** displays the object classification accuracy. One can observe that the accuracy of method (b) is higher because the speech can be recognized more correctly. Moreover, the Serket implementation [method (b)] was comparable to the original implementation [method (c)]. Thus, the learning of the object concepts and language model presented



```
import serket as srk

# Create instances of MLDA modules
mlda = MLDA( 4 )
lang_model = LangModel()

# Connect modules and observations
model = srk.Model()
model.connect( mlda1, obs1, obs2, obs3, srk.SIR(lang_model) )
model.connect( lang_model, obs4 )

# Learn all parameters of the model
model.update_params()
```

FIGURE 11 | Pseudocode of mutual learning of concept model and language model.

TABLE 3 | Accuracies of speech recognition, segmentation, and object classification.

	(i) Speech recognition	**(ii) Segmentation**			**(iii) Object classification**
Methods		**Precision**	**Rcall**	**F-measure**	
(a) w/o mutual learning	0.64	0.50	0.68	0.58	0.80
(b) Serket implementation	0.74	0.91	0.59	0.72	0.94
(c) Original model	0.77	0.95	0.59	0.73	0.94

TABLE 4 | Evaluation of segmentation.

	(a)	**(b)**	**(c)**	**(d)**	**(e)**	**(f)**	**(g)**
Correct segmentation:	A	/	B		C	/	D
Estimated segmentation:	A	/	A	/	C		D
Evaluation:	TN	TP	TN	FP	TN	FN	TN

TABLE 5 | Computation time of mutual learning model.

Methods	**Time (seconds)**
w/o mutual learning	135
Serket implementation	2,640
Original model	2,637

in Nakamura et al. (2014); Nishihara et al. (2017) was realized by Serket.

Table 5 shows the computation time of mutual learning models. From this figure, the model without mutual learning is fastest because the parameters of one MLDA and language model are independently learned once. On the other hand, Serket implementation is slower and comparable with the original model. This is because the parameters of the MLDA and language model in the Serket implementation were updated iteratively by communicating the parameters with the MP approach, and the computational cost was not much different from that of the original model.

6. CONCLUSION

In this paper, we proposed a novel architecture where the cognitive model can be constructed by connecting modules, each of which maintains programmatic independence. Two approaches were used to connect these modules. One is the MP approach, where the parameters of the distribution are of a finite dimension and communicated between the modules. If the parameters of the distribution are of an infinite dimension or a complex structure, the SIR approach was utilized to approximate them. In the experiment, we demonstrated two implementations based on Serket and their efficiency. The experimental results demonstrated that the implementations are comparable with the original model.

However, there is an issue with regard to the convergence of the parameters. If a large number of samples can be obtained, each latent variable can be locally converged into global optima because the MP, SIR, and MH approaches are based on the existing Markov chain Monte Carlo method. But when various types of models are connected, it is not clear whether all latent parameters can be converged into global optima as a whole. It was confirmed that the parameters were converged in the models used in the experiments. Nonetheless, this remains a difficult and important issue which will be examined in future work.

We believe that models that can be connected by Serket are not limited to generative probabilistic models, although we focused on the connected generative probabilistic models in this paper. Neural networks or other methods can be one of the modules of Serket, and we are planning to connect them. Furthermore, we believe that large-scale cognitive models can be constructed by connecting various types of modules, each of which represent a particular brain function. In so doing, we will realize our goal of artificial general intelligence. Serket can also contribute to developmental robotics (Asada et al., 2009; Cangelosi et al., 2015), where the human developmental mechanism is understood via a constructive approach. We believe that robots can learn capabilities ranging from motor skills to language, and these can be developed using Serket, as it makes it possible to understand humans.

AUTHOR CONTRIBUTIONS

ToN, TaN and TT conceived of the presented idea. ToN developed the theory and performed the computations. ToN wrote the manuscript with support from TaN and TT. TaN and TT supervised the project. All authors discussed the results and contributed to the final manuscript.

ACKNOWLEDGMENTS

This work was supported by JST CREST Grant Number JPMJCR15E3.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2018.00025/full#supplementary-material

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2016). Tensorflow: large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., et al. (2016). "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning* (San Juan), 173–182.

Anderson, J. R. (2009). *How Can the Human Mind Occur in the Physical Universe?* Oxford, UK: Oxford University Press.

Ando, Y., Nakamura, T., Araki, T., and Nagai, T. (2013). "Formation of hierarchical object concept using hierarchical latent dirichlet allocation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Tokyo), 2272–2279.

Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., et al. (2009). Cognitive developmental robotics: a survey. *IEEE Trans Auton. Mental Develop.* 1, 12–34. doi: 10.1109/TAMD.2009.2021702

Attamimi, M., Ando, Y., Nakamura, T., Nagai, T., Mochihashi, D., Kobayashi, I., et al. (2016). Learning word meanings and grammar for verbalization of daily life activities using multilayered multimodal latent dirichlet allocation and bayesian hidden markov models. *Adv. Robot.* 30, 806–824. doi: 10.1080/01691864.2016.1172507

Blei, D., Griffiths, T., and Jordan, M. (2010). The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *J. ACM* 57:7. doi: 10.1145/1667053.1667056

Blei, D. M., and Jordan, M. I. (2003). "Modeling annotated data," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval* (Toronto, ON), 127–134.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *J. Mach. Learn. Res.* 3, 993–1022.

Cangelosi, A., Schlesinger, M., and Smith, L. B. (2015). *Developmental Robotics: From Babies to Robots*. Cambridge, MA: The MIT Press.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., et al. (2016). Stan: A probabilistic programming language. *J. Statist. Softw.* 20, 1–37. doi: 10.18637/jss.v076.i01

Chen, Y., and Filliat, D. (2015). "Cross-situational noun and adjective learning in an interactive scenario," in *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics* (Providence, RI), 129–134.

Chollet, F. (2015). *Keras.* Available online at: https://github.com/fchollet/keras

Françoise, J., Schnell, N., and Bevilacqua, F. (2013). "A multimodal probabilistic model for gesture-based control of sound synthesis," in *21st ACM international conference on Multimedia (MM'13)* (Barcelona), 705–708.

Ghahramani, Z., Jordan, M. I., and Adams, R. P. (2010). "Tree-structured stick breaking for hierarchical data," in *Advances in Neural Information Processing Systems*, eds J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel and A. Culotta (Vancouver, BC), 19–27.

Goodman, N., Mansinghka, V., Roy, D. M., Bonawitz, K., and Tenenbaum, J. B. (2012). Church: a language for generative models. *arXiv preprint arXiv:1206.3255*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision* (Santiago), 1026–1034.

Kim, D.-k., Voelker, G., and Saul, L. (2013). "A variational approximation for topic modeling of hierarchical corpora," in *International Conference on Machine Learning* (Atlanta, GA), 55–63.

Laird, J. E. (2008). Extending the soar cognitive architecture. *Front. Artif. Intell. Appl.* 171:224.

Lallee, S., and Dominey, P. F. (2013). Multi-modal Convergence Maps: From Body Schema and Self-Representation to Mental Imagery. *Adapt. Behav.* 21, 274–285. doi: 10.1177/1059712313488423

Li, H., Liu, J., and Zhang, S. (2011). "Hierarchical latent dirichlet allocation models for realistic action recognition," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing* (Prague: ICASSP), 1297–1300.

Li, W. and McCallum, A. (2006). "Pachinko allocation: Dag-structured mixture models of topic correlations," in *Proceedings of the 23rd International Conference on Machine Learning* (Pittsburgh, PA: ACM), 577–584.

Mangin, O., Filliat, D., Ten Bosch, L., and Oudeyer, P.-Y. (2015). Mca-nmf: multimodal concept acquisition with non-negative matrix factorization. *PLoS ONE* 10:e0140732. doi: 10.1371/journal.pone.0140732

Mangin, O., and Oudeyer, P.-Y. (2013). "Learning Semantic Components from Subsymbolic Multimodal Perception," in *IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics* (New Delhi), 1–7.

Margaritis, D. (2003). *Learning Bayesian Network Model Structure From Data.* Technical Report, Carnegie-Mellon University Pittsburgh pa School of Computer Science.

Mimura, T., Hagiwara, Y., Taniguchi, T., and Inamura, T. (2017). Bayesian body schema estimation using tactile information obtained through coordinated random movements. *Adv. Robot.* 31, 118–134. doi: 10.1080/01691864.2016.1270854

Minka, T. and Lafferty, J. (2002). "Expectation-propagation for the generative aspect model," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence* (Alberta: Morgan Kaufmann Publishers Inc.), 352–359.

Mochihashi, D., Yamada, T., and Ueda, N. (2009). "Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling," in *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing*, Vol. 1 (Singapore), 100–108.

Nakamura, T., Iwata, K., Nagai, T., Mochihashi, D., Kobayashi, I., Asoh, H., et al. (2016). "Continuous motion segmentation based on reference point dependent gp-hsmm," in *IROS2016: Workshop on Machine Learning Methods for High-Level Cognitive Capabilities in Robotics* (Daejeon).

Nakamura, T., Nagai, T., Funakoshi, K., Nagasaka, S., Taniguchi, T., and Iwahashi, N. (2014). "Mutual learning of an object concept and language model based on mlda and npylm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Chicago, IL), 600–607.

Nakamura, T., Nagai, T., and Iwahashi, N. (2007). "Multimodal object categorization by a Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (San Diego, CA), 2415–2420.

Nakamura, T., Nagai, T., and Iwahashi, N. (2009). "Grounding of word meanings in multimodal concepts using LDA," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (St. Louis, MO), 3943–3948.

Nguyen, V.-A., Boyd-Graber, J. L., Resnik, P., and Chang, J. (2014). "Learning a concept hierarchy from multi-labeled documents," in *Advances in Neural Information Processing Systems*, Vol. 27 (Montreal, QC: Curran Associates, Inc.), 3671–3679.

Nishihara, J., Nakamura, T., and Nagai, T. (2017). Online algorithm for robots to learn object concepts and language model. *IEEE Trans. Cogn. Develop. Syst.* 9, 255–268. doi: 10.1109/TCDS.2016.2552579

Ogata, T., Nishide, S., Kozima, H., Komatani, K., and Okuno, H. (2010). Inter-Modality Mapping in Robot with Recurrent Neural Network. *Patt. Recogn. Lett.* 31, 1560–1569. doi: 10.1016/j.patrec.2010.05.002

Patil, A., Huard, D., and Fonnesbeck, C. J. (2010). Pymc: Bayesian stochastic modelling in python. *J. stat. softw.* 35:1. doi: 10.18637/jss.v035.i04

Piaget, J. and Duckworth, E. (1970). Genetic epistemology. *Am. Behav. Sci.* 13, 459–480. doi: 10.1177/000276427001300320

Putthividhy, D., Attias, H. T., and Nagarajan, S. S. (2010). "Topic regression multi-modal latent dirichlet allocation for image annotation," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (San Francisco, CA: IEEE), 3408–3415.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 779–788.

Ridge, B., Skocaj, D., and Leonardis, A. (2010). "Self-supervised cross-modal online learning of basic object affordances for developmental robotic systems," in *IEEE International Conference on Robotics and Automation* (Anchorage, AK), 5047–5054.

Roy, D. and Pentland, A. (2002). Learning Words from Sights and Sounds: a computational model. *Cogn. Sci.* 26, 113–146. doi: 10.1207/s15516709cog2601_4

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., et al. (2017). Mastering the game of go without human knowledge. *Nature* 550, 354–359. doi: 10.1038/nature24270

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems* (Montreal, QC), 3104–3112.

Taniguchi, A., Hagiwara, Y., Taniguchi, T., and Inamura, T. (2017). "Online spatial concept and lexical acquisition with simultaneous localization and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vancouver, BC).

Taniguchi, T., Hamahata, K., and Iwahashi, N. (2011). Imitation learning architecture for unsegmented human motion using sticky hdp-hmm and mdl-based phrase extraction method. *Adv. Robot.* 25, 2143–2172. doi: 10.1163/016918611X594775

Taniguchi, T., Nagai, T., Nakamura, T., Iwahashi, N., Ogata, T., and Asoh, H. (2016a). Symbol emergence in robotics: a survey. *Adv. Robot.* 11, 706–728. doi: 10.1080/01691864.2016.1164622

Taniguchi, T., Nakanishi, H., and Iwahashi, N. (2010). "Simultaneous estimation of role and response strategy in human-robot role-reversal imitation learning," in *The 11th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems*, Vol. 43 (Valenciennes), 460–464.

Taniguchi, T., Nakashima, R., Liu, H., and Nagasaka, S. (2016b). Double articulation analyzer with deep sparse autoencoder for unsupervised word discovery from speech signals. *Adv. Robot.* 30, 770–783. doi: 10.1080/01691864.2016.1159981

Tokui, S., Oono, K., Hido, S., and Clayton, J. (2015). "Chainer: a next-generation open source framework for deep learning," in *Workshop on Machine Learning Systems in The Twenty-ninth Annual Conference on Neural Information Processing Systems* (Montreal, QC).

Tran, D., Kucukelbir, A., Dieng, A. B., Rudolph, M., Liang, D., and Blei, D. M. (2016). Edward: a library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). "Show and tell: A neural image caption generator," in *IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 3156–3164.

Wang, C., Blei, D., and Fei-Fei, L. (2009). "Simultaneous image classification and annotation," in *IEEE Conference on Computer Vision and Pattern Recognition* (Miami Beach, FL), 1903–1910.

Wermter, S., Weber, C., Elshaw, M., Panchev, C., Erwin, H., and Pulvermuller, F. (2004). Towards multimodal neural robot learning. *Robot. Auton. Syst.* 47, 171–175. doi: 10.1016/j.robot.2004.03.011

Wood, F., van de Meent, J. W., and Mansinghka, V. (2014). "A new approach to probabilistic programming inference," in *International Conference on Artificial Intelligence and Statistics* (Reykjavik), 1024–1032.

Wu, Q., Wang, P., Shen, C., Dick, A., and van den Hengel, A. (2016). "Ask me anything: Free-form visual question answering based on knowledge from external sources," in *IEEE Conference on Computer Vision and Pattern Recognition* (Las Vegas, NV), 4622–4630.

Yang, S., Yuan, C., Hu, W., and Ding, X. (2014). "A hierarchical model based on latent dirichlet allocation for action recognition," in *International Conference on Pattern Recognition* (Stockholm), 2613–2618.

Yuruten, O., Sahin, E., and Kalkan, S. (2013). The learning of adjectives and nouns from affordance and appearance features. *Adapt. Behav.* 21, 437–451. doi: 10.1177/1059712313497976

Zhang, Z., Wu, J., Li, Q., Huang, Z., Traer, J., McDermott, J. H., et al. (2017). "Generative modeling of audible shapes for object perception," in *IEEE International Conference on Computer Vision* (Venice).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 24 July 2018
doi: 10.3389/fnbot.2018.00046

Acquisition of Viewpoint Transformation and Action Mappings via Sequence to Sequence Imitative Learning by Deep Neural Networks

Ryoichi Nakajo [1], *Shingo Murata* [2], *Hiroaki Arie* [2] *and Tetsuya Ogata* [1]*

[1] Department of Intermedia Art and Science, Waseda University, Tokyo, Japan, [2] Department of Modern Mechanical Engineering, Waseda University, Tokyo, Japan

We propose an imitative learning model that allows a robot to acquire positional relations between the demonstrator and the robot, and to transform observed actions into robotic actions. Providing robots with imitative capabilities allows us to teach novel actions to them without resorting to trial-and-error approaches. Existing methods for imitative robotic learning require mathematical formulations or conversion modules to translate positional relations between demonstrators and robots. The proposed model uses two neural networks, a convolutional autoencoder (CAE) and a multiple timescale recurrent neural network (MTRNN). The CAE is trained to extract visual features from raw images captured by a camera. The MTRNN is trained to integrate sensory-motor information and to predict next states. We implement this model on a robot and conducted sequence to sequence learning that allows the robot to transform demonstrator actions into robot actions. Through training of the proposed model, representations of actions, manipulated objects, and positional relations are formed in the hierarchical structure of the MTRNN. After training, we confirm capability for generating unlearned imitative patterns.

Keywords: imitative learning, human-robot interaction, recurrent neural networks, deep neural networks, sequence to sequence learning

OPEN ACCESS

Edited by:
Ganesh R. Naik,
Western Sydney University, Australia

Reviewed by:
Bolin Liao,
Jishou University, China
Eiji Uchibe,
Advanced Telecommunications Research Institute International (ATR), Japan

***Correspondence:**
Tetsuya Ogata
ogata@waseda.jp

Received: *30 November 2017*
Accepted: *03 July 2018*
Published: *24 July 2018*

Citation:
Nakajo R, Murata S, Arie H and Ogata T (2018) Acquisition of Viewpoint Transformation and Action Mappings via Sequence to Sequence Imitative Learning by Deep Neural Networks. Front. Neurorobot. 12:46. doi: 10.3389/fnbot.2018.00046

1. INTRODUCTION

Today there is increased interest in robots capable of working in human living environments. Robot motions are generally preprogrammed by engineers, but it is crucial for robots to learn new actions in work environment contexts if they are to work with humans. One way for robots to learn new actions is imitation, which is the behavioral capability to generate the equivalent actions after the observation of the demonstrator's actions. Imitation is a powerful learning method that humans apply to acquire new actions without resorting to trial-and-error attempts. Hence, robot acquisition of imitative abilities will realize *programming by demonstration* (PbD) (Billard et al., 2008), in which new action skills are acquired from demonstrators without any prior design.

Early studies of imitation learning are related to computational neuroscience, focusing on task-level imitation such as assembly (Kuniyoshi et al., 1994), kendama manipulation (Miyamoto et al., 1996), and tennis serves (Miyamoto and Kawato, 1998). To date, the main approaches to imitative learning have been probabilistic models, reinforcement learning, and neural networks. Among probabilistic models, hidden Markov models realize behavior recognition, generation through imitative learning (Inamura et al., 2004), and imitation of object manipulation (Sugiura et al., 2010). Gaussian mixture models allow robots to imitate human gestures (Calinon et al., 2010). Reinforcement learning has been used for robot acquisition of motor primitives (Kober and Peters, 2010) and applied to task-level learning (Schaal, 1997). By combining reinforcement learning with a Gaussian mixture model, Guenter et al. (2007) achieved robot imitation of reaching movements. Neural network approaches mainly use recurrent neural networks that allow robots to imitate human gesture patterns (Ito and Tani, 2004) and object manipulations (Ogata et al., 2009; Arie et al., 2012).

As another perspective, cognitive developmental robotics (Asada et al., 2009; Cangelosi et al., 2010) has tried to understand the development of the human cognitive abilities through robot experiments based on constructive approaches. In studies focusing on imitative learning, robots were trained to learn imitative tasks by Hebbian learning (Nagai et al., 2011; Kawai et al., 2012) and neural networks (Ogata et al., 2009; Arie et al., 2012; Nakajo et al., 2015). Through training, experimenters observe behavior changes in robots and in the internal states of the learning models, then consider the developmental processes of imitation. The Hebbian learning approach reveals changes in granularity on visual development, allowing the robot to recognize self–other correspondences (Nagai et al., 2011; Kawai et al., 2012). Our previous studies used recurrent neural networks to demonstrate how robots can translate from other to own actions (Ogata et al., 2009), imitative ability for the composition of behaviors (Arie et al., 2012), and recognition of positional relations between self and other (Nakajo et al., 2015).

For robots working in human living environments, imitation of demonstrator behaviors roughly comprises two processes: (1) observing the behavior and (2) transforming the observed behavior into an action. During observations, robots are expected to extract information about the imitated behavior. In the transformation process, robots must extract necessary information from the observations, and match them with their own actions. Robots cannot always observe behaviors from the same position, but are expected to recognize and reproduce behaviors regardless of the position from which they were observed. However, few previous studies have focused on positional relations between robots and demonstrators or considered correspondences between actions provided from various positions.

If robots are to observe demonstrated actions and transform them into the robots' own actions, robots must process raw images and extract from them information necessary for imitation. However, the huge dimensionality of raw data makes direct processing too difficult. Deep-learning techniques are looked to as a solution to this problem (LeCun et al., 2015), because deep learning can process raw data and allows machines to automatically extract necessary information about requested tasks. For instance, deep learning techniques have outperformed previous methods for image recognition (Krizhevsky et al., 2012). Over the past several years, deep learning has been applied to action learning by robots, and many studies have investigated imitative learning through deep learning (Liu et al., 2017; Sermanet et al., 2017; Stadie et al., 2017). Stadie et al. applied deep learning methods to transformation of demonstrator views into robot control features. Sermanet et al. and Liu et al. trained learning models to relate demonstrator views from various positions with the robot view. After training learning models to transform demonstrator views, reinforcement learning (Liu et al., 2017; Stadie et al., 2017) or supervised learning methods (Sermanet et al., 2017) are applied to allow robots to imitate behaviors. Although these learning methods are suited to allowing robots to acquire imitative skills regardless of positional relations, demonstrators cannot provide their views to robots in actual environments; robots must instead capture demonstrator behaviors via cameras, and relate observed behaviors to their own situation.

Various training methods have also been researched in the field of deep learning. One common method applied to robot action learning is end-to-end learning, in which the learning model receives images and robot motor commands, and directly plans the robot's actions. Another technique often applied to natural language translation is sequence to sequence learning (Sutskever et al., 2014), which allows translation of a multi-dimensional time series into another time series. Utilizing this characteristic, Yamada et al. (2016) allowed a robot to perform tasks based on language instructions. This characteristic can also be applied to imitative learning, because robots must translate observations of demonstrator actions into their own actions. We thus consider the application of sequence to sequence learning to imitative learning.

The main contribution of this paper is demonstration of how a robot can acquire the following two abilities: (1) automatic visual-feature extraction, and (2) transformation from human demonstration into robotic action when positional differences are present. This paper proposes an imitative learning model that simultaneously enables a robot to acquire positional relations between a demonstrator and the robot, and transforms observed actions into the robot's own actions. In the learning process, the robot observes demonstrator actions using a mounted camera, and no pre-training is provided. To achieve imitative abilities, we combined two deep neural network models. An autoencoder extracts visual features from raw camera images, and a dynamic neural network model called a multiple timescale recurrent neural network (MTRNN) (Yamashita and Tani, 2008) is trained to learn how to imitate tasks . An MTRNN learns positional relations between a demonstrator and a robot. To allow the robot to learn how to translate observed actions into its own actions, the MTRNN is trained based on a sequence to sequence approach (Sutskever et al., 2014). In experiments, we imposed object manipulation tasks on a robot and conducted predictive learning to train the proposed learning model. After training, we confirmed that the robot could translate observed actions into its own actions. By inspecting the internal states of the MTRNN, we show how the robot recognizes positional relations between the demonstrator and the robot during tasks. We also considered what information the robot extracts through observation and translates into actions.

2. METHODS

2.1. Sequence to Sequence Learning of Imitative Interaction

We first describe the method by which robots use our proposed learning model to learn imitative interactions. We apply sequence to sequence learning (Sutskever et al., 2014) to map observed demonstrator actions to robot actions. sequence to sequence learning is a learning method for RNNs that is mainly used in the machine translation field. By inputting to RNNs series of sentences in the original and target languages, sequence to sequence learning allows forward propagation in the RNNs both to recognize the meaning of the original sentence and to generate a sentence in the target language by using the internal states acquired through encoding the original language. We use sequence to sequence learning to encode demonstrator actions and to generate robot actions. As **Figure 1** shows, by concatenating demonstrator and robot actions and inputting the concatenated sequences to a RNN, the network is expected to learn how to map the demonstrator actions to robot actions.

2.2. Overview of Proposed Learning Model

Robot imitation of demonstrator actions requires observation of demonstrator actions and transformation of observed actions into robot actions. The robot must process visual information to extract information related to demonstrator actions. Captured camera images have too many dimensions to process directly. The robot thus requires functions for automatically compressing and extracting visual information. To map extracted visual information from demonstrator actions to robot actions, visual features and robot motor information must be integrated into a single learning scheme. Doing so requires another learning model for integrating this information, separate from visual feature compression.

Our proposed learning model satisfies these conditions by including two neural networks. The first is a deep neural network called a convolutional autoencoder (CAE), which is applied to extraction of visual features from camera images. The second is a multiple timescale recurrent neural network (MTRNN), which we use to integrate time series of extracted visual features with robot motor information. **Figure 2** shows an overview of the proposed learning model. In the following subsections, we explain the CAE method for extracting visual features and the MTRNN method for integrating them with motor information.

2.3. Visual Feature Extraction via Convolutional Autoencoder

An autoencoder is a neural network with bottleneck layers, and comprises an encoder for dimensionally compressing input images and a decoder for restoring dimensionality in output images (Hinton and Salakhutdinov, 2006). Updating learnable parameters in the autoencoder to identically output an input image allows the network to acquire lower-dimensional features representing input images at the narrowest layer. By compressing input images, the robot can nondestructively extract visual features of camera images.

In this study, we applied a convolutional autoencoder (CAE), which is an autoencoder including convolution layers (Masci et al., 2011). Convolution is an arithmetic process inspired by the mammalian visual cortex, and is expected to extract visual features by focusing on spatial localities in the images. We combined a conventional CAE with fully connected layers. Camera images are taken as input, then the CAE is trained to minimize the mean squared error between input and reconstructed images. The mean squared error E_{AE} is processed as

$$E_{\mathrm{AE}} = \frac{1}{N}\sum_{n}^{N} E_{\mathrm{AE}}^{(n)}, \tag{1}$$

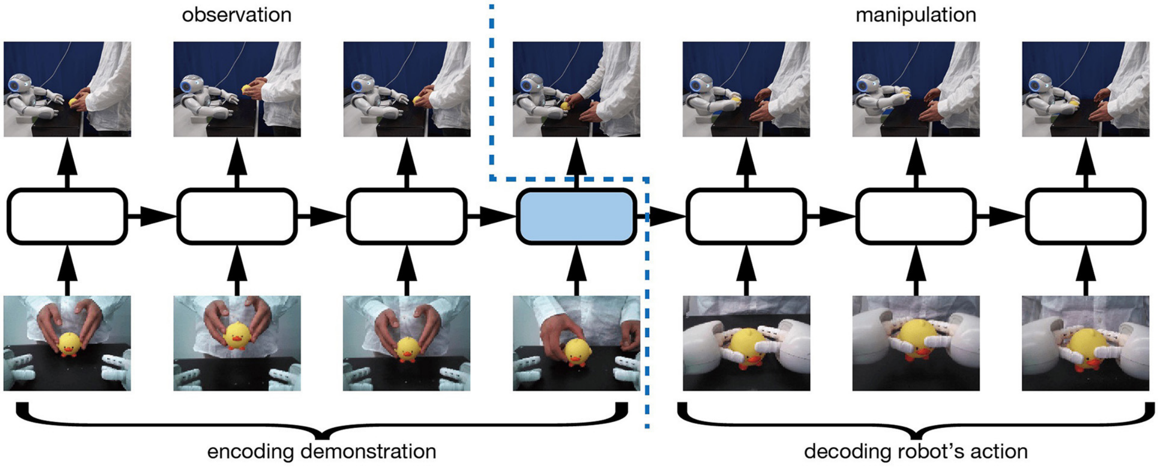


FIGURE 1 | sequence to sequence learning scheme of the RNN. In the first half of the time sequence, the robot moves only its head and captures images of only the action being demonstrated. From the captured images, the RNN is expected to recognize and encode the demonstrator actions. In the second half of the time sequence, the RNN receives encoded internal states, plans robot actions, and issues robot motor commands.

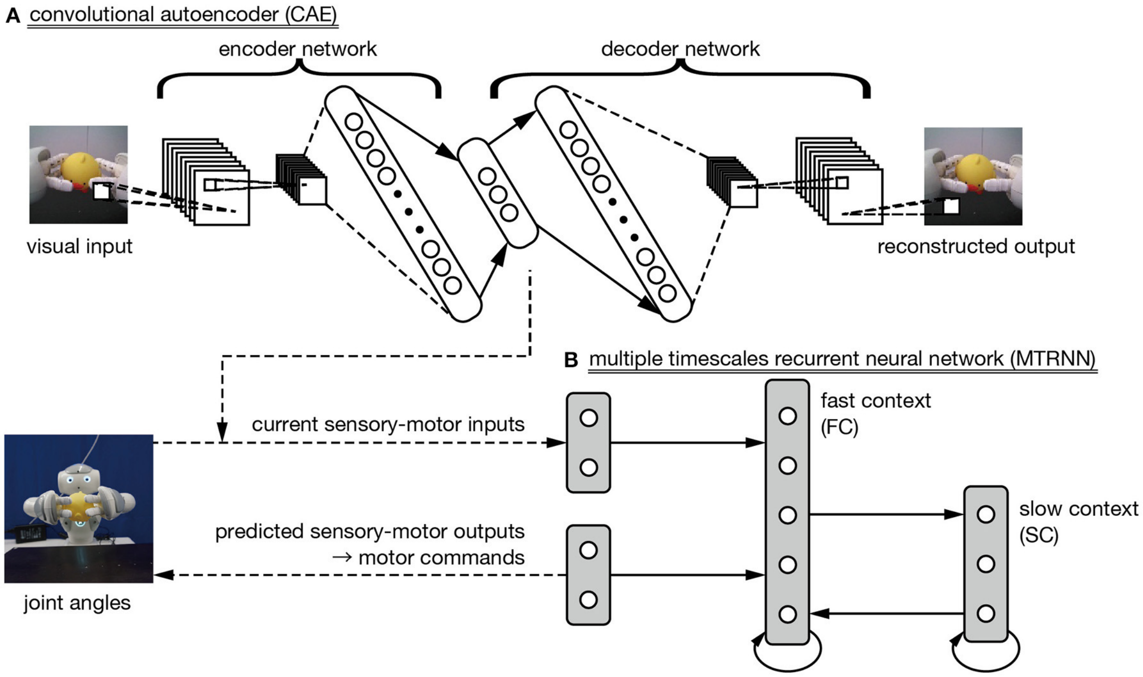

A convolutional autoencoder (CAE)
encoder network
decoder network
visual input
reconstructed output
B multiple timescales recurrent neural network (MTRNN)
current sensory-motor inputs
fast context (FC)
slow context (SC)
predicted sensory-motor outputs
→ motor commands
joint angles

FIGURE 2 | The proposed learning model. **(A)** A convolutional autoencoder (CAE) is trained to extract visual features in images from a robot-mounted camera. **(B)** A multiple timescale recurrent neural network is used to integrate CAE-extracted visual features and robot motor information.

$$E_{\mathrm{AE}}^{(n)} = \frac{1}{HWC} ||\hat{\mathbf{X}}^{(n)} - \mathbf{X}^{(n)}||_2^2, \qquad (2)$$

where N is the number of mini-batches; $\hat{\mathbf{X}}^{(n)}$ is the nth input image; $\mathbf{X}^{(n)}$ is the nth reconstructed image; and H, W, and C indicate the height, width, and channel, respectively, of the images. To avoid drastic changes in extracted visual features between continuous time steps, we furthermore applied the following slow penalty introduced in Finn et al. (2016):

$$g(\mathbf{f}_t) = \eta \cdot ||(\mathbf{f}_{t+2} - \mathbf{f}_{t+1}) - (\mathbf{f}_{t+1} - \mathbf{f}_t)||_2^2 \qquad (t \geq 1), \quad (3)$$

where $\mathbf{f}_t$ indicates the visual features extracted from an image at time step t, and η is a hyper-parameter to control the strength of the penalty.

2.4. Sensory-Motor Integration by Multiple Timescale Recurrent Neural Network

Generating imitative actions from observation of demonstrator actions requires a function that integrates visual features extracted by the CAE with robot motor information. In this work, we use a dynamic neural network model called a multiple timescales recurrent neural network (MTRNN) (Yamashita and Tani, 2008). An MTRNN has different time constants in its hierarchically context layers. The layer connected to the input–output layers [“fast context” (FC) in **Figure 2B**] is a group of neurons with a smaller time constant, and so responds more quickly to current external inputs. Another layer connected only to neurons in the context layers [“slow context” (SC) in **Figure 2B**] has a larger time constant, and so responds more slowly. Yamashita and Tani (2008) demonstrated that stacking layers with different timescales allows the robot to acquire action primitives in the FC layer, and described the order of sequential combinations of primitives in the SC layer.

In MTRNN forward propagation, the internal state of the ith FC, SC, and output neural unit at time step t, $(u_{t,i})$, for the sth sequence is calculated as

$$u_{t,i}^{(s)} = \begin{cases} \left(1 - \frac{1}{\tau_i}\right) u_{t-1,i}^{(s)} + \frac{1}{\tau_i} \left(\sum_{j \in I_{\mathrm{I}}} w_{ij} x_{t,j}^{(s)} + \sum_{j \in I_{\mathrm{FC}} \cup I_{\mathrm{SC}}} w_{ij} c_{t-1,j}^{(s)} + b_i \right) & (t \geq 1, i \in I_{\mathrm{FC}}), \\ \left(1 - \frac{1}{\tau_i}\right) u_{t-1,i}^{(s)} + \frac{1}{\tau_i} \left(\sum_{j \in I_{\mathrm{FC}} \cup I_{\mathrm{SC}}} w_{ij} c_{t-1,j}^{(s)} + b_i \right) & (t \geq 1, i \in I_{\mathrm{SC}}), \\ \sum_{j \in I_O} w_{ij} c_{t,j}^{(s)} + b_i & (t \geq 1, i \in I_{\mathrm{O}}), \end{cases} \qquad (4)$$

where I_{FC}, I_{SC}, and I_{O} are index sets of the respective neural units, τ_i is the time constant of the ith neuron, w_{ij} is the connective weight from the jth to the ith neural units, $x_{t,j}^{(s)}$ is the external input of the jth neural unit at time step t of the sth sequential data, $c_{t,j}^{(s)}$ is the activation value of the jth context neuron at time step t of the sth sequence, and b_i is the bias of the ith neural unit. We use tanh as the activation function for the context neural unit $c_{t,i}^{(s)}$ and output unit $y_{t,i}^{(s)}$.

We trained the MTRNN by minimizing the mean squared error with the gradient descent method. The mean squared error E_{RNN} is described as

$$E_{\mathrm{RNN}} = \frac{1}{S}\sum_{s}^{S}\frac{1}{T^{(s)}}\sum_{t}^{T}E_{\mathrm{RNN},t}^{(s)}, \tag{5}$$

$$E_{\mathrm{RNN},t}^{(s)} = \frac{1}{Y}\,||\hat{\mathbf{y}}_t^{(s)} - \mathbf{y}_t^{(s)}||_2^2, \tag{6}$$

where S is the number of sequential data, $T^{(s)}$ is the number of time steps of the sth sequential data item, Y is the number of neural units in the output layer, $\hat{\mathbf{y}}_t^{(s)}$ is the target sensory-motor values at time step t of the sth sequence, and $\mathbf{y}_t^{(s)}$ is the predicted sensory-motor values at time step t of the sth sequence. The learnable parameters of the MTRNN are composed of connected weights $\mathbf{w}$, biases $\mathbf{b}$, and initial internal states in context layers $\mathbf{u}_0^{(s)}$. The gradients of these learnable parameters follow a conventional back propagation through time method (Rumelhart et al., 1986).

3. EXPERIMENT

3.1. Task Design

This section describes an experimental task given to a humanoid robot (NAO; Aldebaran Robotics). The task in this experiment is imitative interaction for object manipulation as shown in **Figure 3A.** Imitative interaction cycles comprised four processes: (i) the demonstrator shows the object manipulation action to the robot, then (ii) passes the manipulated object to the robot. Next, (iii) the robot mimics the observed manipulation, and (iv) the demonstrator receives the object from the robot. Furthermore, actions, manipulated objects, and positional relationships between the robot and the demonstrator were varied between cycles. Manipulated objects were two toys (a *chick* and a *watering can*), shown in **Figure 3B.** Objects were manipulated in two ways (*move-side* and *move-up*) as shown in **Figure 3C**. The positional relationship between the robot and the demonstrator varied according to where the demonstrator presented the action. We define 180° as the position when the robot presents a motion in front of itself. Accordingly, 120, 150, 180, 210, and 240° counterclockwise in the positive direction are used as the positional relationship between the demonstrator and the robot. **Figure 3D** shows a schematic diagram of positional relations between the demonstrator and the robot . Under these conditions, combinations that can be taken in a single cycle come in 20 patterns, from two objects, two movements, and five positional relations.

3.2. Training Data

This subsection describes the method for creating sequential training data. In this experiment, the training data consisted of time series of the robot joint angles and 120 × 160 RGB images captured by a front-facing camera mounted in its mouth. The CAE extracts visual features from captured images. Controlled joints had four degrees of freedom (DoF) (ShoulderPitch, ShoulderRoll, ElbowYaw, and ElbowRoll) at each arm and two DoF (HeadPitch and HeadYaw) at the neck.

To prepare the training data, the robot was controlled and actual joint angles and images were recorded. A control method for both arms was predesigned and the arms tracked the planned trajectories with noise. Gaussian noise was added into the planned trajectories to augment the training data, with the noise variance set as 0.0001. Neck joint angles were operated by proportional–integral–derivative control, so manipulated object centroids were centered in camera images during interaction. While recording training data, joint angles and camera images were sampled every 400 ms. Because recorded joint angle and camera image information had different value ranges, the information was normalized before input to the neural networks: joint angles were scaled to $[-1.0, 1.0]$ according to angle limits, and image pixel values were normalized from $[0, 255]$ to $[-1.0, 1.0]$.

This experiment separately recorded the processes of imitative interaction tasks such as demonstrator and robot actions and object passing. After recording, processes were combined and an imitative interaction cycle was generated. There were 160 time steps for demonstrator and robot actions and 60 for passing objects between the demonstrator and the robot, for a total of 440 time steps. Each sequence of 20 combinations was generated five times, for a total of 100 instances of recorded data.

3.3. Training of CAE and MTRNN

The robot was trained with imitative interaction tasks through predictive learning of recorded time series including joint angles and camera images.

3.3.1. Visual Feature Learning via CAE

We first trained the CAE with camera images to extract visual features for input to the MTRNN with robot joint angles. Input 120 × 160 RGB images have 57,600 dimensions. These input images were trained to minimize errors between the original inputs and reconstructed images, and to extract 10 visual features from the middle CAE layer. **Table 1** presents the detailed CAE structure used in this learning experiment. For CAE training, we conducted mini-batch training with an Adam optimizer (Kingma and Ba, 2015), setting Adam hyperparameters as $\alpha = 0.01$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$, mini-batch sizes of 200, and slow penalty strength as $\eta = 1.0 \times 10^{-5}$. Learnable CAE parameters were updated 7,500 times.

3.3.2. Sensory-Motor Integration Learning via MTRNN

After extracting visual features by the trained CAE, time series of sensory-motor information were generated by concatenating robot joint angles and extracted visual features. To allow the robot to carry out imitative interactions, training sequences

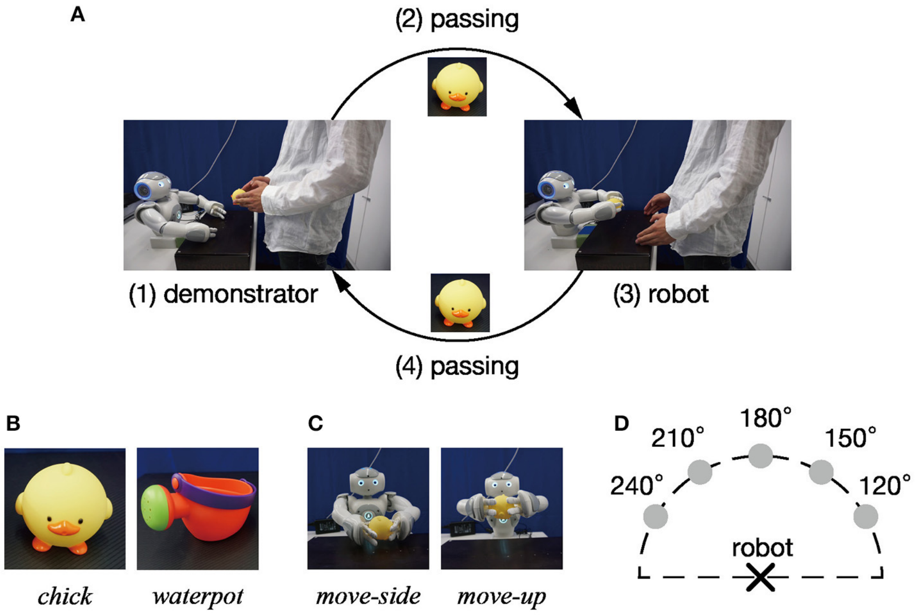


FIGURE 3 | Task design: **(A)** A single cycle of the imitative interaction task given to the robot comprises four components: (1) action presentation by the demonstrator, (2) passing the manipulated object to the robot, (3) generating an imitative action by the robot, and (4) receiving the object by the demonstrator. **(B)** Objects manipulated during imitative interaction. **(C)** Imitative robot actions. **(D)** Positional relation between robot and demonstrator. The position of actions that the robot observed in front of the demonstrator is defined as 180°, and five positions (120, 150, 180, 210, and 240°) are labeled counterclockwise.

TABLE 1 | The structure of the CAE.

The *l*th layer	Input	Output	Processing	Kernel size	Stride	Padding
1	(120, 160, 3)	(60, 80, 16)	Conv	(4, 4)	(2, 2)	(4, 4)
2	(60, 80, 16)	(30, 40, 32)	Conv	(4, 4)	(2, 2)	(4, 4)
3	(30, 40, 32)	(10, 10, 64)	Conv	(6, 8)	(3, 4)	(6, 8)
4	(10, 10, 64)	(2, 2, 128)	Conv	(10, 10)	(5, 5)	(10, 10)
5	512	250	Linear	–	–	–
6	250	10	Linear	–	–	–
7	10	250	Linear	–	–	–
8	250	512	Linear	–	–	–
9	(2, 2, 128)	(10, 10, 64)	Deconv	(10, 10)	(5, 5)	(10, 10)
10	(10, 10, 64)	(30, 40, 32)	Deconv	(6, 8)	(3, 4)	(6, 8)
11	(30, 40, 32)	(60, 80, 16)	Deconv	(4, 4)	(2, 2)	(4, 4)
12	(60, 80, 16)	(120, 160, 3)	Deconv	(4, 4)	(2, 2)	(4, 4)

In the "Processing" column, conv, deconv, and linear respectively indicate convolutional encoding, deconvolutional decoding, and fully-connected transformation. The input dimensions for convolutional and deconvolutional layers are shown as (height, width, channel), and fully-connected layers are shown as d.

for input to the MTRNN were created by connecting several combinations of imitative tasks. In this case, training sequences were sequences of four randomly selected imitative tasks, with overlapping allowed. An interval of 5–30 time steps was inserted between the connected time series. The robot retained the same pose during this interval. Under these conditions, 100 sequences were generated as MTRNN training data.

While there were 20 combinations of imitative tasks, we trained the MTRNN with 10 combinations to evaluate generalizability to unlearned combinations. **Table 2** shows the 10 combinations used for MTRNN training to predict the next state of joint angles and visual features. There were 10 joint angles and 10 extracted visual features, for a total of 20 dimensions input to the MTRNN. We set the number of neural units in the FC and SC layers as 180 and 20 and time constant values as 2.0 and

TABLE 2 | MTRNN training sequences.

	120°	150°	180°	210°	240°
move-side	C	W	C	W	C
move-up	W	C	W	C	W

Rows show actions, and columns show positional relationships. In each cell, characters C and W indicate the manipulated object (chick or watering can). The time sequence indicated in each cell is used for MTRNN training.

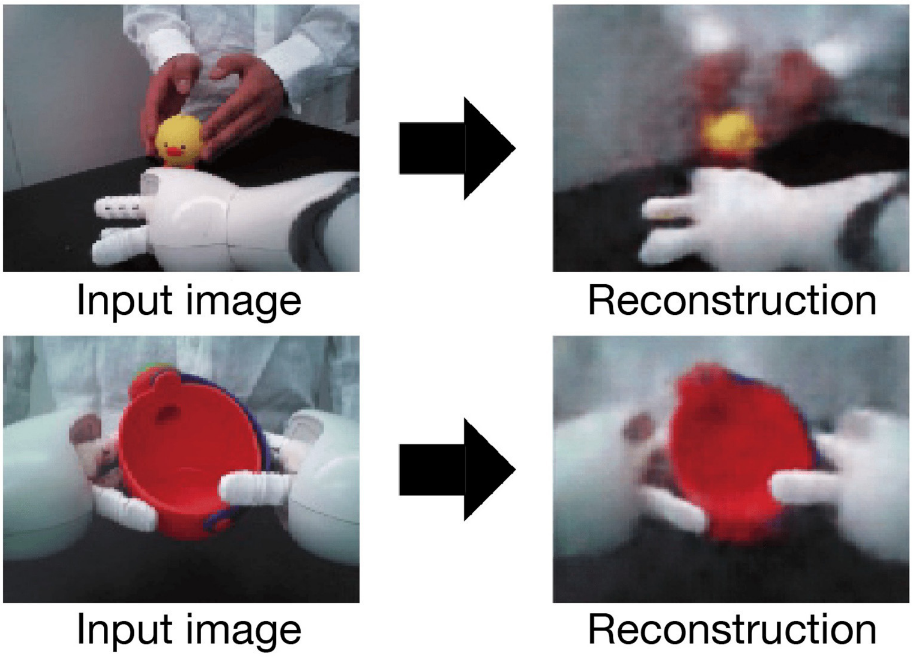


FIGURE 4 | Image reconstruction by trained CAE. The upper figure shows an image of a [*move-side*, *chick*] demonstration, and the lower figure shows an image of [*move-up*, *watering can*] generated by the robot.

64.0, respectively. For training, we used the Adam optimizer with hyperparameters $\alpha = 0.01$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. Learnable parameters were updated with these settings 10,000 times.

4. TRAINING RESULTS

4.1. Reconstructed Images by CAE

After CAE training, the mean squared error between trained images and their reconstructed output was at most 0.0141. The worst mean squared error between untrained and reconstructed images was 0.0150. **Figure 4** shows a selection of reconstructed and untrained images. The reconstructed image in **Figure 4** suggests that the trained CAE could regenerate original input images. We applied principal component analysis to visual features extracted by the CAE at the beginnings of the demonstrations and robotic actions. As shown in **Figure 5A**, the positional relationships between the demonstrator and the robot were separated in the visual features at the beginning of the demonstrations. **Figure 5B** shows that the manipulated objects were separated in the visual features at the beginning of the robotic actions. The CAE could extract the visual features from images, thus we used time series of the extracted visual features for training of the MTRNN. An example of a time series of the extracted visual features is shown in **Figure 5C**.

4.2. Robot Action Generation

After MTRNN training, we evaluated the mean squared error between trained target sequences and predicted output, which was 0.00140 at worst. We input new sequences generated with the combination including untrained series, and evaluated the mean squared error. In that case, the evaluated value was 0.00164 at worst. **Figure 6** shows the MTRNN-predicted output against the untrained input [*move-side*, *chick*] as observed from position 150°. By using predicted output of the MTRNN against untrained input, the robot could imitate demonstrator actions.

4.3. Internal States in MTRNN

Principal component analysis was performed on the internal MTRNN state to grasp the internal structure the MTRNN acquired through predictive learning of robot sensory-motor information. We conducted PCA on internal states in the FC and SC layers at the time when the demonstrator ended the actions. **Figure 7** shows the difference in the positional relationship between the demonstrator and the robot in the FC layer, and **Figure 8** shows the difference between imitative actions and manipulated objects. As shown in **Figure 7**, the FC layer in the MTRNN separated positional relationships between the robot and the demonstrator when demonstrator actions were complete. At the same time, differences in imitative actions are clustered in the plane described by PC1 and PC2 of the internal states in the SC layer (see the upper graph in **Figure 8**). In contrast, in the plane described by PC3 and PC4 the differences between manipulated objects are separated by the dashed line in the lower graph in **Figure 8**.

We next extracted internal states in the SC layer at the time when the robot starts its action, and plotted the PCA results in **Figure 9**. As that figure shows, combinations of imitative actions and manipulated objects were clustered in the SC layer. The actions were distinguished at the beginning of robot imitation, so

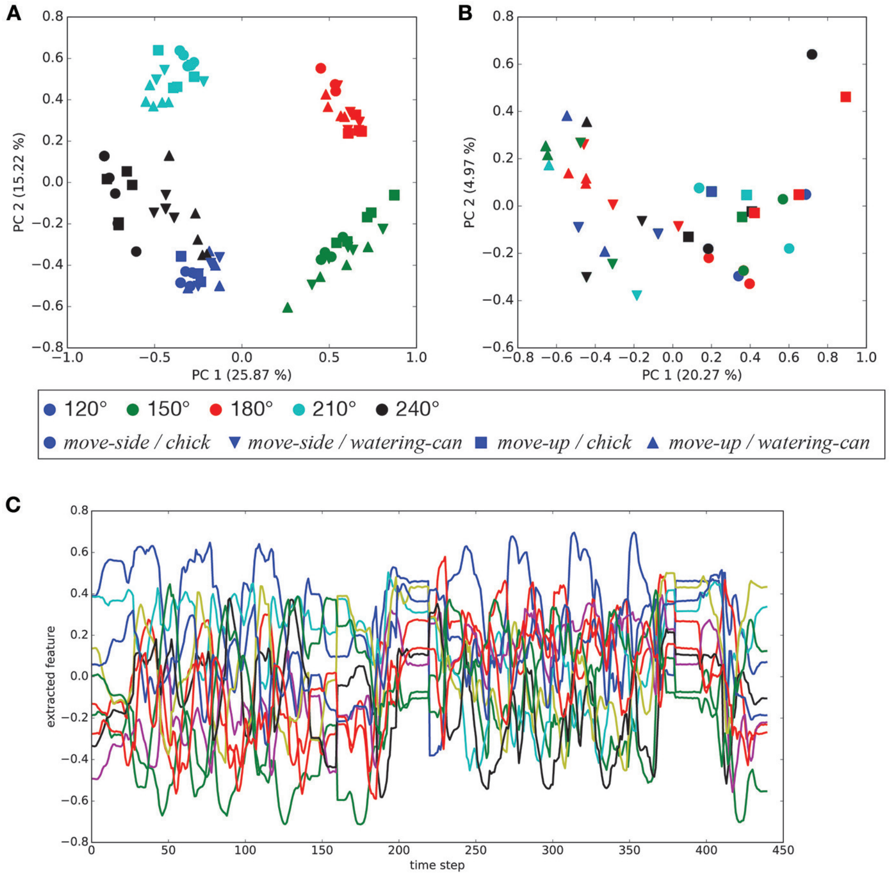


FIGURE 5 | Visual features extracted by the CAE: **(A)** principal components of the visual features at the beginning of demonstrations (PC1–PC2), **(B)** principal components of the visual features at the beginning of robotic actions (PC1–PC2), and **(C)** an example of a time series of the visual features for [*move* − *side*, *watering* − *can*, 150°].

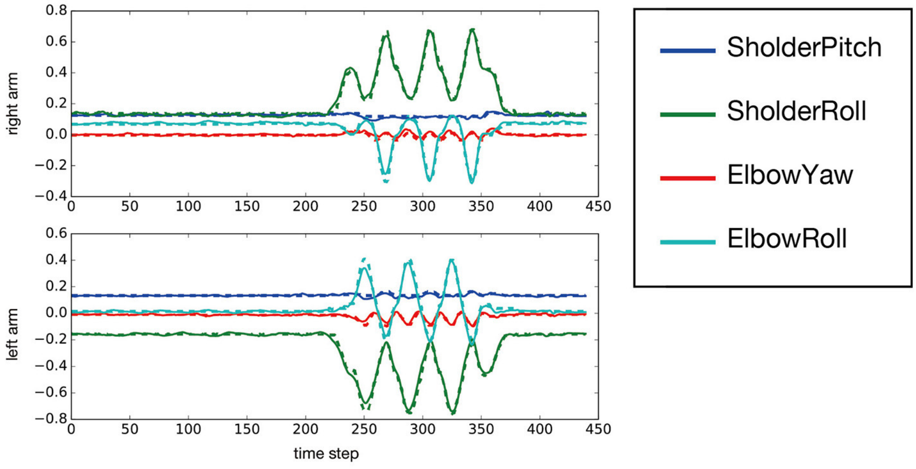


FIGURE 6 | The predicted output of an untrained [*move* − *side*, *chick*] sequence observed from the 150° position. This figure shows only the prediction for both arms. The horizontal axis indicates time steps, and the vertical axis represents predicted output of the joint angles. The solid and dotted lines show output by the MTRNN and target sequences, respectively.

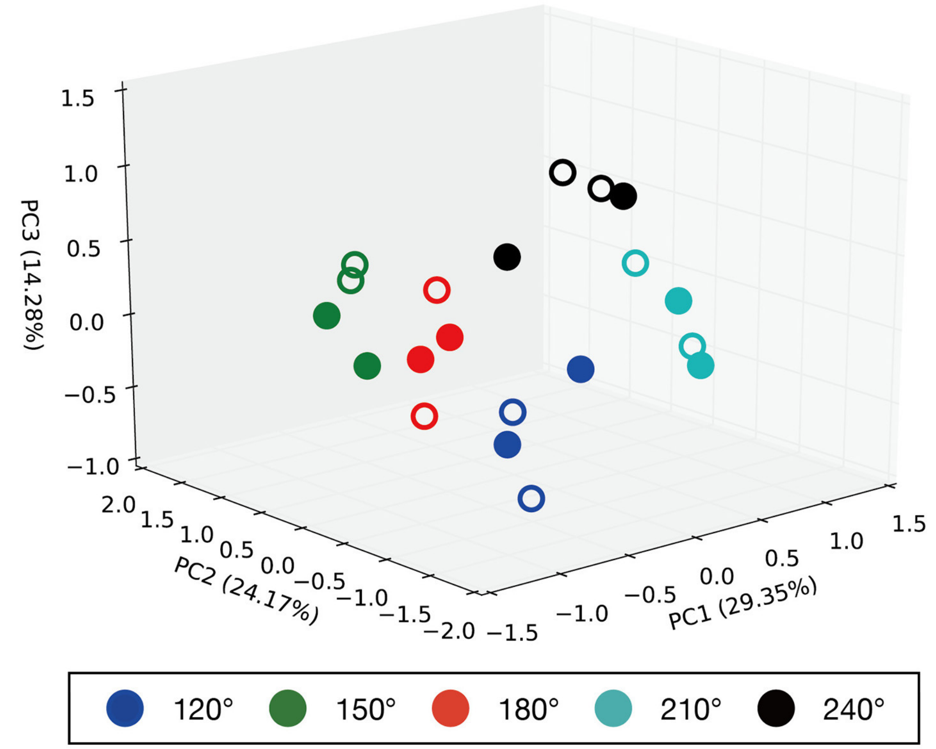


FIGURE 7 | Results of PCA of the internal states in the FC layer when demonstrator actions are finished. PC1, PC2, and PC3 are plotted in the 3D space. Numbers in parentheses indicate contribution ratios of each principle component. Filled points are trained imitative patterns, and others are unlearned patterns. The positional relationships are separated in the 3D space.

the robot could map observed actions to corresponding imitative actions in advance. Similarly, the robot could acquire an ability to carry out imitative actions while retaining information about manipulated objects in the internal MTRNN states. Furthermore, unlearned patterns indicated in **Figure 9** were recognized, so the MTRNN could acquire the ability to generalize via combinations of actions and manipulated objects.

One time step during the robot action was chosen and the internal states were analyzed at that time. Since robot motions comprised 160 steps, we chose the middle (80th) time step and visualized the internal states by PCA. **Figure 10** shows internal states of the FC layer at that time, and confirms that the robot distinguished between different combinations of actions and manipulations while performing imitative actions. In contrast, principle components in the FC layer do not show positional relations between the demonstrator and the robot. Therefore, the robot could transform observations into actions regardless of the positional relation. Finally, to confirm how internal MTRNN states transit during imitative interaction, we plotted the time development of neural units in the SC layer during interaction in a plane. **Figure 11** shows transitions of neural activities in the SC layer during imitative interactions. The positional relationship between the demonstrator and the robot is fixed as 120°, and combinations of actions and manipulated objects are separately shown. The figure shows that the internal states for all patterns start from the beginning of demonstrator actions (◯), transit to robot actions (△), and finally reach the same point where manipulated objects are passed from the robot to the demonstrator (□). Since the internal states always reach the same point, the robot could continue to recognize the actions, manipulated objects, and positional relations after a single imitative interaction. Other positional relations also acquired results similar to those in **Figure 11**.

5. DISCUSSION

We proposed a possible imitative model that allows a robot to acquire the ability to recognize positional relations between the demonstrator, and to transform observed actions into robot actions. The imitative model had two neural networks: (1) a CAE that was trained to extract visual features from captured raw images, and (2) an MTRNN that integrated and predicted sensory-motor information. Through training of image reconstruction by the CAE, the robot could extract visual features from raw images captured by its camera. By sensory-motor integration through predictive learning with the MTRNN, the robot could recognize information that relates imitative interactions, such as positional relations between the demonstrator and the robot. In the rest of this section, we compare earlier studies with our current work, and clarify the distinction between them.

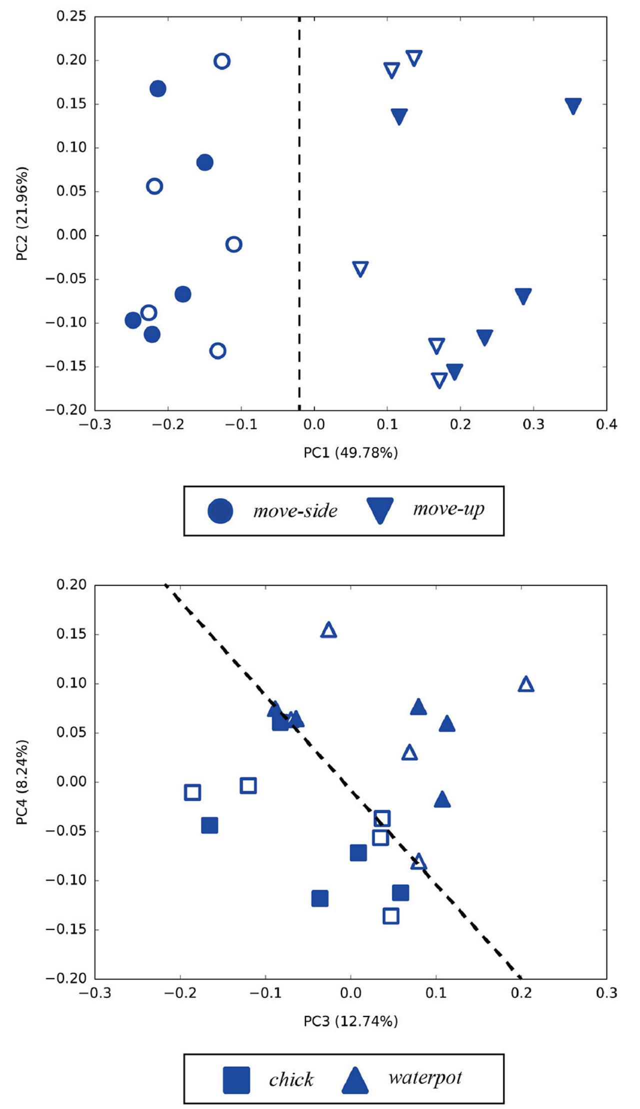


FIGURE 8 | Results of PCA of internal states in the SC layer when demonstrator actions are finished. Numbers in parentheses indicate contribution ratios of each principle component. Filled points are trained imitative patterns, and others are unlearned patterns. In the upper figure (PC1–PC2), differences of actions are separated in the PC1 direction. In the lower figure (PC3–PC4), differences of manipulated objects are classified by the dashed line.

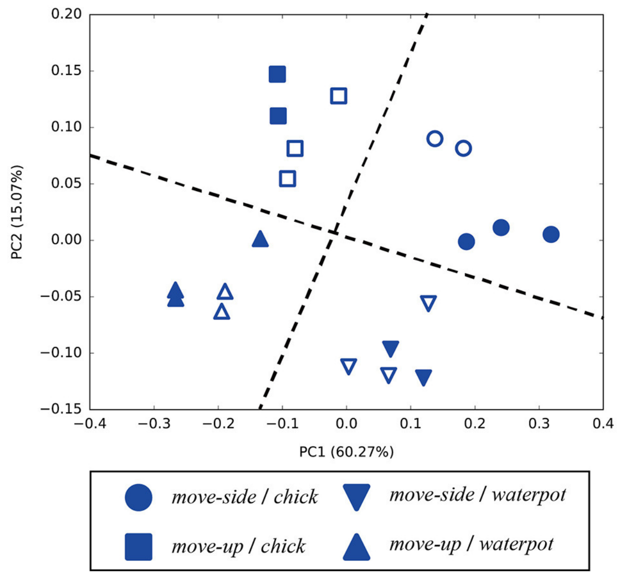


FIGURE 9 | Internal states in the SC layer at the beginning of robot actions (PC1–PC2). Filled points indicate trained imitative patterns, and outlined marks are unlearned. Combinations of imitative actions and manipulated objects can be clustered by the two dotted lines.

From the viewpoint of acquiring positional relations between the demonstrator and robot, our proposed model allows the robot to recognize positional relations via predictive learning of sensory-motor sequences. By including differences in positions between the demonstrator and the robot, the proposed learning model might be forced to optimize these differences during predictive learning. Thanks to the hierarchical structure of the MTRNN and the sequence to sequence learning methods, the robot might come to process positional differences in the FC layer (shown in **Figure 7**), and possess information required for robot actions, such as kinds of actions and manipulated objects in the SC layer (see **Figures 8**, **9**). In this work, the sequence to sequence learning method was tried for encoding the demonstrator's actions into the plan of robotic actions. Thus, the information necessary for the robotic actions may be encoded in the SC layer, and the information necessary for the current prediction may appear in the FC layer. In the current experiment, the robotic actions do not require any positional relationships between the demonstrator and the robot. Therefore, positional relationships may remain in the FC layer. Furthermore, from **Figure 10**, conducting sequence to sequence learning that translates demonstrator actions into robot actions might allow the robot to properly transform observed actions into the same actions. In previous works, positional relations between demonstrator and robot were represented by coordinate transformations described as mathematical formulations (Billard et al., 2004; Lopes et al., 2010). Our proposed model requires no designed transformation to acquire positional relations between the demonstrator and robot. In this experiment, the robotic head moved through imitative interaction, and its joint angles differed for each positional relationship during the demonstration phase. These difference in the robotic head depended on the positional relationships between the demonstrator and the robot. Thus, the proposed learning model might require optimizing for these differences during predictive learning. Through predictive learning of sensory-motor sequences, including positional differences between the demonstrator and robot, the robot could automatically recognize differences and transform demonstrator actions into robot actions. Our previous work (Nakajo et al., 2015) allowed robots to acquire information about actions and positional relations by labeling this information and providing constraints that make

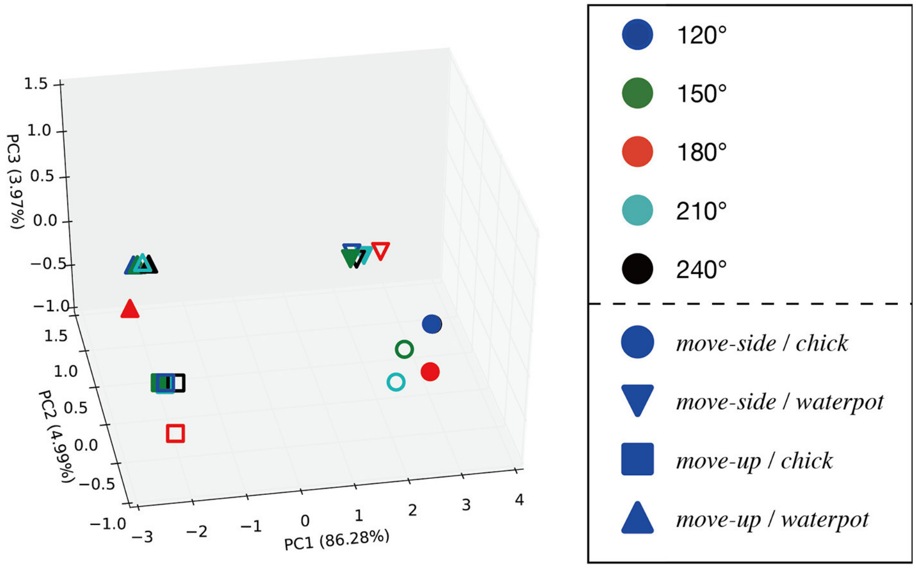


FIGURE 10 | Internal states in the FC layer while conducting robot actions (PC1–PC2–PC3). Filled points indicate trained imitative patterns, and outlined marks are unlearned patterns. Actions and objects are distinguished between in this 3D space, but positional differences between the demonstrator are ignored.

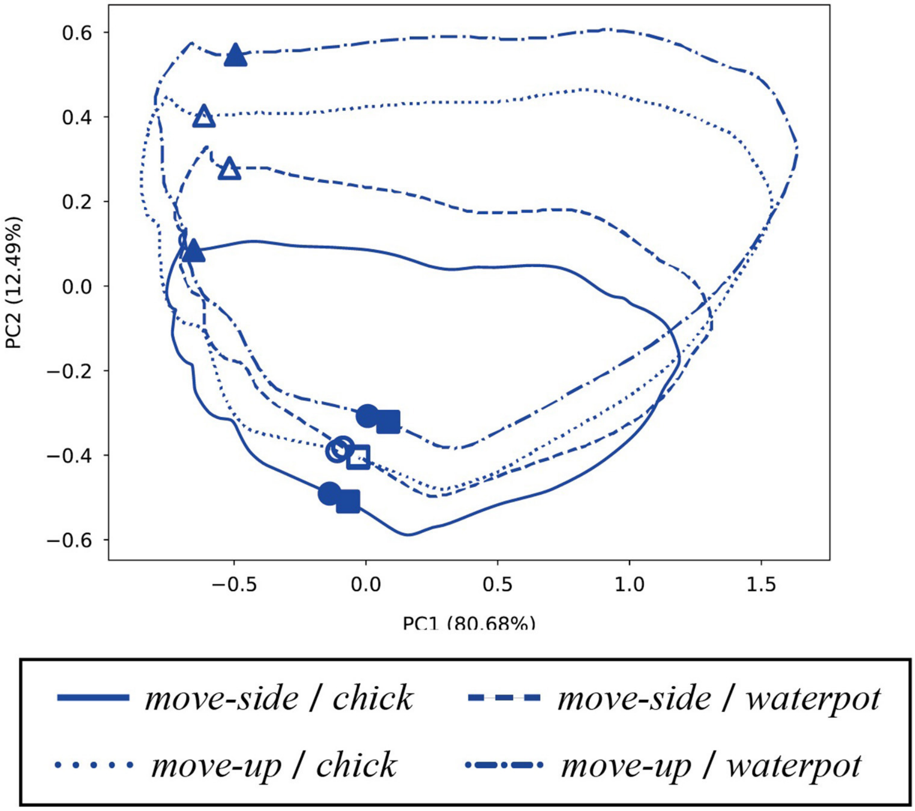


FIGURE 11 | Transition of neural activities in the SC layer during imitative interaction (PC1–PC2). The positional relationship between the demonstrator and the robot is fixed as 120°, and combinations of actions and manipulated objects are separately plotted. Symbols ○, △, and □ respectively indicate the beginning of demonstrator actions, the beginning of robot actions, and the end of passing objects to the demonstrator. Filled marks indicate trained patterns, and others indicate unlearned patterns. All transitions start from ○ points, pass through △, and finally return to similar □ points near the beginning of demonstrator actions (○).

activities of neural units representing the same information close. In contrast, the current work eliminates labeling of actions and positional relations by conducting sequence to sequence learning.

From the perspective of action translations, sequence to sequence learning methods might contribute to learning how to translate demonstrator actions into robot actions. As **Figures 7**, **8** show, the robot recognized positional relations, actions, and manipulated objects in the demonstration phase. From **Figure 10**, after a demonstration, the robot could perform observed actions regardless of positional relation. Thanks to the characteristics of sequence to sequence learning, which can translate one multidimensional sequence into another sequence, the robot acquired the ability to choose information necessary for conducting actions. In addition, we conducted a validation trial in which the demonstrations from untrained positional relationships (135, 165, 195, and 225°) were given to the MTRNN. The demonstrations observed from all untrained positions could be translated into the proper robotic actions by the MTRNN. On the other hand, although the MTRNN could map the untrained positional relationships into the points between the trained positional relationships, sometimes mapping failed and these relationships appeared at different points in the PCA space of **Figure 7**. These failures might come from visual features extracted by the CAE. In the current experiment, differences in the positional relationships were present in the visual images and the joint angles of the robotic head. However, the CAE did not learn to extract visual features from the untrained positional relationships. Thus, it may be difficult to extract these visual features with the CAE, which could affect predictions by the MTRNN. Previous studies applied separate modules to transform positional differences (Ogata et al., 2009; Liu et al., 2017; Sermanet et al., 2017). Ogata et al. (2009) used a mixture-of-experts algorithm, where each expert module translated demonstrator actions provided from a different position. Positional relations that the robot recognized were thus limited by the number of experts, although the robot could imitate observed actions from various positions. In this paper, every positional relationship is acquired within the internal structure of a single RNN, so the robot can process various positional relations. Sermanet et al. (2017) and Liu et al. (2017) used deep neural networks that associated demonstrator views with robot views. These methods were very powerful, because no previous knowledge was required to associate the views. However, third-person views were synchronized with robot views where needed to translate actions. In this paper, the robot required its own views, so a robot-mounted camera was necessary in an actual environment. Furthermore, from the viewpoint of transforming actions, previous works used separate modules to extract invariances that were included in views, and additional training was required to learn robot actions. Our proposed model allowed the robot to simultaneously learn recognition of positional relations and action transformation, so no pre-training was needed to integrate sensory-motor information.

When we train the CAE to extract visual features from the robot's vision, we discretely input visual frames. However, in sensory-motor integration for achieving sequential tasks, visual feature learning in which the learning model sequentially predicts images may be required. In the experiment described in this paper, robot actions were determined at the end of the demonstration, and only passing of objects occurs between the end of demonstrator actions and the beginning of robot actions. Thus, both internal representations in the SC layer might be similar. However, discrimination of manipulated objects was not acquired at the end of demonstrator actions, as shown in **Figure 8**. Discrimination of manipulated objects was instead achieved at the beginning of robot actions, as shown in **Figure 9**. This difference in representations might come from prediction error arising from visual information. For the CAE, the difficulty of reconstructing any object comes from the size of object regions. Specifically, reconstructing smaller objects is more difficult than larger objects. In this paper, the regions of manipulated objects during demonstration are smaller than those during robot actions. It thus seems more difficult for the CAE to reconstruct manipulated objects in the demonstration phase. This difficulty of reconstruction might affect sensory-motor integration, as seen in the internal representations in the SC layer. Video prediction in which the learning model is trained to sequentially predict images would contribute to overcoming this problem. Thanks to sequential prediction, the learning model applies histories of past predictions to the current prediction. Moreover, we separately trained the CAE and the MTRNN. Therefore, through training of sensory-motor integration with the MTRNN, no feedback was sent to visual processing by the CAE. However, to allow the robot to more properly process sensory-motor sequences, the prediction error should affect all processing in the learning model. A previous work by Hwang and Tani (2017) prepared a neural network that processes visual sequences, and another that controls the robot. By combining two neural networks through another subnetwork, they realized end-to-end training of sensory-motor integration. Our learning model has a structure similar to the model proposed by Hwang and Tani (2017), so combining two neural networks through another subnetwork might also be applicable to the proposed method.

We conducted sequence to sequence learning to allow the robot to transform each demonstrator action into robot actions. However, by giving the learning model pairs of demonstrator and robot actions that differ from the demonstrator's, sequence to sequence learning can realize translation of demonstrator actions into robot actions differing from the demonstrator's. Furthermore, we gave only one-to-one pairs of demonstrator and robot actions as training data during sequence to sequence learning. The robot can thus only imitate demonstrated actions in a single way, and cannot acquire imitative ability that performs demonstrated actions with equivalent goals but conducted by differing means, such as using both hands vs. using only one hand. Such an imitative ability is important for robots, but has not yet been realized by current methods using sequence to sequence learning. To realize this imitative ability, in future studies we should enrich training data to allow the robot to imitate demonstrated actions by various means. In the training data, the demonstrator and robot conduct equivalent actions by various means. Through training pairs of demonstrator and robot actions, the robot might come to imitate demonstrated actions in various ways. As has been found in the field of neural machine translation (Cho et al., 2014; Johnson et al., 2016), RNNs with an encoder–decoder architecture trained by sequence to sequence learning methods can acquire both syntactic and semantic structures. Thus, by applying sequence to sequence learning to action learning by robots, RNNs might allow robots to capture the underlying structures of demonstrated actions.

In this paper, imitative learning using a sequence to sequence learning method required an RNN to deal with long sequences. Therefore, RNNs other than MTRNN could be used to learn sensory-motor sequences. For example, we tried a continuous-time recurrent neural network (CTRNN) for the current experiment. Although the CTRNN generated the trained imitative patterns after predictive learning, it sometimes failed to generate untrained imitative patterns. As another example, it is well known that the long short-term memory technique (LSTM) can process long sequences because of its gating mechanisms. Thus, replacing MTRNN with LSTM will yield similar results. Although an RNN other than MTRNN could have been used, we adopted MTRNN because of its simpler representation of the internal state.

Moreover, future studies from the viewpoint of imitative learning should discuss the existence of mirror neurons (Rizzolatti et al., 1996), which by themselves show common ignition states in the imitation ability of primates with the perception of other acts and movement. This mirror neuron system has also been discussed from the viewpoint of cognitive development robotics, because human beings lead the development of behavioral understandings in others (Nagai et al., 2011; Arie et al., 2012; Kawai et al., 2012). In a previous study (Nakajo et al., 2015), we realized robot acquisition of common neuronal transitions in the robot's own and other behaviors by constraint to neurons representing labeled information, but the internal states of all neurons were separated according to their own actions in this work. Therefore, as a future method for realizing neuron activity simulating mirror neurons, it is conceivable to consider an imitation experiment using a group of neurons with slow response speeds in the context layer of the RNN.

AUTHOR CONTRIBUTIONS

RN, SM, HA, and TO conceived, designed the research, and wrote the paper. RN performed the experiment and analyzed the data.

FUNDING

This work was supported by the JST, CREST Grant Number JPMJCR15E3, and JSPS KAKENHI Grant Numbers 15H01710 and 16H05878.

REFERENCES

Arie, H., Arakaki, T., Sugano, S., and Tani, J. (2012). Imitating others by composition of primitive actions: A neuro-dynamic model. *Rob. Auton. Syst.* 60, 729–741. doi: 10.1016/j.robot.2011.11.005

Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., et al. (2009). Cognitive developmental robotics: a survey. *IEEE Trans. Auton. Mental Dev.* 1, 12–34. doi: 10.1109/TAMD.2009.2021702

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). "Robot programming by demonstration," in *Handbook of Robotics*, eds B. Siciliano and O. Khatib (Berlin; Heidelberg: Springer) 1371–1394.

Billard, A., Epars, Y., Calinon, S., Schaal, S., and Cheng, G. (2004). Discovering optimal imitation strategies. *Rob. Auton. Syst.* 47, 69–77. doi: 10.1016/j.robot.2004.03.002

Calinon, S., Florent, D., Sauser, E. L., Caldwell, D. G., and Billard, A. G. (2010). Learning of gestures by imitation A probabilistic approach based on dynamical systems. *Rob. Autom. Mag. IEEE* 17, 44–54. doi: 10.1109/MRA.2010.936947

Cangelosi, A., Metta, G., Sagerer, G., Nolfi, S., Nehaniv, C., Fischer, K., et al. (2010). Integration of action and language knowledge: a roadmap for developmental robotics. *IEEE Trans. Auton. Mental Dev.* 2, 167–195. doi: 10.1109/TAMD.2010.2053034

Cho, K., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (Doha), 1724–1734.

Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. (2016). "Deep spatial autoencoders for visuomotor learning," in *Proceedings-IEEE International Conference on Robotics and Automation* (Stockholm), 512–519.

Guenter, F., Hersch, M., Calinon, S., and Billard, A. (2007). Reinforcement Learning for Imitating Constrained Reaching Movements. *Adv. Rob.* 21, 1521–1544. doi: 10.1163/156855307782148550

Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313, 504–507. doi: 10.1126/science.1127647

Hwang, J., and Tani, J. (2017). Seamless integration and coordination of cognitive skills in humanoid robots: a deep learning approach. *IEEE Trans. Cogn. Dev. Syst.* 10, 345–358. doi: 10.1109/TCDS.2017.2714170

Inamura, T., Toshima, I., Tanie, H., and Nakamura, Y. (2004). Embodied symbol emergence based on mimesis theory. *Int. J. Rob. Res.* 23, 363–377. doi: 10.1177/0278364904042199

Ito, M., and Tani, J. (2004). On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behav.* 12, 93–115. doi: 10.1177/105971230401200202

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., et al. (2016). Google's multilingual neural machine translation system: enabling zero-shot translation. *arXiv preprint arXiv:1611.04558.*

Kawai, Y., Nagai, Y., and Asada, M. (2012). "Perceptual development triggered by its self-organization in cognitive learning," in *IEEE International Conference on Intelligent Robots and Systems* (Algarve), 5159–5164.

Kingma, D. P., and Ba, J. L. (2015). "Adam: a Method for Stochastic Optimization," in *International Conference on Learning Representations 2015* (San Diego, CA), 1–15.

Kober, J., and Peters, J. (2010). Imitation and reinforcement learning. *IEEE Rob. Autom. Mag.* 17, 55–62. doi: 10.1109/MRA.2010.936952

Krizhevsky, A., Hinton, G. E., and Sutskever, I. (2012). "ImageNet classification with deep convolutional neural networks," in *the Neural Information Processing Systems Foundation 2012 Conference* (Lake Tahoe, NV), 1–9.

Kuniyoshi, Y., Inaba, M., and Inoue, H. (1994). Learning by watching: extracting reusable task knowledge from visual observation of human performance. *IEEE Trans. Rob. Autom.* 10, 799–822. doi: 10.1109/70.338535

LeCun, Y., Bengio, Y., and Hinton, G. E. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Liu, Y., Gupta, A., and Levine, S. (2017). "Imitation from Observation: learning to imitate behaviors from raw video via context translation," in (*Conference on Neural Information Processing Systems*) (Long Beach, CA).

Lopes, M., Melo, F., Montesano, L., and Santos-Victor, J. (2010). Abstraction levels for robotic imitation: overview and computational approaches. *Stud. Comput. Int.* 264, 313–355. doi: 10.1007/978-3-642-05181-4_14

Masci, J., Meier, U., Cireşan, D., and Schmidhuber, J. (2011). "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 6791 (LNCS), 52–59.

Miyamoto, H., and Kawato, M. (1998). A tennis serve and upswing learning robot based on bi-directional theory. *Neural Netw.* 11, 1331–1344. doi: 10.1016/S0893-6080(98)00062-8

Miyamoto, H., Schaal, S., Gandolfo, F., Gomi, H., Koike, Y., Osu, R., et al. (1996). A kendama learning robot based on bi-directional theory. *Neural Netw.* 9, 1281–1302. doi: 10.1016/S0893-6080(96)00043-3

Nagai, Y., Kawai, Y., and Asada, M. (2011). "Emergence of mirror neuron system: Immature vision leads to self-other correspondence," in *2011 IEEE International Conference on Development and Learning, ICDL 2011* (Frankfurt).

Nakajo, R., Murata, S., Arie, H., and Ogata, T. (2015). "Acquisition of viewpoint representation in imitative learning from own sensory-motor experiences,". in *5th Joint International Conference on Development and Learning and Epigenetic Robotics, ICDL-EpiRob 2015* (Providence, RI), 326–331.

Ogata, T., Yokoya, R., Tani, J., Komatani, K., and Okuno, H. G. (2009). "Prediction and imitation of other's motions by reusing own forward-inverse model in robots," in *Proceedings-IEEE International Conference on Robotics and Automation* (Kobe), 4144–4149.

Rizzolatti, G., Fadiga, L., Gallese, V., and Fogassi, L. (1996). Premotor cortex and the recognition of motor actions. *Cogn. Brain Re.* 3, 131–141. doi: 10.1016/0926-6410(95)00038-0

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Chapter 8, (Cambridge, MA: MIT Press), 318–362.

Schaal, S. (1997). "Learning from demonstration," in *Advances in Neural Information Processing Systems* (Denver, Co), 1040–1046.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., et al. (2017). Time-contrastive networks: self-supervised learning from video. *arXiv preprint arXiv:1704.06888.*

Stadie, B. C., Abbeel, P., and Sutskever, I. (2017). "THIRD PERSON IMITATION LEARNING," in *ICLR 2017* (Toulon), 1–12.

Sugiura, K., Iwahashi, N., Kashioka, H., and Nakamura, S. (2010). "Statistical imitation learning in sequential object manipulation tasks," in *Advances in Robot Manipulators*, ed Ernest Hall (Rijeka: InTech), 589–606.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems (NIPS)* (Montreal, QC), 3104–3112.

Yamada, T., Murata, S., Arie, H., and Ogata, T. (2016). Dynamical integration of language and behavior in a recurrent neural network for human-Robot interaction. *Front. Neurorob.* 10:5. doi: 10.3389/fnbot.2016.00005

Yamashita, Y., and Tani, J. (2008). Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. *PLoS Comput. Biol.* 4:e1000220. doi: 10.1371/journal.pcbi.1000220

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 08 November 2018
doi: 10.3389/fnbot.2018.00073

Neural-Dynamic Based Synchronous-Optimization Scheme of Dual Redundant Robot Manipulators

Zhijun Zhang, Qiongyi Zhou and Weisen Fan*

School of Automation Science and Engineering, South China University of Technology, Guangzhou, China

In order to track complex-path tasks in three dimensional space without joint-drifts, a neural-dynamic based synchronous-optimization (NDSO) scheme of dual redundant robot manipulators is proposed and developed. To do so, an acceleration-level repetitive motion planning optimization criterion is derived by the neural-dynamic method twice. Position and velocity feedbacks are taken into account to decrease the errors. Considering the joint-angle, joint-velocity, and joint-acceleration limits, the redundancy resolution problem of the left and right arms are formulated as two quadratic programming problems subject to equality constraints and three bound constraints. The two quadratic programming schemes of the left and right arms are then integrated into a standard quadratic programming problem constrained by an equality constraint and a bound constraint. As a real-time solver, a linear variational inequalities-based primal-dual neural network (LVI-PDNN) is used to solve the quadratic programming problem. Finally, the simulation section contains experiments of the execution of three complex tasks including a couple task, the comparison with pseudo-inverse method and robustness verification. Simulation results verify the efficacy and accuracy of the proposed NDSO scheme.

Keywords: dual-redundant-manipulators, redundant robot, complex tasks, motion planning, acceleration-level, neural dynamic method

OPEN ACCESS

Edited by:
Hong Qiao,
University of Chinese Academy of Sciences (UCAS), China

Reviewed by:
Bolin Liao,
Jishou University, China
Ning Sun,
Nankai University, China

****Correspondence:***
Zhijun Zhang
drzhangzhijun@gmail.com

Received: *21 November 2017*
Accepted: *22 October 2018*
Published: *08 November 2018*

Citation:
Zhang Z, Zhou Q and Fan W (2018) Neural-Dynamic Based Synchronous-Optimization Scheme of Dual Redundant Robot Manipulators. Front. Neurorobot. 12:73. doi: 10.3389/fnbot.2018.00073

1. INTRODUCTION

Redundancy resolution problem is an important issue in the control of redundant robot manipulators. The redundancy of the robot manipulators endows us with extra degrees-of-freedom to finish some subtasks in addition to the end-effector main task (Jin and Li, 2016; Reynoso-Mora et al., 2016; Guo et al., 2017; Huang et al., 2017). Control of dual-redundant-manipulators is more complex because they have twice degrees-of-freedom than a single-redundant manipulator does. With more and redundant degrees-of-freedom, dual-redundant-manipulators can not only complete the main task of the end-effectors, but also finish various subtasks, such as joint-limitation avoidance, obstacle avoidance, singularity avoidance, and dual-arms cooperations (Zhang et al., 2014; Liu et al., 2015; Jin et al., 2017; Chikhaoui et al., 2018).

For each manipulator of the dual-redundant-robot-manipulators, since the number n of degrees-of-freedom of joints is greater than the dimension m of end-effectors' position and posture, solutions to the inverse kinematic problem of each manipulator as same as dual-manipulators are infinite (i.e., the multiple-solution problem). In order to solve such a multiple-solution problem, a number of methods have been proposed (Chevallereau and Khalil, 1988; Jin and Zhang, 2014; Toshani and Farrokhi, 2014; Luo et al., 2017). The conventional method is the pseudo-inverse formulation $\dot{\theta} = J^{+}\dot{r} + (I - J^{+}J)z_v$ or $\ddot{\theta} = J^{+}(\ddot{r} - \dot{J}\dot{\theta}) + (I - J^{+}J)z_a$, which contains a specific minimum-norm solution plus a homogeneous solution (Lin and Zhang, 2013). The pseudo-inverse method has a simple form and has been applied to dual-redundant-manipulators (Zheng and Luh, 1986), but it has to compute the matrix inverse which may have high computational cost (Ho et al., 2005), algorithm singularities and have difficulty in containing $z_v, z_a \in R^n$ into inequality form. That is to say, it cannot solve inequality constrain problems (Cheng et al., 1994). What's worse, the determining the magnitude of z_v and z_a is based on trial-and-error approach and is over-dependent on subjective judgement and experience (Zhang et al., 2004). Although some improved pseudo-inverse methods have been developed in recent years, such as joint torque optimization (Flacco and De Luca, 2015; Wang et al., 2015; Xiao et al., 2016), but it still cannot solve the inequality problems.

A repetitive motion is a basic requirement of redundant-robot-manipulators in practical applications if they are expected to execute cyclic tasks. A repetitive motion is that when the end-effector tracks a closed path in Cartesian space, all the joint trajectories should be closed. That is to say, the final states of joints must coincide with the initial ones when the end-effector completes a closed end-effector path. If this issue is not considered into the motion planning scheme of dual-redundant-manipulators, the joint-drift phenomenon would happen. In order to realize repetitive motions, additional self-motion strategy is necessary to readjust the joints of dual-manipulators to the initial states at the end of each cycle. Evidently, this is much inefficient and is not acceptable in a factory automation assembly line. Klein firstly studied this problem in a single redundant-robot-manipulator, and his research showed that the joint-drift that occurred in the pseudo-inverse control scheme is not unpredictable (Klein and Kee, 1989). In the last two decades, in order to solve the joint-drift problem, many quadratic-programming-based repetitive motion planning schemes have been proposed and solved by neural networks but most of them are about the single redundant robot manipulator (Zhang et al., 2008, 2018; Zhang and Zhang, 2012, 2013b). The control methodology of dual-redundant manipulators is imperative, as there are more and more complex end-effector tasks, such as unscrewing caps (Felip and Morales, 2015), grasping and moving of an object (Shin and Kim, 2015; Dong et al., 2017). These tasks can not be completed by a single manipulator and need dual-robot-manipulators. In recent years, some researchers have proposed impedance and admittance control methods to dual-arms coordination. For example, Lee et al. (2014) and Jr and Roberts (2015) proposed a novel relative impedance control based on the relative Jacobian expression. These works more focus on dual-arms cooperation and allocating task through force/torque, and the force/torque sensors are necessary. In fact, some tasks only need dual-manipulators synchronous working and cooperation. For instance, moving a heavy box. To finish these tasks, some researchers exploited quadratic-programming-based repetitive motion planning scheme for dual-redundant-manipulators and then used neural network as a quadratic programming solver. In our previous work, a neural dynamic method based repetitive motion planning scheme was proposed for humanoid robot arms (Zhang et al., 2015), but it is on velocity-level and cannot consider the joint-acceleration limits. In addition, the velocity-level repetitive motion planning scheme can not be directly applied to acceleration controlled robots. Jin and Zhang proposed a repetitive motion planning scheme at acceleration level (Jin and Zhang, 2014). However, the scheme is only performed on dual-manipulators with simple planar three links, and the end-effector tasks are very simple. It is worth pointing out that very few acceleration-level repetitive motion planning schemes take position-error feedback into consideration to make the position-error convergent as time involves.

The studying motivations of this paper can be summarized as: 1) A repetitive motion is a basic requirement of redundant-robot-manipulators in practical applications. 2) Most researches on the repetitive motion planning are based on a single-manipulator with less degrees-of-freedom, and very few researches considered the synchronous-optimization scheme of dual redundant robot manipulators. 3) The traditional resolution scheme at the velocity level cannot consider the acceleration limit avoidance, which may lead to acceleration limitation exceeded problem. In order to resolve the redundancy problem of dual-redundant-robot-manipulators with 14 degrees-of-freedom, a neural-dynamic based synchronous-optimization scheme of dual redundant robot manipulators (NDSO) is proposed in this paper. Different from the existing work (Jin and Zhang, 2014), the proposed NDSO scheme can be performed on dual-redundant-manipulators with 14 degrees-of-freedom and working in three-dimensional space. In addition, the dual-redundant-manipulators can track some complex paths (such as geometric curves and numbers) and complete coupled tracking task. Furthermore, the NDSO scheme has excellent robustness under the perturbation of systematic error.

The remainder of the paper is organized into four sections. In section 2, the neural-dynamic based synchronous-optimization subschemes (Sub-NDSO) of the left and right manipulators are formulated. In section 3, the Sub-NDSO of the left and right manipulators are unified into a standard quadratic programming problem, which is equivalent to a piecewise-linear projection equation, and then solved by a linear variational inequalities-based primal-dual neural network (LVI-PDNN). Section 4 shows the simulation result that the NDSO scheme performed on dual-redundant-manipulators to track three complex end-effector tasks in three-dimensional space. Comparison experiments and robustness verification experiment with perturbed LVI-PDNN are also conducted and the related results are showed in this section. Section 5 concludes this paper with final remarks.

The main contributions of the paper are as follows.

(1) A neural-dynamic based synchronous-optimization scheme of dual redundant robot manipulators (NDSO) is proposed to solve the joint-drift phenomena at the joint-acceleration

level. The advantage of the NDSO scheme is that it can not only complete the traditional end-effector tasks but also some couple tasks. In addition, the physical limit constraints allow the scheme to apply to actual situations because it guarantees the robot joints not to exceed their physical limits. In addition, it is easier than velocity-level scheme to conduct such a scheme on an acceleration/torque controlled manipulator.

(2) The NDSO scheme works for dual-robot-manipulator system, which has twice degrees-of-freedom than the same model single-robot-manipulator and thus has better coordination and flexibility compared with a single robot manipulator. Evidently, the dual-redundant-manipulator with the NDSO scheme can complete more complex and heavy tasks. It is convenient to make adjustment to the original scheme through changing the definition of matrixes in order to achieve better results because the scheme is based on a standard quadratic programming form.

(3) Three complex end-effector tasks, i.e., a pentagram tracking, a number "47" writing and a couple task, are completed by three-dimensional dual-redundant-manipulators, which validate the efficiency and accuracy of the proposed NDSO scheme.

(4) The simulation experiment verifies the robustness of the NDSO scheme with the perturbation of the systematic error. That means the proposed scheme will have strong capacity of anti-disturbance considering practical scenarios.

Before ending this section, the system structure of the scheme can be seen from **Figure 1**. First of all, the performance indices of the left and right arms are obtained by using neural dynamic method twice. Next, considering the position and velocity error, joint-angle, joint-velocity and joint-acceleration limits, the repetitive motion planning subschemes of left and right arms are constructed. Furthermore, by combining the repetitive motion planning subschemes of left and right arms, the NDSO scheme is obtained, which is further unified into a standard quadratic programming problem. The quadratic programming problem (i.e., QP in the figure) is equivalent to a set of linear variational inequalities problem (i.e., LVI in the figure) and is finally equivalent to a piecewise linear projection equation (i.e., PLPE in the figure). Finally, the piecewise linear projection equation is solved by a linear variational inequalities-based primal-dual neural network (LVI-PDNN).

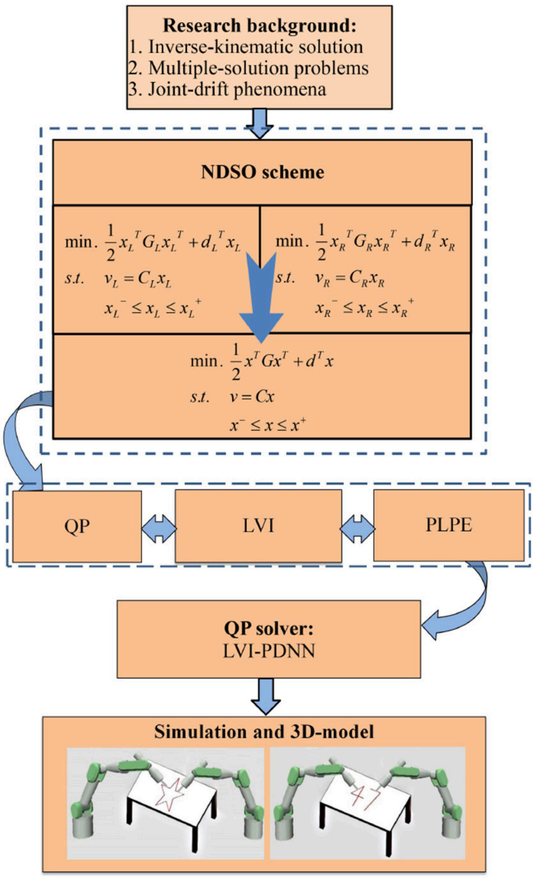

Research background:
1. Inverse-kinematic solution
2. Multiple-solution problems
3. Joint-drift phenomena
NDSO scheme
$\min.\ \frac{1}{2}x_L^T G_L x_L^T + d_L^T x_L$
$s.t.\ \ v_L = C_L x_L$
$x_L^- \le x_L \le x_L^+$
$\min.\ \frac{1}{2}x_R^T G_R x_R^T + d_R^T x_R$
$s.t.\ \ v_R = C_R x_R$
$x_R^- \le x_R \le x_R^+$
$\min.\ \frac{1}{2}x^T G x^T + d^T x$
$s.t.\ \ v = Cx$
$x^- \le x \le x^+$
QP
LVI
PLPE
QP solver:
LVI-PDNN
Simulation and 3D-model

FIGURE 1 | System structure of the neural-dynamic based synchronous-optimization scheme of dual redundant robot manipulators (NDSO). It visualizes the logical structure of the paper starting from background analysis, then the problem formulation and finally the simulation.

2. PROBLEM FORMULATION

In this section, a forward kinematic equation is first presented. Next, an acceleration-level feedback is designed. Third, an acceleration-level repetitive motion criterion is deduced by neural dynamic method two times.

2.1. Preliminaries

For simplicity, we use the subscript L/R to represent the left and right redundant manipulators. The kinematic equations of the left or right arm of the dual-redundant-manipulators at position level, velocity level and acceleration level are formulated respectively as

$$f_{\mathrm{L/R}}(\theta_{\mathrm{L/R}}) = r_{\mathrm{L/R}}(t) \tag{1}$$

$$J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\dot{\theta}_{\mathrm{L/R}}(t) = \dot{r}_{\mathrm{L/R}}(t) \tag{2}$$

$$J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\ddot{\theta}_{\mathrm{L/R}}(t) = \ddot{r}_{\mathrm{aL/R}}(t) = \ddot{r}_{\mathrm{L/R}}(t) - \dot{J}_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\dot{\theta}_{\mathrm{L/R}}(t) \tag{3}$$

where $r_{\mathrm{L/R}}(t), \dot{r}_{\mathrm{L/R}}(t)$, and $\ddot{r}_{\mathrm{L/R}}(t) \in R^m$ denote the position-and-orientation vector, velocity vector, and acceleration vector of an end-effector, $\theta_{\mathrm{L/R}}(t)$, $\dot{\theta}_{\mathrm{L/R}}(t)$, and $\ddot{\theta}_{\mathrm{L/R}}(t) \in R^n$ denote the joint angle, joint velocity and joint acceleration of the left or right manipulator, Jacobian matrix $J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}}) = \partial f_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})/\partial\theta_{\mathrm{L/R}}$, matrix $\dot{J}_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})$ is the first order derivation of Jacobian matrix $J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})$ with respect to time t. In this paper, since one manipulator has seven degrees-of-freedom and the task is performed in a three dimensional space, $n = 7$ and $m = 3$. In Equation (1), $\theta_{\mathrm{L/R}}(t)$ and $r_{\mathrm{L/R}}(t)$ are related via a nonlinear function $f_{\mathrm{L/R}}(\cdot)$. If $\theta_{\mathrm{L/R}}(t)$ is known, it is easy to compute $r_{\mathrm{L/R}}(t)$ since $f_{\mathrm{L/R}}(\cdot)$ can be uniquely determined by a given redundant robot manipulator. This process is called a forward kinematic resolution. On the contrary, it is very difficult to compute directly $\theta_{\mathrm{L/R}}(t)$ if $r_{\mathrm{L/R}}(t)$ is known because it is difficult to obtain the inverse function $f_{\mathrm{L/R}}^{-1}(\cdot)$ of nonlinear function $f_{\mathrm{L/R}}(\cdot)$. That is to say, an inverse kinematic problem of a redundant robot manipulator (or termed redundancy problem) is a difficult problem.

Remark: In practical systems, the control inputs are sometimes subject to the saturation problem and uncertainties. Many methods have been proposed to solve the issues such as (Tran et al., 2015; Eremin and Shelenok, 2017; Sun et al., 2017, 2018). Since we only focus on the redundancy resolution problem and it is assumed that the control inputs satisfy the condition, the saturation problem and uncertainties are out of our research scope, and are ignored here.

2.2. Acceleration-Level Forward Equation With Feedback

In practical applications, error feedback should be considered in Equation (3). With the following theorem, the acceleration-level forward equation with feedback is obtained, i.e.,v

Theorem 1. *Considering an end-effector motion of a robot manipulator, for any scalar parameters $\rho_V > 0$ and $\rho_P > 0$, the error-feedback included acceleration-level forward kinematic equation is*

$$J(\theta)\ddot{\theta}(t) = \ddot{r}_d(t) - \dot{J}(\theta)\dot{\theta}(t) + \rho_V(\dot{r}_d(t) - J(\theta)\dot{\theta}(t)) + \rho_P(r_d(t) - f(\theta)), \tag{4}$$

where r_d, $\dot{r}_d$, and $\ddot{r}_d$ denote desired end-effector path, desired end-effector velocity, and desired end-effector acceleration, respectively; θ, $\dot{\theta}$, and $\ddot{\theta}$ denote the joint-angular variable, joint-velocity variable, and joint-acceleration variable; Function $f(\theta)$ is a continuous nonlinear mapping function with known parameters for a given robot; $J(\theta)$ and $\dot{J}(\theta)$ are the Jacobian matrix and the first order derivative of Jacobian matrix; parameters $\rho_V > 0$ and $\rho_P > 0$ are the feedback coefficients of velocity and position errors, respectively. With these error feedbacks, the end-effector position error would converge exponentially to zero.

***Proof 1**: Considering the following state-equations of two dimensional linear system

$$\dot{\chi}(t) = A\chi(t), \tag{5}$$

$$y(t) = Q\chi(t), \tag{6}$$

where $\chi(t) = [\chi_1(t), \chi_2(t)]^{\mathrm{T}}$ is the the state vector consisting of two state variables as its elements; $\dot{\chi}(t) = [\dot{\chi}_1(t), \dot{\chi}_2(t)]^{\mathrm{T}}$ is the time derivative of the state vector $\chi(t)$; $y(t) = [y_1(t)]$ is an output vector consisting of two outputs as its elements, and A and Q are the coefficient matrices.

In order to make the position error converge to zero at the end of each cycle, an error function $E_f(t)$ is defined as

$$E_f(t) = r_{\mathrm{dL/R}}(t) - f(\theta_{\mathrm{L/R}}) \tag{7}$$

where $r_{\mathrm{dL/R}}(t)$ denotes the desired end-effector path. Its first-order and second-order derivative with time t (i.e., the velocity error $\dot{E}_f$ and acceleration error $\ddot{E}_f$) are

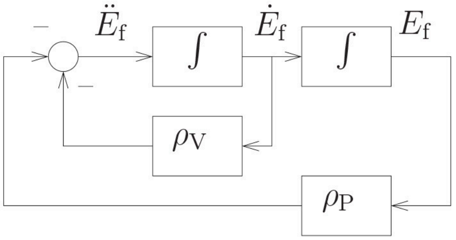


FIGURE 2 | The simulation diagram of the position and velocity feedback.

$$\dot{E}_f(t) = \dot{r}_{\mathrm{dL/R}}(t) - J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\dot{\theta}_{\mathrm{L/R}}(t), \tag{8}$$

$$\ddot{E}_f(t) = \ddot{r}_{\mathrm{dL/R}}(t) - \dot{J}_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\dot{\theta}_{\mathrm{L/R}}(t) - J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\ddot{\theta}_{\mathrm{L/R}}(t) \tag{9}$$

respectively.

For the convenience of analysis, the state variables χ_1 and χ_2 are set as E_f and $\dot{E}_f$, respectively, i.e.,

$$\chi = \begin{bmatrix} E_f \\ \dot{E}_f \end{bmatrix}, \dot{\chi} = \begin{bmatrix} \dot{E}_f \\ \ddot{E}_f \end{bmatrix}. \tag{10}$$

In addition, by defining

$$A = \begin{bmatrix} 0 & 1 \\ -\rho_{\mathrm{P}} & -\rho_{\mathrm{V}} \end{bmatrix} \text{ and } Q = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

with $\rho_{\mathrm{V}} > 0$ and $\rho_{\mathrm{P}} > 0$, the state-equations (5) and (6) are equivalent to the following second order differential equation

$$\ddot{E}_f = -\rho_{\mathrm{V}}\dot{E}_f - \rho_{\mathrm{P}}E_f \tag{11}$$

where $\rho_{\mathrm{V}} > 0$ and $\rho_{\mathrm{P}} > 0$ are the feedback coefficients of velocity and position errors, respectively. **Figure 2** shows the simulation diagram of the position and velocity feedback based on Equation (11). Substituting (7)–(9) into (11), we obtain

$$J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\ddot{\theta}_{\mathrm{L/R}}(t) = \ddot{r}_{\mathrm{afL/R}}(t) = \ddot{r}_{\mathrm{dL/R}}(t) - \dot{J}_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\dot{\theta}_{\mathrm{L/R}}(t) + \rho_{\mathrm{V}}(\dot{r}_{\mathrm{dL/R}} - J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\dot{\theta}_{\mathrm{L/R}}(t)) + \rho_{\mathrm{P}}(r_{\mathrm{dL/R}}(t) - f(\theta_{\mathrm{L/R}})). \tag{12}$$

Equation (4) is thus proved.

Next, we will prove the exponential convergence performance of the position errors $E_f(t)$. According to modern control theory (Tewari, 2002), characteristic roots ϱ_1 and ϱ_2 of the system matrix A can be obtained by solving the following characteristic equation

$$\left|\varrho I - A\right| = \left|\begin{bmatrix} \varrho & -1 \\ \rho_{\mathrm{P}} & \varrho + \rho_{\mathrm{V}} \end{bmatrix}\right| = \varrho^2 + \rho_{\mathrm{V}}\varrho + \rho_{\mathrm{P}} = 0, \tag{13}$$

where I is an identity matrix, $|\cdot|$ is the determinant notation, and ϱ is the characteristic root of Equation (13), which is determined by the coefficients ρ_{P} and ρ_{V} of characteristic Equation (13).

Since the position error $E_f(t)$ and the velocity error $\dot{E}_f(t)$ are the elements of state vector $\chi(t)$, discussion of the time-domain response of the state variables $\chi(t)$ is equivalent to discussion of errors. Based on modern control theory (Tewari, 2002), if the initial state $\chi(0) = \chi_0$ is determined, the unique solution of the state-equation (5) can be represented as

$$\chi(t) = \Phi(t)\chi(0), \tag{14}$$

where $\Phi(t) = e^{At}$.

Considering $A = [0, 1; -\rho_{\mathrm{P}}, -\rho_{\mathrm{V}}]$, based on characteristic Equation (13), the time-domain response of the state variables $\chi(t)$ (i.e., Equation 14) can be discussed according to the following three situations.

From the formula of root, we have the characteristic roots of Equation (13) as

$$\varrho_1 = \frac{-\rho_{\mathrm{V}} + \sqrt{\rho_{\mathrm{V}}^2 - 4\rho_{\mathrm{P}}}}{2}, \varrho_2 = \frac{-\rho_{\mathrm{V}} - \sqrt{\rho_{\mathrm{V}}^2 - 4\rho_{\mathrm{P}}}}{2}. \tag{15}$$

(i) When $\rho_{\mathrm{V}}^2 > 4\rho_{\mathrm{P}}$, from Equation (15), we have $\rho_{\mathrm{V}} > \sqrt{(\rho_{\mathrm{V}}^2 - 4\rho_{\mathrm{P}})} > 0$, thus real characteristic roots $\varrho_1 < 0$ and $\varrho_2 < 0$. Based on modern control theory (Tewari, 2002), there exists a nonsingular matrix T satisfying

$$\Phi(t) = T \begin{bmatrix} e^{\varrho_1 t} & 0 \\ 0 & e^{\varrho_2 t} \end{bmatrix} T^{-1}. \tag{16}$$

Substituting (16) into (14), we obtain that

$$\|\chi(t)\|_2 = \|\Phi(t)\chi(0)\|_2 \leqslant \|\Phi(t)\|_F\|\chi(0)\|_2 \leqslant \|T\|_F\|T^{-1}\|_F\sqrt{e^{2\varrho_1 t} + e^{2\varrho_2 t}}\|\chi(0)\|_2$$

is globally exponentially convergent to zero since $\|T\|_F$ and $\|T^{-1}\|_F$ are limited. Therefore, the first element of $\chi(t)$, i.e., position error $E_f(t)$, is globally exponentially convergent to zero.

(ii) When $\rho_{\mathrm{V}}^2 = 4\rho_{\mathrm{P}}$, from Equation (15) we have real equal characteristic roots $\varrho_1 = \varrho_2 = \varrho_e = -\rho_{\mathrm{V}}/2 < 0$. Based on modern control theory (Tewari, 2002), there exists a nonsingular matrix T satisfying

$$\Phi(t) = T \begin{bmatrix} e^{\varrho_e t} & te^{\varrho_e t} \\ 0 & e^{\varrho_e t} \end{bmatrix} T^{-1}. \tag{17}$$

Substituting (17) into (14), we obtain that

$$\|\chi(t)\|_2 = \|\Phi(t)\chi(0)\|_2 \leqslant \|\Phi(t)\|_F\|\chi(0)\|_2 \leqslant \|T\|_F\|T^{-1}\|_F\sqrt{t^2 + 2}e^{\varrho_e t}\|\chi(0)\|_2$$

is globally exponentially convergent to zero. Therefore, the first element of $\chi(t)$, i.e., position error $E_f(t)$, is globally exponentially convergent to zero.

(iii) When $\rho_{\mathrm{V}}^2 < 4\rho_{\mathrm{P}}$, from Equation (15) we have two imaginary characteristic roots and set them as $\varrho_1 = \sigma + j\omega$ and $\varrho_2 = \sigma - j\omega$ with the real part $\sigma < 0$. Based on modern control theory (Tewari, 2002),

$$\Phi(t) = \begin{bmatrix} \cos\omega t & \sin\omega t \\ -\sin\omega t & \cos\omega t \end{bmatrix} e^{\sigma t}. \tag{18}$$

Substituting (18) into (14), we obtain that

$$\|\chi(t)\|_2 = \|\Phi(t)\chi(0)\|_2 \leqslant \|\Phi(t)\|_F\|\chi(0)\|_2 = \sqrt{2}e^{\sigma t}\|\chi(0)\|_2$$

is globally exponentially convergent to zero. Therefore, the first element of $\chi(t)$, i.e., position error $E_f(t)$, is globally exponentially convergent to zero.

In conclusion, it is proved that the position error $E_f(t)$ is globally convergent to zero with the kind of error feedback in Equation (11) where ρ_{V} and ρ_{P} are both set positive.

2.3. NDSO Subscheme of Left/Right Arm

In order to remedy the joint-angle drift problem, a neural-dynamic based synchronous-optimization subscheme (Sub-NDSO) of left/right arm (i.e., the following theorem) is proposed.

Theorem 2. *For a left or right arm of dual-redundant-manipulators, given a closed end-effector path, i.e., $r_{L/R}(T) = r_{L/R}(0)$ where T denotes a task execution period, if Equations (19)–(23) are satisfied, the left or right arm of dual-redundant-manipulators achieves repetitive motion, and the joint-drift $\theta_{L/R}(t) - \theta_{L/R}(0)$ would converge exponentially to zero. In addition, all the joint-angles, joint-velocities and joint-accelerations are constrained within their limits, i.e.,*

$$\begin{aligned}
&\textit{minimize} && \frac{1}{2}\|\ddot{\theta}_{L/R}(t) + b_{L/R}(t)\|_2^2 && (19)\\
&\textit{subject to} && J_{L/R}(\theta_{L/R})\ddot{\theta}_{L/R}(t) = \ddot{r}_{afL/R}(t) && (20)\\
& && \theta_{L/R}^- \leqslant \theta_{L/R}(t) \leqslant \theta_{L/R}^+ && (21)\\
& && \dot{\theta}_{L/R}^- \leqslant \dot{\theta}_{L/R}(t) \leqslant \dot{\theta}_{L/R}^+ && (22)\\
& && \ddot{\theta}_{L/R}^- \leqslant \ddot{\theta}_{L/R}(t) \leqslant \ddot{\theta}_{L/R}^+ && (23)\\
&\textit{with} && b_{L/R}(t) = (\alpha + \beta)\dot{\theta}_{L/R}(t) + \alpha\beta(\theta_{L/R}(t) - \theta_{L/R}(0)), &&\\
& && \ddot{r}_{afL/R}(t) = \ddot{r}_{dL/R}(t) - \dot{J}_{L/R}(\theta_{L/R})\dot{\theta}_{L/R}(t) &&\\
& && +\rho_v(\dot{r}_{dL/R}(t) - J_{L/R}(\theta_{L/R})\dot{\theta}_{L/R}(t)) + \rho_p(r_{dL/R}(t) &&\\
& && -f(\theta_{L/R})) &&
\end{aligned}$$

where $\|\cdot\|_2$ denotes the two-norm of a vector; $\theta_{L/R}$, $\dot{\theta}_{L/R}$, and $\ddot{\theta}_{L/R}$ denote the joint angle, joint velocity, and joint acceleration of the left or right arms of dual-redundant-manipulators; $r_{dL/R}$, $\dot{r}_{dL/R}$, and $\ddot{r}_{dL/R}$ denote desired end-effector position, desired end-effector velocity, and desired end-effector acceleration of the left or right arm of dual-redundant-manipulators; $J(\theta)$ and $\dot{J}(\theta)$ are the Jacobian matrix and the first order derivative of Jacobian matrix; $\alpha > 0$ and $\beta > 0$ are used to scale the joint displacement; $\theta_{L/R}^{\pm}$, $\dot{\theta}_{L/R}^{\pm}$ and $\ddot{\theta}_{L/R}^{\pm}$ denote the upper and lower limits of the joint angles, joint velocities and joint accelerations of the left/right manipulator, respectively.

Proof 2: First, a vector-valued error function, i.e., the deviation between the joint instant angle $\theta_{\mathrm{L/R}}(t)$ and the initial joint angle $\theta_{\mathrm{L/R}}(0)$ of the left/right manipulator, is defined as

$$e_{1\mathrm{L/R}}(t) = \theta_{\mathrm{L/R}}(t) - \theta_{\mathrm{L/R}}(0). \tag{24}$$

The joint-angle drift is zero if and only if the value of the error function $e_{1\mathrm{L/R}}(t) = 0$. In order to reduce and eventually eliminate the joint displacement, by the neural-dynamic method, we can obtain

$$\dot{e}_{1\mathrm{L/R}}(t) = -\alpha e_{1\mathrm{L/R}}(t) = -\alpha[\theta_{\mathrm{L/R}}(t) - \theta_{\mathrm{L/R}}(0)], \qquad (25)$$

where design parameter α is used to adjust the convergence rate of $e_{1\mathrm{L/R}}(t)$ to zero. By taking the derivative of Equation (24) with time t, $\dot{e}_{1\mathrm{L/R}}(t) = \dot{\theta}_{\mathrm{L/R}}(t)$ is obtained. Substituting it into Equation (25), the following equation is obtained, i.e.,

$$\dot{\theta}_{\mathrm{L/R}}(t) + \alpha(\theta_{\mathrm{L/R}}(t) - \theta_{\mathrm{L/R}}(0)) = 0. \qquad (26)$$

Second, in order to obtain the acceleration-level repetitive motion criterion, the joint acceleration should be included in the criterion. That is to say, there should be an equation equivalent to (26), which includes joint acceleration. To do so, the neural dynamic method is applied to Equation (26) again. Similarly, a vector-valued joint-displacement function is defined as

$$e_{2\mathrm{L/R}}(t) = \dot{\theta}_{\mathrm{L/R}}(t) + \alpha(\theta_{\mathrm{L/R}}(t) - \theta_{\mathrm{L/R}}(0)). \qquad (27)$$

According to neural dynamic design method (Cai and Zhang, 2012), i.e.,

$$\dot{e}_{2\mathrm{L/R}}(t) = -\beta e_{2\mathrm{L/R}}(t) \qquad (28)$$

where design parameter $\beta > 0$, we can get

$$\ddot{\theta}_{\mathrm{L/R}}(t) + \alpha\dot{\theta}_{\mathrm{L/R}}(t) = -\beta(\dot{\theta}_{\mathrm{L/R}}(t) + \alpha(\theta_{\mathrm{L/R}}(t) - \theta_{\mathrm{L/R}}(0))). \qquad (29)$$

Equation (29) is rewritten as

$$\ddot{\theta}_{\mathrm{L/R}}(t) + (\alpha+\beta)\dot{\theta}_{\mathrm{L/R}}(t) + \alpha\beta(\theta_{\mathrm{L/R}}(t) - \theta_{\mathrm{L/R}}(0)) = 0. \qquad (30)$$

Considering the motion of the robot manipulator, it is better to minimize the performance $\|\ddot{\theta}_{\mathrm{L/R}}(t)+(\alpha+\beta)\dot{\theta}_{\mathrm{L/R}}(t)+\alpha\beta(\theta_{\mathrm{L/R}}(t)-\theta_{\mathrm{L/R}}(0))\|_2^2/2$ rather than use (30) directly, i.e.,

$$\text{minimize} \quad \frac{1}{2}\|\ddot{\theta}_{\mathrm{L/R}}(t) + b_{\mathrm{L/R}}(t)\|_2^2, \qquad (31)$$

where $b_{\mathrm{L/R}}(t) = (\alpha+\beta)\dot{\theta}_{\mathrm{L/R}}(t) + \alpha\beta(\theta_{\mathrm{L/R}}(t) - \theta_{\mathrm{L/R}}(0))$, and $\|\cdot\|_2$ denotes the two-norm of a vector. If Equation (31) is used as the optimization criterion, the joint angle $\theta_{\mathrm{L/R}}(t)$ tends to converge to $\theta_{\mathrm{L/R}}(0)$. At the end of the task execution period, $\theta_{\mathrm{L/R}}(T) = \theta_{\mathrm{L/R}}(0)$. Equation (19) is thus proved.

In practical applications, the joint physical limits, i.e., joint-angle limits, joint-velocity limits and joint-acceleration limits, should be considered into the scheme, and thus an NDSO subschemes (termed as Sub-NDSO) is obtained as

$$\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\|\ddot{\theta}_{\mathrm{L/R}}(t) + b_{\mathrm{L/R}}(t)\|_2^2 && (32)\\
\text{subject to} \quad & J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\ddot{\theta}_{\mathrm{L/R}}(t) = \ddot{r}_{\mathrm{aL/R}}(t) && (33)\\
& \theta^-_{\mathrm{L/R}} \leqslant \theta_{\mathrm{L/R}}(t) \leqslant \theta^+_{\mathrm{L/R}} && (34)\\
& \dot{\theta}^-_{\mathrm{L/R}} \leqslant \dot{\theta}_{\mathrm{L/R}}(t) \leqslant \dot{\theta}^+_{\mathrm{L/R}} && (35)\\
& \ddot{\theta}^-_{\mathrm{L/R}} \leqslant \ddot{\theta}_{\mathrm{L/R}}(t) \leqslant \ddot{\theta}^+_{\mathrm{L/R}} && (36)\\
\text{with} \quad & b_{\mathrm{L/R}}(t) = (\alpha+\beta)\dot{\theta}_{\mathrm{L/R}}(t) + \alpha\beta(\theta_{\mathrm{L/R}}(t) - \theta_{\mathrm{L/R}}(0)) &&\\
& \ddot{r}_{\mathrm{aL/R}}(t) = \ddot{r}_{\mathrm{dL/R}}(t) - \dot{J}_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\dot{\theta}_{\mathrm{L/R}}(t) &&
\end{aligned}$$

where $\alpha > 0$ and $\beta > 0$ are used to scale the joint displacement.

According to the acceleration-level feedback error design method in Theorem 1, $\ddot{r}_{\mathrm{aL/R}}$ in Equation (33) can be replaced by $\ddot{r}_{\mathrm{afL/R}}(t) = \ddot{r}_{\mathrm{dL/R}}(t) - \dot{J}_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\dot{\theta}_{\mathrm{L/R}}(t) + \rho_{\mathrm{v}}(\dot{r}_{\mathrm{dL/R}}(t) - J_{\mathrm{L/R}}(\theta_{\mathrm{L/R}})\dot{\theta}_{\mathrm{L/R}}(t)) + \rho_{\mathrm{p}}(r_{\mathrm{dL/R}}(t) - f(\theta_{\mathrm{L/R}}))$. Equations (19)–(23) is thus proved. That is to say, with Equations (19)–(23), the left or right arm of dual-redundant-manipulators can achieve repetitive motion, meanwhile it can avoid joint-physical limits during the execution of the task.

Next, the exponential convergence rate of joint-drift $\theta_{\mathrm{L/R}}(t) - \theta_{\mathrm{L/R}}(0)$ will be proved. In the light of differential equation theory (Hartman and Philip, 1982), the ith element of $e_{2\mathrm{L/R}}(t)$ in Equation (28) is

$$e_{2\mathrm{L/Ri}}(t) = e_{2\mathrm{L/Ri}}(0)e^{-\beta t}. \qquad (37)$$

When t approaches to infinity, each element would approach exponentially zero, i.e.,

$$\lim_{t\to\infty} e_{2\mathrm{L/Ri}}(t) = \lim_{t\to\infty} e_{2\mathrm{L/Ri}}(0)e^{-\beta t} = 0. \qquad (38)$$

The proof of Theorem 2 is completed.

2.4. NDSO Scheme

In this section, based on the neural-dynamic based synchronous-optimization subschemes (Sub-NDSO) of the left arm and right arm proposed in Theorem 2, a neural-dynamic based synchronous-optimization scheme of dual redundant robot manipulators (NDSO) is proposed and developed.

Theorem 3. *For a dual-redundant-manipulators system, including left manipulator and right manipulator, given a closed end-effector path, i.e., $r(T) = r(0)$ where T denotes a task execution period, if Equations (39)–(43) are satisfied, the dual-redundant-manipulators will achieve repetitive motion, and the joint-drift $\theta(t) - \theta(0)$ would converge exponentially to zero. In addition, all the joint angles, joint velocities and joint accelerations of the dual-redundant-manipulators are constrained within their limits, i.e.,*

$$\begin{aligned}
\textit{minimize} \quad & \frac{1}{2}\ddot{\theta}^T(t)\ddot{\theta}(t) + b^T(t)\ddot{\theta}(t) && (39)\\
\textit{subject to} \quad & J(\theta)\ddot{\theta}(t) = \ddot{r}_{af}(t) && (40)\\
& \theta^- \leqslant \theta(t) \leqslant \theta^+ && (41)\\
& \dot{\theta}^- \leqslant \dot{\theta}(t) \leqslant \dot{\theta}^+ && (42)\\
& \ddot{\theta}^- \leqslant \ddot{\theta}(t) \leqslant \ddot{\theta}^+ && (43)\\
\textit{with} \quad & b(t) = (\alpha+\beta)\dot{\theta}(t) + \alpha\beta(\theta(t) - \theta(0)), &&\\
& \ddot{r}_{af}(t) = \ddot{r}_d(t) - \dot{J}(\theta)\dot{\theta}(t) + \rho_v(\dot{r}_d(t) - J(\theta)\dot{\theta}(t)) &&\\
& +\rho_p(r_d(t) - f(\theta)) &&
\end{aligned}$$

where $\theta(t) = [\theta_L(t), \theta_R(t)]^T$, $\dot{\theta}(t) = [\dot{\theta}_L(t), \dot{\theta}_R(t)]^T$, and $\ddot{\theta}(t) = [\ddot{\theta}_L(t), \ddot{\theta}_R(t)]^T$ denote the joint angle, joint velocity, and joint acceleration of the dual-redundant-manipulators; $r_d(t) = [r_{dL}(t), r_{dR}(t)]^T$, $\dot{r}_d(t) = [\dot{r}_{dL}(t), \dot{r}_{dR}(t)]^T$, and $\ddot{r}_d(t) = [\ddot{r}_{dL}(t), \ddot{r}_{dR}(t)]^T$ denote the position vector, velocity vector, and *acceleration vector of the end-effector of the dual-redundant-manipulators; Scalar parameters $\alpha > 0$ and $\beta > 0$ are used to scale the joint displacements; $\theta^{\pm} = [\theta_L^{\pm}, \theta_R^{\pm}]^T$, $\dot{\theta}^{\pm} = [\dot{\theta}_L^{\pm}, \dot{\theta}_R^{\pm}]^T$ and $\ddot{\theta}^{\pm} = [\ddot{\theta}_L^{\pm}, \ddot{\theta}_R^{\pm}]^T$ denote the upper and lower limits of the joint angles, joint velocities and joint accelerations of the dual-redundant-manipulator, respectively. The combined Jacobian matrix and the first order derivative of the combined Jacobian matrix of the dual-redundant-manipulators are*

$$J(\theta) = \begin{bmatrix} J_L(\theta_L) & \mathbf{0} \\ \mathbf{0} & J_R(\theta_R) \end{bmatrix}, \dot{J}(\theta) = \begin{bmatrix} \dot{J}_L(\theta_L) & \mathbf{0} \\ \mathbf{0} & \dot{J}_R(\theta_R) \end{bmatrix}. \quad (44)$$

Proof 3: Firstly, the optimization criterion (32) can be simplified as

$$\begin{aligned} &\frac{1}{2}\|\ddot{\theta}_{\mathrm{L/R}}(t) + b_{\mathrm{L/R}}(t)\|_2^2 \\ = &\frac{1}{2}\left(\ddot{\theta}_{\mathrm{L/R}}(t) + b_{\mathrm{L/R}}(t)\right)^{\mathrm{T}}\left(\ddot{\theta}_{\mathrm{L/R}}(t) + b_{\mathrm{L/R}}(t)\right) \\ = &\frac{1}{2}\Big(\ddot{\theta}^{\mathrm{T}}_{\mathrm{L/R}}(t)\ddot{\theta}_{\mathrm{L/R}}(t) + \ddot{\theta}^{\mathrm{T}}_{\mathrm{L/R}}(t)b_{\mathrm{L/R}}(t) + b^{\mathrm{T}}_{\mathrm{L/R}}(t)\ddot{\theta}_{\mathrm{L/R}}(t) \\ &+ b^{\mathrm{T}}_{\mathrm{L/R}}(t)b_{\mathrm{L/R}}(t)\Big) \\ = &\frac{1}{2}\ddot{\theta}^{\mathrm{T}}_{\mathrm{L/R}}(t)\ddot{\theta}_{\mathrm{L/R}}(t) + b^{\mathrm{T}}_{\mathrm{L/R}}(t)\ddot{\theta}_{\mathrm{L/R}}(t) + \frac{1}{2}b^{\mathrm{T}}_{\mathrm{L/R}}(t)b_{\mathrm{L/R}}(t). \quad (45) \end{aligned}$$

Since the redundant resolution problem is solved at the joint-acceleration level and $\ddot{\theta}_{\mathrm{L/R}}$ is the decision variable, $b^{\mathrm{T}}_{\mathrm{L/R}}(t)b_{\mathrm{L/R}}(t)/2$ in Equation (45) can be regarded as a constant. Therefore, minimizing $\|\ddot{\theta}_{\mathrm{L/R}}(t) + b_{\mathrm{L/R}}(t)\|_2^2/2 = \ddot{\theta}^{\mathrm{T}}_{\mathrm{L/R}}(t)\ddot{\theta}_{\mathrm{L/R}}(t)/2 + b^{\mathrm{T}}_{\mathrm{L/R}}(t)\ddot{\theta}_{\mathrm{L/R}}(t) + b^{\mathrm{T}}_{\mathrm{L/R}}(t)b_{\mathrm{L/R}}(t)/2$ is equivalent to minimizing $\ddot{\theta}^{\mathrm{T}}_{\mathrm{L/R}}(t)\ddot{\theta}_{\mathrm{L/R}}(t)/2 + b^{\mathrm{T}}_{\mathrm{L/R}}(t)\ddot{\theta}_{\mathrm{L/R}}(t)$. Combining the joint variables of left and right manipulators into one combined vector, the optimization criterion can be written as

$$\text{minimize} \quad \ddot{\theta}^{\mathrm{T}}(t)\ddot{\theta}(t)/2 + b^{\mathrm{T}}(t)\ddot{\theta}(t) \quad (46)$$

where $\ddot{\theta}(t) = [\ddot{\theta}_{\mathrm{L}}(t), \ddot{\theta}_{\mathrm{R}}(t)]^{\mathrm{T}}$ and $b(t) = [b_{\mathrm{L}}(t), b_{\mathrm{R}}(t)]^{\mathrm{T}}$.

Secondly, acceleration level forward kinematic Equation (20) of left and right manipulators can be written as a combined forward kinematic equation as

$$\begin{bmatrix} J_{\mathrm{L}}(\theta) & \mathbf{0} \\ \mathbf{0} & J_{\mathrm{R}}(\theta) \end{bmatrix} \cdot \begin{bmatrix} \ddot{\theta}_{\mathrm{L}}(t) \\ \ddot{\theta}_{\mathrm{R}}(t) \end{bmatrix} = \begin{bmatrix} \ddot{r}_{\mathrm{afL}}(t) \\ \ddot{r}_{\mathrm{afR}}(t) \end{bmatrix} \quad (47)$$

where

$$\begin{aligned} \ddot{r}_{\mathrm{afL}}(t) &= \ddot{r}_{\mathrm{dL}}(t) - \dot{J}_{\mathrm{L}}(\theta_{\mathrm{L}})\dot{\theta}_{\mathrm{L}}(t) \\ &+ \rho_{\mathrm{v}}(\dot{r}_{\mathrm{dL}}(t) - J_{\mathrm{L}}(\theta_{\mathrm{L}})\dot{\theta}_{\mathrm{L}}(t)) + \rho_{\mathrm{p}}(r_{\mathrm{dL}}(t) - f(\theta_{\mathrm{L}})), \quad (48) \\ \ddot{r}_{\mathrm{afR}}(t) &= \ddot{r}_{\mathrm{dR}}(t) - \dot{J}_{\mathrm{R}}(\theta_{\mathrm{R}})\dot{\theta}_{\mathrm{R}}(t) \\ &+ \rho_{\mathrm{v}}(\dot{r}_{\mathrm{dR}}(t) - J_{\mathrm{R}}(\theta_{\mathrm{R}})\dot{\theta}_{\mathrm{R}}(t)) + \rho_{\mathrm{p}}(r_{\mathrm{dR}}(t) - f(\theta_{\mathrm{R}})). \quad (49) \end{aligned}$$

Combining the upper and lower joint-limits of left and right arms of dual-redundant-manipulators, we can get combined joint-angular, joint-velocity, joint-acceleration limits respectively as

$$\theta^{\pm}(t) = [\theta^{\pm}_{\mathrm{L}}(t), \theta^{\pm}_{\mathrm{R}}(t)]^{\mathrm{T}}, \quad (50)$$

$$\dot{\theta}^{\pm}(t) = [\dot{\theta}^{\pm}_{\mathrm{L}}(t), \dot{\theta}^{\pm}_{\mathrm{R}}(t)]^{\mathrm{T}}, \quad (51)$$

$$\ddot{\theta}^{\pm}(t) = [\ddot{\theta}^{\pm}_{\mathrm{L}}(t), \ddot{\theta}^{\pm}_{\mathrm{R}}(t)]^{\mathrm{T}}. \quad (52)$$

Taking into consideration of optimization criterion (46), feedback considered acceleration-level kinematic equation (47), and joint-limits (50)–(52), NDSO scheme (40)–(43) is obtained. The proof of Theorem 3 is completed. □

TABLE 1 | Joint physical limits used in simulations.

#	θ_{L}^{-} (rad)	θ_{L}^{+} (rad)	θ_{R}^{-} (rad)	θ_{R}^{+} (rad)	$\dot{\theta}_{\mathrm{L/R}}^{-}$ (rad/s)	$\dot{\theta}_{\mathrm{L/R}}^{+}$ (rad/s)	$\ddot{\theta}_{\mathrm{L/R}}^{-}$ (rad/s^2)	$\ddot{\theta}_{\mathrm{L/R}}^{+}$ (rad/s^2)
1	−1	1	−5	0	−1.5	1.5	−6	6
2	−2	2	−2	0	−1.5	1.5	−6	6
3	−1	1	−1	1	−1.5	1.5	−6	6
4	1	3	1	3	−1.5	1.5	−6	6
5	−1	1	−1	1	−1.5	1.5	−6	6
6	−2	0	−2	0	−1.5	1.5	−6	6
7	−1	1	−1	1	−1.5	1.5	−6	6

TABLE 2 | Four sets of equations used in the three groups of contrast experiments.

	1	2	3	4
d	$0 \in R^{2n}$	$b(t) \in R^{2n}$	$b(t) \in R^{2n}$	$b(t) \in R^{2n}$
x^-	ζ^-	$-\varpi\zeta^-$	ζ^-	ζ^-
x^+	ζ^+	$\varpi\zeta^+$	ζ^+	ζ^+
ρ_{p}	1	1	0	1
ρ_{v}	200	200	0	200

3. QUADRATIC PROGRAMMING UNIFICATION & SOLVER

In this section, the proposed NDSO scheme (39)–(43) is unified into a standard quadratic programming problem, which is equivalent to linear variational inequality problem and is further equivalent to a piecewise linear projection equation. Finally,

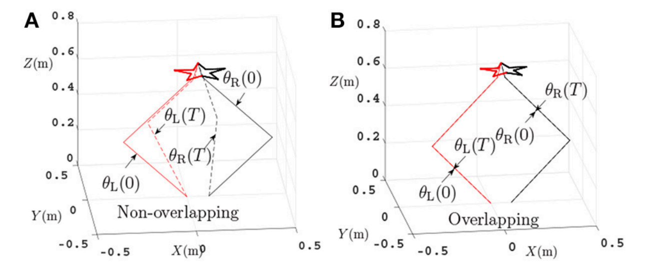


FIGURE 3 | Comparisons between the scheme without considering repetitive motion and the NDSO scheme when tracking a pentagram-path. **(A)** Final states do not coincide with the initial states when using the scheme without considering repetitive motion. **(B)** Final states coincide with initial states when using NDSO scheme considering repetitive motion.

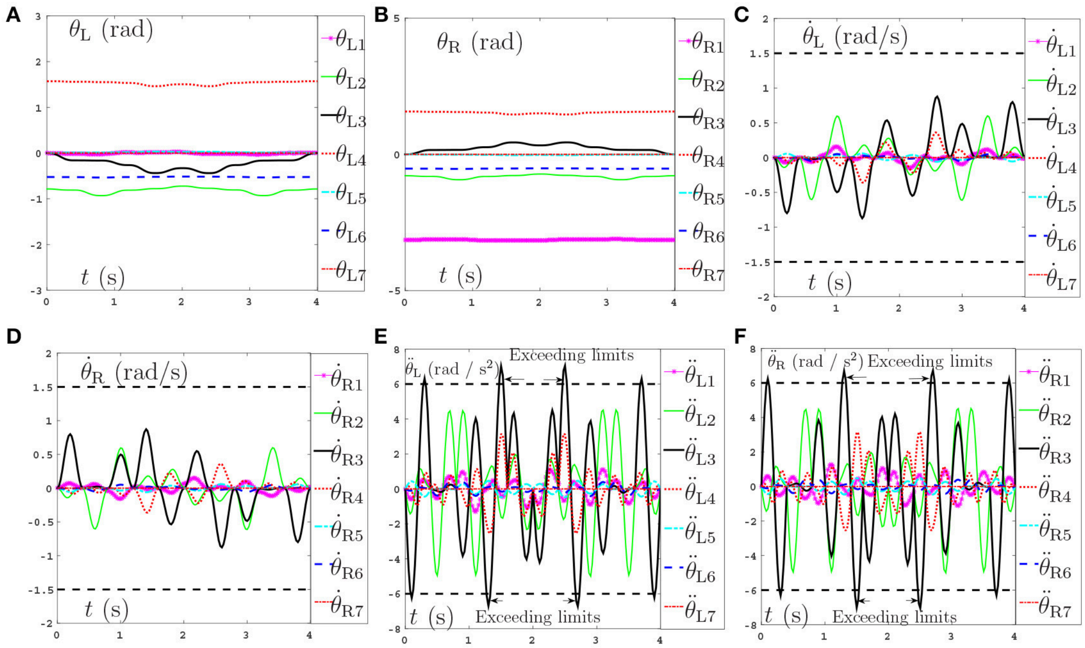


FIGURE 4 | Joint angles, joint velocities, joint accelerations during the dual-redundant manipulators tracking a pentagram path when using the scheme with considering repetitive motion planning and feedback criteria but no physical-limits criterion. **(A)** Joint angle of left arm θ_L. **(B)** Joint angle of right arm θ_R. **(C)** Joint velocity of left arm $\dot{\theta}_L$. **(D)** Joint velocity of right arm $\dot{\theta}_R$. **(E)** Joint acceleration of left arm $\ddot{\theta}_L$. **(F)** Joint acceleration of right arm $\ddot{\theta}_R$.

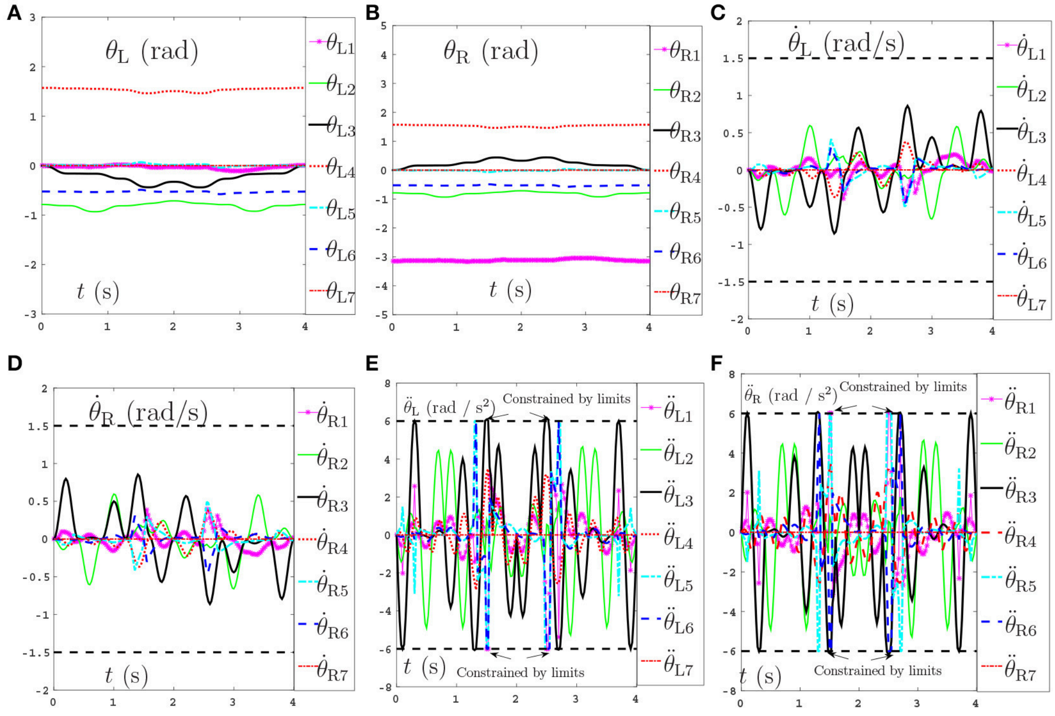


FIGURE 5 | Joint angles, joint velocities, joint accelerations during the dual-redundant manipulators tracking a pentagram path when using the NDSO scheme with considering repetitive motion planning and physical-limits and feedback criterion. **(A)** Joint angle of left arm θ_L. **(B)** Joint angle of right arm θ_R. **(C)** Joint velocity of left arm $\dot{\theta}_L$. **(D)** Joint velocity of right arm $\dot{\theta}_R$. **(E)** Joint acceleration of left arm $\ddot{\theta}_L$. **(F)** Joint acceleration of right arm $\ddot{\theta}_R$.

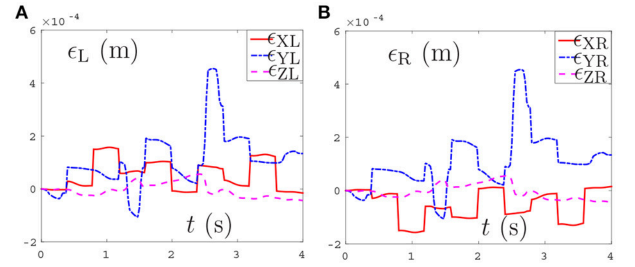


FIGURE 6 | Position errors during the period of pentagram-path tracking synthesized by NDSO scheme without considering position and velocity feedback. **(A)** Position error of left arm ϵ_{L}; **(B)** Position error of right arm ϵ_{R}.

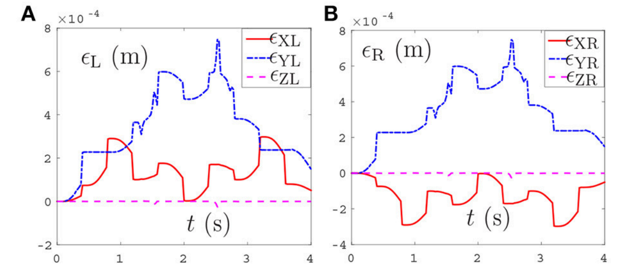


FIGURE 7 | Position errors during the period of pentagram-path tracking synthesized by NDSO scheme with considering position and velocity feedback. **(A)** Position error of left arm ϵ_{L}; **(B)** Position error of right arm ϵ_{R}.

the piecewise linear projection equation is solved by a linear variational inequalities-based primal-dual neural network (LVI-PDNN).

3.1. Joint-Limits Conversion

In order to resolve the redundancy problem at the acceleration-level and satisfy the format requirement of standard quadratic programming, physical limits (41)–(43) at different levels should be converted into one bound constraint with joint-acceleration $\ddot{\theta}(t)$. Specifically, the ith elements of bounds ζ^- and ζ^+ are defined respectively as

$$\zeta_i^-(t) = \max\{\ddot{\theta}_i^-(t), \lambda_{\mathrm{v}}(\dot{\theta}_i^- - \dot{\theta}_i(t)), \lambda_{\mathrm{p}}((\theta_i^- + \vartheta_i) - \theta_i(t))\},$$
$$\zeta_i^+(t) = \min\{\ddot{\theta}_i^+(t), \lambda_{\mathrm{v}}(\dot{\theta}_i^+ - \dot{\theta}_i(t)), \lambda_{\mathrm{p}}((\theta_i^+ - \vartheta_i) - \theta_i(t))\}.$$

Actually, there exist the inertia movement during the deceleration period caused by the mechanical inertia of the dual-redundant-manipulators in practice. Thus critical areas for joint position variables are considered into physical limits' representation so that there will appear a deceleration earlier when they enter the areas but not reach the joint position limit yet. $\vartheta_i > 0$ is a small constant and used to define the critical areas $[\theta_i^-, \theta_i^- + \vartheta_i]$ and $[\theta_i^+ - \vartheta_i, \theta_i^+]$. In the simulation section of the paper, $\vartheta_i > 0$ is set 0.01. The coefficient $\lambda_{\mathrm{v}} > 0$ and $\lambda_{\mathrm{p}} > 0$ denote the decreasing amplitude (Zhang et al., 2008).

TABLE 3 | Joint drifts when dual-redundant-manipulators tracking a pentagram-path synthesized by NDSO scheme with considering repetitive motions, joint limits, and feedback.

	Joint displacements (rad)	Joint displacements (degree)
LEFT ARM		
$\theta_{\mathrm{L}1}(4) - \theta_{\mathrm{L}1}(0)$	$+3.68924 \times 10^{-3}$	+0.21138
$\theta_{\mathrm{L}2}(4) - \theta_{\mathrm{L}2}(0)$	$+1.12535 \times 10^{-4}$	+0.00645
$\theta_{\mathrm{L}3}(4) - \theta_{\mathrm{L}3}(0)$	-8.00902×10^{-4}	−0.04589
$\theta_{\mathrm{L}4}(4) - \theta_{\mathrm{L}4}(0)$	$+6.05971 \times 10^{-5}$	+0.00347
$\theta_{\mathrm{L}5}(4) - \theta_{\mathrm{L}5}(0)$	-6.14706×10^{-3}	−0.35220
$\theta_{\mathrm{L}6}(4) - \theta_{\mathrm{L}6}(0)$	-1.44414×10^{-3}	−0.08274
$\theta_{\mathrm{L}7}(4) - \theta_{\mathrm{L}7}(0)$	0.00000	0.00000
RIGHT ARM		
$\theta_{\mathrm{R}1}(4) - \theta_{\mathrm{R}1}(0)$	-3.68924×10^{-3}	−0.21138
$\theta_{\mathrm{R}2}(4) - \theta_{\mathrm{R}2}(0)$	$+1.12535 \times 10^{-4}$	+0.00645
$\theta_{\mathrm{R}3}(4) - \theta_{\mathrm{R}3}(0)$	$+8.00902 \times 10^{-4}$	+0.04589
$\theta_{\mathrm{R}4}(4) - \theta_{\mathrm{R}4}(0)$	$+6.05971 \times 10^{-5}$	+0.00347
$\theta_{\mathrm{R}5}(4) - \theta_{\mathrm{R}5}(0)$	$+6.14706 \times 10^{-3}$	+0.35220
$\theta_{\mathrm{R}6}(4) - \theta_{\mathrm{R}6}(0)$	-1.44414×10^{-3}	−0.08274
$\theta_{\mathrm{R}7}(4) - \theta_{\mathrm{R}7}(0)$	0.00000	0.00000

Therefore, constraints (39)–(43) can be rewritten as

$$\text{minimize} \quad \frac{1}{2}\ddot{\theta}(t)^{\mathrm{T}} W \ddot{\theta}(t) + b^{\mathrm{T}}(t)\ddot{\theta}(t) \tag{53}$$
$$\text{subject to} \quad J(\theta)\ddot{\theta}(t) = \ddot{r}_{\mathrm{af}}(t) \tag{54}$$
$$\zeta^-(t) \leqslant \ddot{\theta}(t) \leqslant \zeta^+(t) \tag{55}$$

The scheme (53)–(55) can be further unified into the following standard quadratic programming

$$\text{minimize} \quad \frac{1}{2}x^{\mathrm{T}} G x + d^{\mathrm{T}} x \tag{56}$$
$$\text{subject to} \quad Cx = h \tag{57}$$
$$x^- \leqslant x \leqslant x^+ \tag{58}$$

where

$$x = \ddot{\theta}(t) = \begin{bmatrix} \ddot{\theta}_{\mathrm{L}}(t) \\ \ddot{\theta}_{\mathrm{R}}(t) \end{bmatrix} \in R^{2n}, \; G = W = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \in R^{2n \times 2n},$$
$$d = b(t) = \begin{bmatrix} b_{\mathrm{L}}(t) \\ b_{\mathrm{R}}(t) \end{bmatrix} \in R^{2n}, \; h = \ddot{r}_{\mathrm{af}}(t) = \begin{bmatrix} \ddot{r}_{\mathrm{afL}}(t) \\ \ddot{r}_{\mathrm{afR}}(t) \end{bmatrix} \in R^{2m},$$
$$C = J = \begin{bmatrix} J_{\mathrm{L}}(\theta_{\mathrm{L}}) & \mathbf{0} \\ \mathbf{0} & J_{\mathrm{R}}(\theta_{\mathrm{R}}) \end{bmatrix} \in R^{2m \times 2n}, \; x^{\pm} = \zeta^{\pm}(t) \in R^{2n}.$$

3.2. Quadratic Programming Solver

According to Zhang et al. (2008), finding the solutions to quadratic programming problem (56)–(58) is equivalent to finding out a primal-dual equilibrium vector $u^* = [x^*; \eta^*]^{\mathrm{T}} \in \Omega := \{u = [x^{\mathrm{T}}, \eta^{\mathrm{T}}]^{\mathrm{T}} \in R^{2n+2m} | u^- \leqslant u \leqslant u^+\}$ to the following linear variational inequality

$$(u - u^*)^{\mathrm{T}}(Mu^* + q) \geqslant 0, \forall u \in \Omega, \tag{59}$$

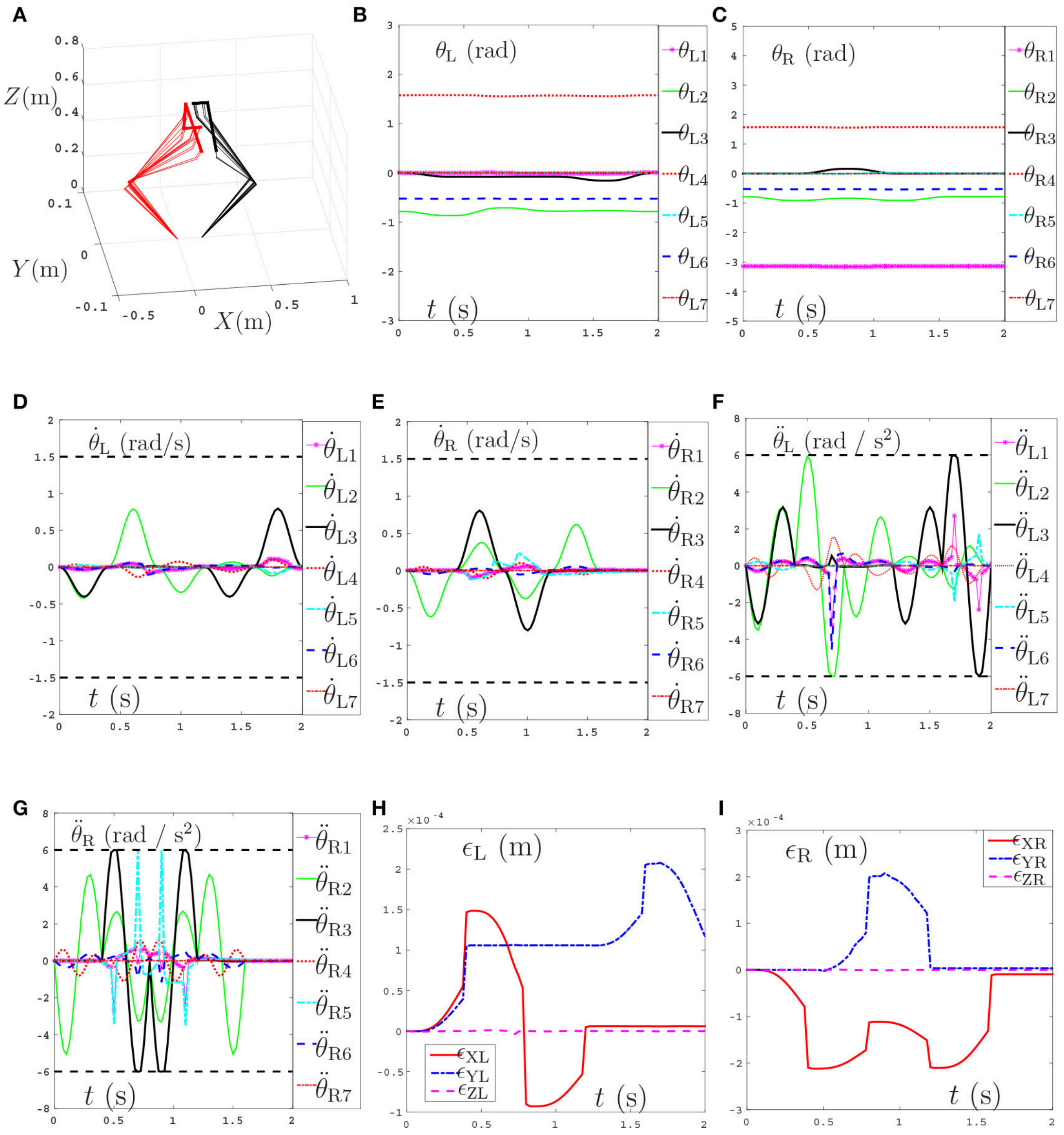


FIGURE 8 | Tracking trajectories, joint angles, joint velocities, and joint accelerations during the period of number "47" writing synthesized by the proposed NDSO scheme (39)–(43) which considers repetitive motion planning, joint limits, and feedbacks. **(A)** 3-D simulation tracking trajectories. **(B)** Left arm joint angle θ_{L}. **(C)** Right arm joint angle θ_{R}. **(D)** Left arm joint velocity $\dot{\theta}_{\mathrm{L}}$. **(E)** Right arm joint velocity $\dot{\theta}_{\mathrm{R}}$. **(F)** Left arm joint acceleration $\ddot{\theta}_{\mathrm{L}}$. **(G)** Right arm joint acceleration $\ddot{\theta}_{\mathrm{R}}$. **(H)** Position error of left arm ϵ_{L}. **(I)** Position error of right arm ϵ_{R}.

where the augmented primal-dual decision variable $u \in R^{(2n+2m)}$, and its bounds $u^{\pm} \in R^{(2n+2m)}$ are respectively defined as

$$u = \begin{bmatrix} x \\ \eta \end{bmatrix},\ u^{+} = \begin{bmatrix} x^{+} \\ 1_{v}\varpi \end{bmatrix},\ u^{-} = \begin{bmatrix} x^{-} \\ -1_{v}\varpi \end{bmatrix},$$

with $\eta \in R^{2m}$ being the corresponding dual decision vectors of Equation (57), $1_{v} = [1, \cdots, 1]^{\mathrm{T}}$ denoting an appropriately-dimensioned vector composed of ones, and $\varpi = 10^{10} \in R$ replacing the $+\infty$ for simulation and implementation purposes. The matrix $M \in R^{(2n+2m)\times(2n+2m)}$ and the vector $q \in R^{2n+2m}$ are defined respectively as

$$M = \begin{bmatrix} G & -C^{\mathrm{T}} \\ C & \mathbf{0} \end{bmatrix}, q = \begin{bmatrix} d \\ -h \end{bmatrix}.$$

The above inequality problem (59) can be solved by the following piecewise-linear projection equation (Zhang and Zhang, 2013a) as

$$P_{\Omega}(u - (Mu + q)) - u = 0 \tag{60}$$

TABLE 4 | Joint drifts during the period of number "47" writing synthesized by the proposed NDSO scheme (39)–(43) which considers repetitive motion planning, joint limits, and feedback.

	Joint displacements (rad)	Joint displacements (degree)
LEFT ARM		
$\theta_{\mathrm{L}1}(2) - \theta_{\mathrm{L}1}(0)$	$+1.78619 \times 10^{-3}$	+0.10234
$\theta_{\mathrm{L}2}(2) - \theta_{\mathrm{L}2}(0)$	$+1.40074 \times 10^{-4}$	+0.00803
$\theta_{\mathrm{L}3}(2) - \theta_{\mathrm{L}3}(0)$	-3.31570×10^{-4}	−0.01900
$\theta_{\mathrm{L}4}(2) - \theta_{\mathrm{L}4}(0)$	$+7.76503 \times 10^{-5}$	+0.00445
$\theta_{\mathrm{L}5}(2) - \theta_{\mathrm{L}5}(0)$	-2.15183×10^{-3}	−0.12329
$\theta_{\mathrm{L}6}(2) - \theta_{\mathrm{L}6}(0)$	-1.79191×10^{-3}	−0.10267
$\theta_{\mathrm{L}7}(2) - \theta_{\mathrm{L}7}(0)$	0.00000	0.00000
RIGHT ARM		
$\theta_{\mathrm{R}1}(2) - \theta_{\mathrm{R}1}(0)$	-1.30848×10^{-3}	−0.07497
$\theta_{\mathrm{R}2}(2) - \theta_{\mathrm{R}2}(0)$	$+4.72086 \times 10^{-5}$	+0.00270
$\theta_{\mathrm{R}3}(2) - \theta_{\mathrm{R}3}(0)$	$+2.97110 \times 10^{-4}$	+0.01702
$\theta_{\mathrm{R}4}(2) - \theta_{\mathrm{R}4}(0)$	$+2.60082 \times 10^{-5}$	+0.00149
$\theta_{\mathrm{R}5}(2) - \theta_{\mathrm{R}5}(0)$	$+2.39140 \times 10^{-3}$	+0.13702
$\theta_{\mathrm{R}6}(2) - \theta_{\mathrm{R}6}(0)$	-6.03425×10^{-4}	−0.03457
$\theta_{\mathrm{R}7}(2) - \theta_{\mathrm{R}7}(0)$	0.00000	0.00000

where $P_{\Omega}(\cdot) \in R^{2n+2m} \rightarrow \Omega \subset R^{2n+2m}$ is a projection operator defined from R^{2n+2m} onto Ω, and the ith element of $P_{\Omega}(u)$ is

$$\begin{cases} u_{\mathrm{i}}^{-}, & if\ u_{\mathrm{i}} < u_{\mathrm{i}}^{-} \\ u_{\mathrm{i}}, & if\ u_{\mathrm{i}}^{-} < u_{\mathrm{i}} < u_{\mathrm{i}}^{+}, \forall i \in \{1, 2, \cdots, 2n+2m\} \\ u_{\mathrm{i}}^{+}, & if\ u_{\mathrm{i}} > u_{\mathrm{i}}^{+} \end{cases}$$

According to previous design experience on recurrent neural networks (Zhang and Zhang, 2013a), a linear-variational-inequality-based primal-dual neural network (abbreviated as LVI-PDNN) is employed to solve the piecewise-linear projection Equation (60) as well as the quadratic programming problem (56)–(58), i.e.,

$$\dot{u} = \gamma(I + M^{\mathrm{T}})(P_{\Omega}(u - (Mu + q)) - u), \tag{61}$$

where I is an identity matrix, and parameter $\gamma \in R$ is a positive design parameter designed to scale the convergence rate of neural network. From Zhang and Zhang (2013a), the state vector $u(t)$ of the primal-dual neural network in Equation (61) is globally convergent to an equilibrium point u^{*}. Furthermore, the first $2n$ elements of u^{*} constitute the solutions to the original quadratic programming problem (56)–(58).

Considering the systematic error generally including the differentiation error and the implementation error, the perturbed LVI-PDNN is formulated as

$$\dot{u} = \gamma(I + M^{\mathrm{T}} + \Delta D)(P_{\Omega}(u - (Mu + q)) - u) + \Delta S, \tag{62}$$

where $\Delta D \in R^{(2n+2m)\times(2n+2m)}$ and $\Delta S \in R^{2n+2m}$ denote the differentiation error matrix and the implementation error vector respectively. Equation (62) will be used in the experiment on robustness verification.

4. COMPUTER SIMULATIONS

In this section, the dual PA10 robot manipulators synthesized by the presented NDSO scheme are expected to track three closed complex trajectories, i.e., a pentagram, number "47" writing and end-effector-coupled pentagram. Each manipulator has 7 degrees-of-freedom, and the dual-manipulators have 14 degrees-of-freedom in total. All joint physical limits are shown in **Table 1**. The design parameter α and β are set 4, and the design parameter $\gamma = 10^{5}$ in the ensuing simulations.

4.1. Pentagram Path-Tracking

In this section, the dual PA10 robot manipulators are expected to cooperatively track a pentagram-path. Initial joint angles of the left arm are $\theta_{\mathrm{L}}(0) = [0; -\pi/4; 0; \pi/2; 0; -\pi/6; 0]$ rad, and initial joint angles of the right arm are $\theta_{\mathrm{R}}(0) = [-\pi; -\pi/4; 0; \pi/2; 0; -\pi/6; 0]$ rad. The task execution period is 4 s. For comparisons, four sets of equations in which the variables $d, x^{-}, x^{+}, \rho_{\mathrm{p}}, \rho_{\mathrm{v}}$ in Equation (56)–(58) are set different values are showed in **Table 2**. Then the four sets of equations make up three groups of contrast experiments which are performed to prove the efficiency of repetitive motion criterion, physical limits criterion and feedback criterion. Firstly, comparison results between the scheme considering physical-limits, feedback criteria but no repetitive motion criterion (experiment 1) and the NDSO scheme considering the repetitive motion, physical limits and feedback criteria (experiment 4) performed on dual PA10 robot manipulators are illustrated in **Figures 3A,B**, respectively. **Figure 3A** shows that the final states of the end-effectors of the left and right arms of the dual-redundant-manipulators do not coincide with the initial states, which means that the end-effectors of the dual-redundant-manipulators can not return to the initial states when the task is completed. That is to say, the joint drift phenomenon has happened. It is noticed that this phenomenon is not expected in practical applications because it is necessary to add extra self-motion to readjust the manipulator's configuration at the end of each task execution period in the cyclic motions. Evidently, this approach is inefficient. To remedy this joint-drift problem, the repetitive motion planning criterion is developed, and the corresponding result is shown in **Figure 3B**. Evidently, the final states of the dual-redundant-manipulators coincide well with their initial states. Comparing **Figures 3A,B**, we can see that the NDSO scheme nearly eliminates the joint-drift phenomena since it considers the repetitive motion criterion, and the efficiency of repetitive motion criterion is verified.

Secondly, comparisons between the scheme with considering the repetitive motion planning and feedback criterion but without considering limits (experiment 2) and the NDSO scheme with considering the limits criterion (experiment 4) are illustrated in **Figures 4**, **5**, respectively. The joint angles are shown in **Figures 4C,D**, **5C,D**. We can see that the final states of joints coincide with the initial ones and thus the efficiency of the repetitive motion planning criterion are illustrated. The velocities are shown in **Figures 4C,D**, **5C,D**. It can be seen from the figures that the velocities start from zero and end at zero, which is fit with the actual situations. However, **Figures 4E,F** show that $\ddot{\theta}_{\mathrm{L}3}$

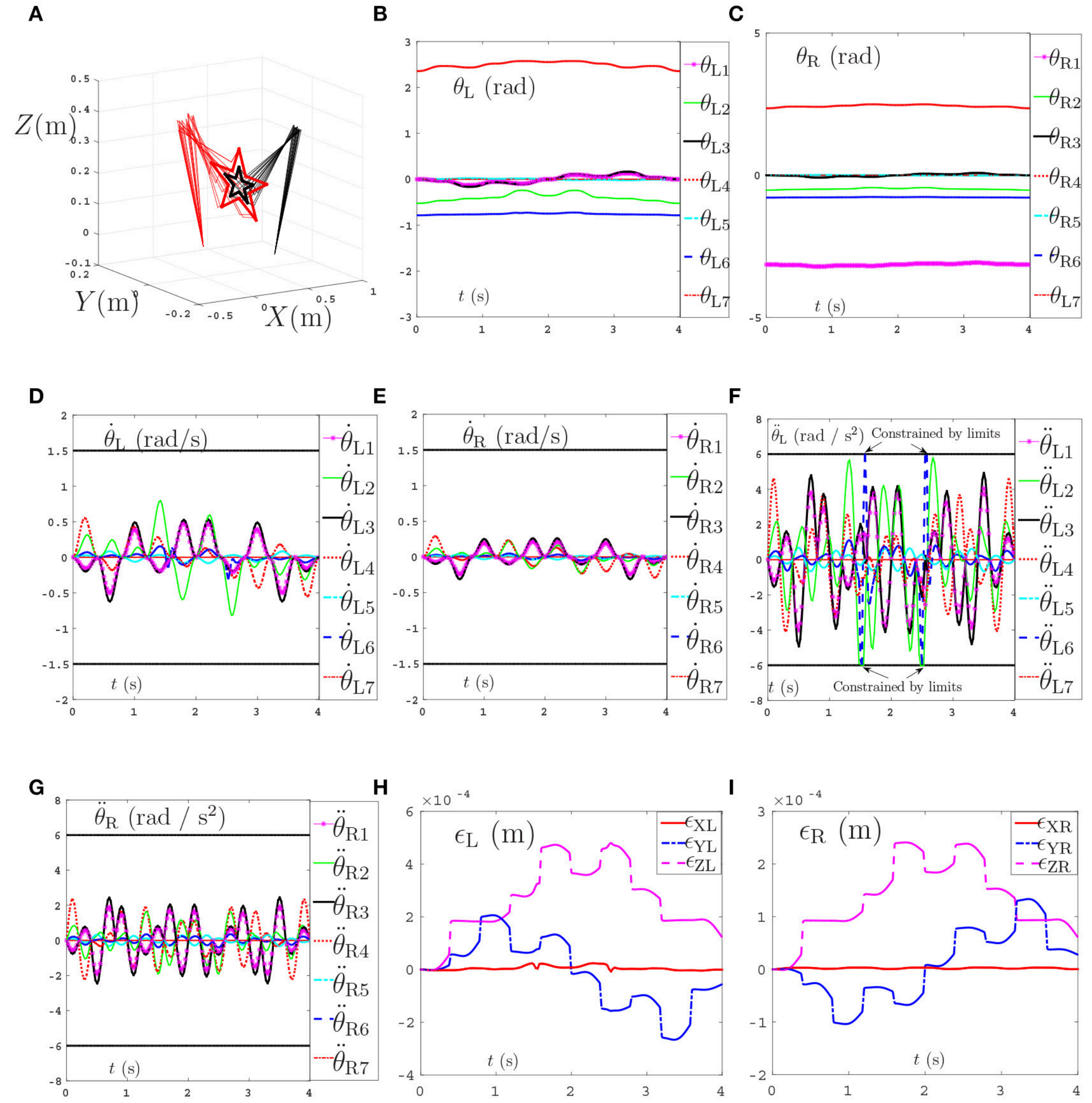


FIGURE 9 | Tracking trajectories, joint angles, joint velocities, and joint accelerations during the period of end-effector-coupled pentagram-path tracking synthesized by the proposed NDSO scheme (39)–(43) which considers repetitive motion planning, joint limits, and feedbacks. **(A)** 3-D simulation tracking trajectories. **(B)** Left arm joint angle θ_{L}. **(C)** Right arm joint angle θ_{R}. **(D)** Left arm joint velocity $\dot\theta_{\mathrm{L}}$. **(E)** Right arm joint velocity $\dot\theta_{\mathrm{R}}$. **(F)** Left arm joint acceleration $\ddot\theta_{\mathrm{L}}$. **(G)** Right arm joint acceleration $\ddot\theta_{\mathrm{R}}$. **(H)** Position error of left arm ϵ_{L}. **(I)** Position error of right arm ϵ_{R}.

and $\ddot\theta_{\mathrm{R3}}$ exceed their upper or lower acceleration limits in 0–4s. This may lead to the damage to the dual-redundant-manipulators and is less desirable for practical applications. By comparison, joint accelerations $\ddot\theta_{\mathrm{L3}}$ and $\ddot\theta_{\mathrm{R3}}$ in **Figures 4E,F** reach but never exceed their acceleration limits. This comparison result verifies the efficiency of the physical limits are very useful in applications.

Thirdly, comparisons between the NDSO scheme without considering feedback (experiment 3) and the NDSO scheme proposed in this paper with considering feedback (experiment 4) are illustrated in **Figures 6**, **7**, respectively. In the NDSO scheme, the feedback parameters ρ_{P} and ρ_{V} are set as 1 and 200, respectively. It can be seen from **Figure 6** that the end-effector position errors of left and right arms are less than 6.0 × 10^{-4} m. However, the position errors become bigger and bigger as the task execution, i.e., the trend of the position errors are diverging. This would lead to bigger accumulated errors if the scheme is applied to perform cyclic tasks. Contrastively, the position errors in **Figures 7A,B** show that position errors are very tiny and become smaller and smaller since the proposed NDSO scheme is applied.

Last but not least, the joint drifts are measured when the position, velocity and acceleration feedback are taken into consideration in the NDSO scheme. **Table 3** lists small joint drifts which are all less than 6.2 × 10^{-3} rads when

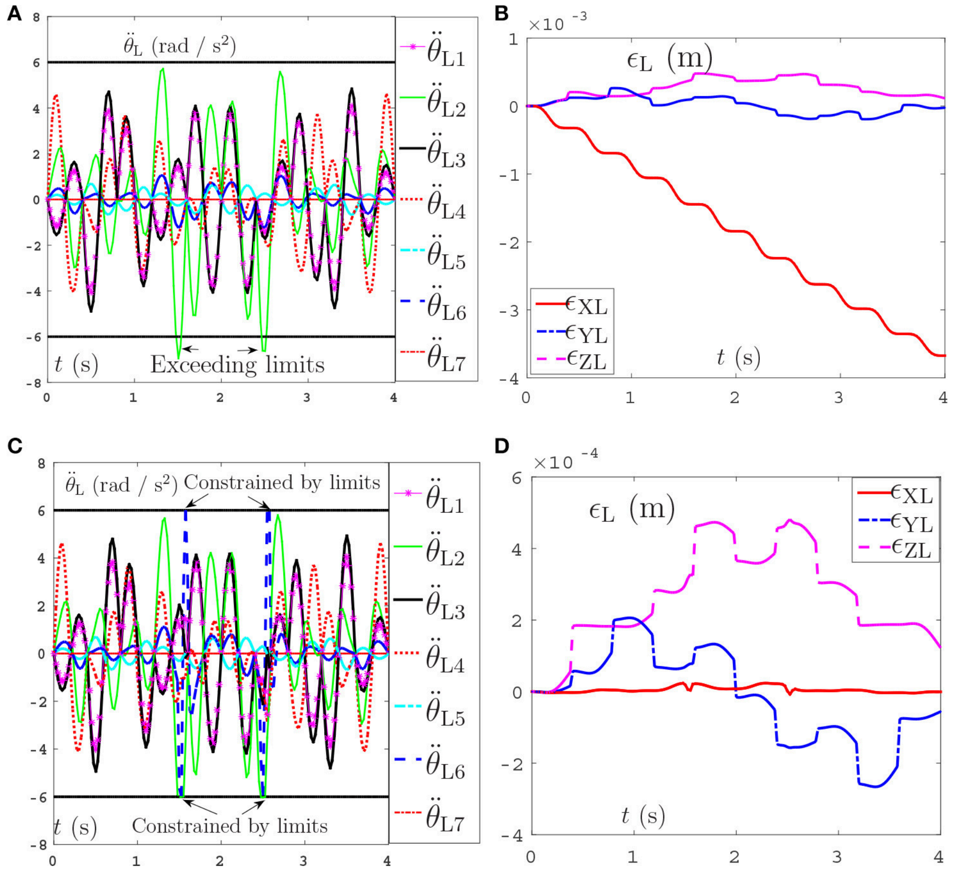


FIGURE 10 | Joint accelerations and position errors of the left arm during the period of end-effector-coupled pentagram-path tracking synthesized by pseudo-inverse scheme (63) and NDSO scheme (53)–(55). **(A)** Left arm joint acceleration $\ddot{\theta}_L$ of pseudo-inverse scheme. **(B)** Position error of left arm ϵ_L of pseudo-inverse scheme. **(C)** Left arm joint acceleration $\ddot{\theta}_L$ of NDSO scheme. **(D)** Position error of left arm ϵ_L of NDSO scheme.

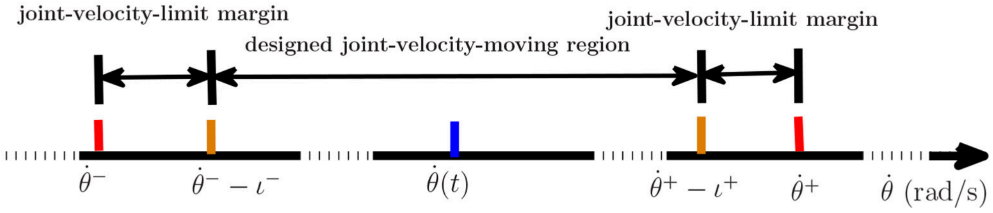

joint-velocity-limit margin
joint-velocity-limit margin
designed joint-velocity-moving region
$\dot{\theta}^-$
$\dot{\theta}^- - \iota^-$
$\dot{\theta}(t)$
$\dot{\theta}^+ - \iota^+$
$\dot{\theta}^+$
$\dot{\theta}$ (rad/s)

FIGURE 11 | The form of the joint-velocity-limit margins and joint-velocity moving region where ι^- is minus and ι^+ is positive.

the dual-redundant-manipulators track a pentagram-path synthesized by NDSO scheme.

In a word, the above three comparison experiments on tracking a pentagram-path illustrate well the effectiveness, safety and accuracy of the proposed NDSO scheme (39)–(43) and the LVI-PDNN to solve the joint-drift problem.

4.2. Number Writing

In order to further verify the effectiveness, accuracy and generalization of the proposed NDSO scheme (39)–(43), another new end-effector task, i.e., number "47" writing, is expected to finished by the same dual PA10 robot manipulators which is synthesized by the NDSO scheme. In the simulations, ρ_P and ρ_V in Equations (48) and (49) are set as 1 and 100 respectively. Initial joint angles of the left arm are $\theta_L(0) = [0; -\pi/4; 0; \pi/2; 0; -\pi/6; 0]$ rad, and initial joint angles of the right arm are $\theta_R(0) = [-\pi; -\pi/4; 0; \pi/2; 0; -\pi/6; 0]$ rad. The task execution period is 2s.

The tracking trajectories, joint angles, joint velocities, joint accelerations and end-effector position errors are shown in **Figure 8**, and the joint drifts between the final state and the initial states of the left and right arms are listed in **Table 4**. As

TABLE 5 | $\dot{\theta}_{\mathrm{L}}^{\pm}$ and $\dot{\theta}_{\mathrm{R}}^{\pm}$ (rad/s) used in Robustness experiments.

	Joint L1	Joint L2	Joint L3	Joint L4	Joint L5	Joint L6	Joint L7
LEFT ARM							
$\dot{\theta}^{+}$	1.5	1.5	1.5	1.5	1.5	1.5	1.5
$\dot{\theta}^{-}$	−**0.5**	−1.5	−1.5	−1.5	**-0.5**	**-0.5**	−1.5
RIGHT ARM							
$\dot{\theta}^{+}$	**0.5**	1.5	1.5	1.5	**0.4**	1.5	1.5
$\dot{\theta}^{-}$	−1.5	−1.5	−1.5	−1.5	−1.5	**-0.4**	−1.5

TABLE 6 | Joint drifts when dual-redundant-manipulators tracking a pentagram-path synthesized by NDSO scheme considering differentiation errors and implementation errors.

	Joint displacements (rad)	Joint displacements (degree)
LEFT ARM		
$\theta_{\mathrm{L1}}(4) - \theta_{\mathrm{L1}}(0)$	$+3.89070 \times 10^{-3}$	+0.22292
$\theta_{\mathrm{L2}}(4) - \theta_{\mathrm{L2}}(0)$	$+1.86462 \times 10^{-4}$	+0.01068
$\theta_{\mathrm{L3}}(4) - \theta_{\mathrm{L3}}(0)$	-8.38750×10^{-4}	−0.04806
$\theta_{\mathrm{L4}}(4) - \theta_{\mathrm{L4}}(0)$	-3.11366×10^{-5}	−0.00178
$\theta_{\mathrm{L5}}(4) - \theta_{\mathrm{L5}}(0)$	-5.90065×10^{-3}	−0.33808
$\theta_{\mathrm{L6}}(4) - \theta_{\mathrm{L6}}(0)$	-1.51023×10^{-3}	−0.08653
$\theta_{\mathrm{L7}}(4) - \theta_{\mathrm{L7}}(0)$	-6.41855×10^{-6}	−0.00037
RIGHT ARM		
$\theta_{\mathrm{R1}}(4) - \theta_{\mathrm{R1}}(0)$	-3.54920×10^{-3}	−0.20335
$\theta_{\mathrm{R2}}(4) - \theta_{\mathrm{R2}}(0)$	-1.56336×10^{-5}	−0.00090
$\theta_{\mathrm{R3}}(4) - \theta_{\mathrm{R3}}(0)$	$+7.25769 \times 10^{-4}$	+0.04158
$\theta_{\mathrm{R4}}(4) - \theta_{\mathrm{R4}}(0)$	$+2.49020 \times 10^{-4}$	+0.01427
$\theta_{\mathrm{R5}}(4) - \theta_{\mathrm{R5}}(0)$	$+5.66373 \times 10^{-3}$	+0.32451
$\theta_{\mathrm{R6}}(4) - \theta_{\mathrm{R6}}(0)$	-8.71307×10^{-4}	−0.04992
$\theta_{\mathrm{R7}}(4) - \theta_{\mathrm{R7}}(0)$	$+1.68829 \times 10^{-5}$	+0.00097

can be seen from **Figure 8A**, the end-effector task, i.e., number "47" writing is finished by the dual-redundant-manipulators synthesized by NDSO scheme (39)–(43) very well. In addition, as is shown in **Figures 8B–E**, all joint angles and joint velocities are within their joint limits, and the initial and final joint velocities and joint accelerations are both zero. From **Figures 8F,G**, we can see that the joint accelerations $\ddot{\theta}_{\mathrm{L2}}$ and during the range 0.3s–0.5s, $\ddot{\theta}_{\mathrm{L3}}$ during the range 1.6s–2s, $\ddot{\theta}_{\mathrm{R3}}$ and $\ddot{\theta}_{\mathrm{R5}}$ during the range 0.3s–1.3s increase sharply and are constrained by the upper and lower acceleration limits. This means that all the joint variables are in safe motion ranges. End-effector position errors ϵ of the dual-redundant-manipulators are shown in **Figures 8H,I**, which are very small ($\leqslant 3 \times 10^{-4}$ m). It is worth pointing out that the end-effector position errors tend to convergence as the task execution due to the position and velocity feedbacks considered in the NDSO scheme. **Table 4** shows that the small joint displacements of NDSO scheme are all less than 2.4×10^{-3} rads.

This number writing simulations further verify the effectiveness of the proposed NDSO scheme.

4.3. Coupled Task Tracking Example

In order to further verify the well-coordinated performance between dual-redundant-manipulators of the proposed NDSO scheme (39)–(43), a complex end-effector-coupled task, i.e., the left arm is drawing an outside pentagram while the right arm is drawing an inside one synchronously by the dual PA10 robot manipulators. Initial joint angles of the left arm are $\theta_{\mathrm{L}}(0) = [0; -\pi/6; 0; 3\pi/4; 0; -\pi/4; 0]$ rad, and initial joint angles of the right arm are $\theta_{\mathrm{R}}(0) = [-\pi; -\pi/6; 0; 3\pi/4; 0; -\pi/4; 0]$ rad. The relation of the left and right end-effector tasks is

$$\begin{cases} \ddot{r}_{\mathrm{RX}} = \ddot{r}_{\mathrm{LX}} \\ \ddot{r}_{\mathrm{RY}} = 0.5 \times \ddot{r}_{\mathrm{LY}}, \forall t \in \{0, T\} \\ \ddot{r}_{\mathrm{RZ}} = 0.5 \times \ddot{r}_{\mathrm{LZ}} \end{cases}$$

In the simulations, ρ_{P} and ρ_{V} in Equations (48) and (49) are set as 1 and 200 respectively. The task execution period is 4s. The tracking trajectories, joint angles, joint velocities, joint accelerations and end-effector position errors are shown in **Figure 9**. From **Figure 9A** we can see that the coupled end-effector task is completed by the dual-redundant-manipulators synthesized by NDSO scheme. What's more, the final states perfectly coincide the initial states. In addition, in **Figures 9B–E**, all joint angles and joint velocities are within their joint limits, and the initial and final joint velocities and joint accelerations are both zero. From **Figures 9F,G**, we can see that the joint accelerations $\ddot{\theta}_{\mathrm{L2}}$ and $\ddot{\theta}_{\mathrm{L6}}$ during 1–3s, change sharply but both are constrained by their acceleration limits. This means that all the joint variables are in the safe motion ranges. The end-effector position errors ϵ shown in **Figures 9H,I** are very small ($\leqslant 6 \times 10^{-4}$ m) and convergent.

In summary, the above three end-effector tasks and comparisons, i.e., pentagram-path tracking, number "47" writing, and the coupled task tracking example, demonstrate that complex end-effector tasks can be well performed by the presented NDSO scheme (39)–(43). From the simulations, it is known that the NDSO scheme can achieve the repetitive motion effectively and accurately. In addition, the position errors of the end-effectors can converge to nearly zero at the end of each cycle due to taking feedback into consideration.

4.4. Compared With Pseudo-Inverse Method

In order to further illustrate the advantages of the proposed NDSO scheme, both of the traditional pseudo-inverse method and the proposed NDSO are used to perform on a dual-redundant-manipulators to track the previous coupled pentagram paths. Initial joint angles of the left and right arms are set the same as before. The formulation of the pseudo-inverse method is

$$\ddot{\theta} = J^{+}\ddot{r}_{\mathrm{af}}(t) - [I - J^{+}J]b(t) \tag{63}$$

where $\ddot{\theta}$, $\ddot{r}_{\mathrm{af}}(t)$, J and $b(t)$ have the same definition in the NDSO scheme. J^{+} means the pseudo-inverse matrix of Jacobian matrix J and I is an identity matrix in $m + n$ dimensions.

The comparative simulations are shown in **Figure 10**. Due to space limitation, only the joint acceleration and the position errors of left manipulators between the proposed NDSO scheme and the pseudo-inverse method are shown here. Specifically,

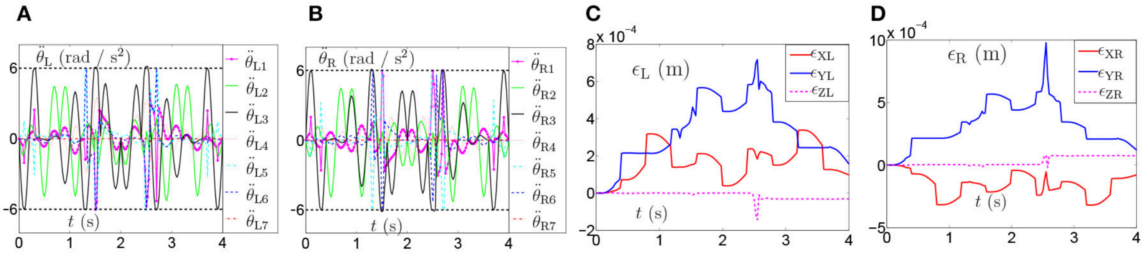


FIGURE 12 | Joint accelerations and position errors during the period of pentagram-path tracking task synthesized by NDSO scheme considering differentiation errors and implementation errors. **(A)** Left arm joint acceleration $\ddot{\theta}_L$; **(B)** Right arm joint acceleration $\ddot{\theta}_R$; **(C)** Position error of left arm ϵ_L; **(D)** Position error of right arm ϵ_R.

Figures 10A,B show the simulation result of the pseudo-inverse method, and **Figures 10C,D** show the simulation result of the proposed NDSO method. From **Figure 10A**, we can see that the joint acceleration $\ddot{\theta}_{L2}$ exceeds its limits about 1.3s and 2.6s, and the end-effector position errors of the left arm shown in **Figure 10C** ϵ_{XL} are divergent as time goes on. That is to say, the end-effector of the dual-redundant-manipulators synthesized by the pseudo-inverse method can track the desired path but may lead to exceeding limit problem and the positioning errors will accumulate.

This comparison result further illustrate the efficiency and excellent advantages of the proposed NDSO scheme.

4.5. Robustness Verification

In this subsection, systematic errors are taken into consideration and the perturbed LVI-PDNN in Equation (62) is used to solve the path-tracking problem of the dual redundant manipulators. The pentagram path-tracking task in 4.1 is adopted to compare the joint displacements without perturbation in **Table 3**. During the simulations, error-matrix ΔD and error-vector ΔS are generated randomly. The element Δ_i of them is formulated as

$$\Delta_i = 0.1 * \nu_a(\nu_c sin(\nu_b t) + (1 - \nu_c)cos(\nu_b t)) \qquad (64)$$

where ν_a is a random integer in $[-5, 5]$, ν_b is a random integer in $[1, 5]$ and ν_c is a random integer in $[0, 1]$. All of them are distributed evenly. ν_a and ν_b control the amplitude and frequency of the element respectively. ν_c controls the form of the perturbation function to be sine function ($\nu_c = 1$) or to be cosine function ($\nu_c = 0$). The initial joint angles of dual arms are set as same in 4.1. The parameters $d, x^-, x^+, \rho_p, \rho_v$ are set according to the 4th set of equations in **Table 2**. Inspired by Zhang and Zhang (2013b), we consider joint-velocity-limit margins ι shown in **Figure 11** in our experiments. The updated $\dot{\theta}_L^{\pm}(t)$ and $\dot{\theta}_R^{\pm}(t)$ in (51) are shown in **Table 5**, where the margins considered joint-velocity-limits are highlighted in bold.

The joint drifts of dual arms are shown in **Table 6**, which shows that the joint displacement of every joint almost has no change compared with the result in **Table 3**. The joint accelerations and position errors during the period of pentagram-path tracking task are recorded in **Figures 12A–D**. The curves in **Figures 12A–D** show that the joint accelerations are all constrained within the limits (i.e., $\pm$6rad/s^2). Besides, the position errors have been controlled within a very small range which is lower than 1×10^{-3} (m). Although there exists time-varying systematic perturbation, the position errors are still convergent at the end of the task execution. In summary, the proposed NDSO method performs well under the perturbation and has strong robustness.

5. CONCLUSION

In this paper, a neural-dynamic based synchronous-optimization scheme of dual redundant robot manipulators scheme (NDSO) of dual-redundant-manipulators for tracking complex paths has been proposed to solve the joint-drift problem. The scheme can not only finish the end-effector task collaboratively with the dual-redundant-manipulators, but also achieve repetitive motion, avoid physical limits and position-error convergence. First, the left and right manipulator subschemes are formulated and then are combined to one quadratic program scheme, i.e., the NDSO scheme. Next, the scheme is unified into a standard quadratic programming problem. Finally, the quadratic programming problem is solved by a linear-variational-inequality primal-dual neural networks. Three complex end-effector tasks and comparisons, i.e., pentagram-path tracking, number writing, and coupled tasks have verified the effectiveness, accuracy, repeatability, safety, generality and robustness of the proposed NDSO scheme. To the best of authors' knowledge, it is the first time to propose such a NDSO scheme with so many optimization criteria and can solve the joint-drift problems in three three-dimensional workspace. The future work is to exploit higher efficient resolving algorithms to further improve the performance of the scheme and consider the control input saturation problem and uncertainties.

AUTHOR CONTRIBUTIONS

The idea of the paper that we can try to solve the optimization problem on the acceleration level of dual redundant manipulators is proposed by ZZ. The paper is drafted and revised by ZZ and QZ together. QZ and WF design and implement the experiment in coordination.

FUNDING

This work was supported in part by the National Natural Science Foundation under Grants 61603142 and 61633010, the Guangdong Foundation for Distinguished Young Scholars under Grant 2017A030306009, the Guangdong Youth Talent Support Program of Scientific and Technological Innovation under Grant 2017TQ04X475, the Science and Technology Program of Guangzhou under Grant 201707010225, the Fundamental Research Funds for Central Universities under Grant x2zdD2182410, the Scientific Research Starting Foundation of South China University of Technology, the National Key R&D Program of China under Grant 2017YFB1002505, the National Key Basic Research Program of China (973 Program) under Grant 2015CB351703, and the Guangdong Natural Science Foundation under Grant 2014A030312005.

REFERENCES

Cai, B., and Zhang, Y. (2012). Different-level redundancy-resolution and its equivalent relationship analysis for robot manipulators using gradient-descent and zhang 's neural-dynamic methods. *IEEE Trans. Indus. Electron.* 59, 3146–3155. doi: 10.1109/TIE.2011.2106092

Cheng, F. T., Chen, T. H., and Sun, Y. Y. (1994). Resolving manipulator redundancy under inequality constraints. *IEEE Trans. Robot. Autom.* 10, 65–71. doi: 10.1109/70.285587

Chevallereau, C., and Khalil, W. (1988). "A new method for the solution of the inverse kinematics of redundant robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1 (Philadelphia, PA), 37–42.

Chikhaoui, M. T., Granna, J., Starke, J., and Burgner-Kahrs, J. (2018). Toward motion coordination control and design optimization for dual-arm concentric tube continuum robots. *IEEE Robot. Autom. Lett.* 3, 1793–1800. doi: 10.1109/LRA.2018.2800037

Dong, I. P., Kim, H., Park, C., and Kim, D. (2017). "Design and analysis of the dual arm manipulator for rescue robot," in *Proceedings of IEEE International Conference on Advanced Intelligent Mechatronics*, Vol. 1 (Munich), 608–612.

Eremin, E. L., and Shelenok, E. A. (2017). "Simulation modeling of the control system for robotic manipulator with input saturation," in *Proceedings of International Siberian Conference on Control and Communications* (Astana), 1–5.

Felip, J., and Morales, A. (2015). "A solution for the cap unscrewing task with a dual arm sensor-based system," in *Proceedings of the 15th IEEE-RAS International Conference on Humanoid Robots*, Vol. 1 (Seoul), 823–828.

Flacco, F., and De Luca, A. (2015). Discrete-time redundancy resolution at the velocity level with acceleration/torque optimization properties. *Robot. Auton. Syst.* 70, 191–201. doi: 10.1016/j.robot.2015.02.008

Guo, D., Su, Z., Sun, S., Lin, X., and Liu, Q. (2017). "A new feedback-added obstacle avoidance scheme for motion planning of redundant robot manipulators," in *Proceedings of the 29th Chinese Control and Decision Conference*, Vol. 1 (Chongqing), 6601–6606.

Hartman, and Philip (1982). *Ordinary Differential Equations, 2nd Edn*. Boston, MA: Birkhauser.

Ho, E. S. L., Komura, T., and Lau, R. W. H. (2005). Computing inverse kinematics with linear programming. In *ACM Symposium on Virtual Reality Software and Technology*, Vol. 1 (Monterey, CA), 163–166.

Huang, S., Xiang, J., Wei, W., and Chen, M. Z. Q. (2017). On the virtual joints for kinematic control of redundant manipulators with multiple constraints. *IEEE Trans. Control Sys. Technol.* 26, 65–76. doi: 10.1109/TCST.2017.2650684

Jin, L., and Li, S. (2016). Distributed task allocation of multiple robots: A control perspective. *IEEE Trans. Sys. Man Cybern. Sys.* 48, 693–701. doi: 10.1109/TSMC.2016.2627579

Jin, L., Liao, B., Liu, M., Xiao, L., Guo, D., and Yan, X. (2017). Different-level simultaneous minimization scheme for fault tolerance of redundant manipulator aided with discrete-time recurrent neural network. *Front. Neurorobot.* 11, 1–7. doi: 10.3389/fnbot.2017.00050

Jin, L., and Zhang, Y. (2014). G2-type srmpc scheme for synchronous manipulation of two redundant robot arms. *IEEE Trans. Cybern.* 45, 153–164. doi: 10.1109/TCYB.2014.2321390

Jr, R. S. J., and Roberts, R. G. (2015). A more compact expression of relative jacobian based on individual manipulator jacobians. *Robot. Auton. Sys.* 63, 158–164. doi: 10.1016/j.robot.2014.08.011

Klein, C. A., and Kee, K. B. (1989). The nature of drift in pseudoinverse control of kinematically redundant manipulators. *IEEE Trans. Robot. Autom.* 5, 231–234. doi: 10.1109/70.88043

Lee, J., Chang, P. H., and Jamisola, R. S. (2014). Relative impedance control for dual-arm robots performing asymmetric bimanual tasks. *IEEE Trans. Indus. Electron.* 61, 3786–3796. doi: 10.1109/TIE.2013.2266079

Lin, X., and Zhang, Y. (2013). Acceleration-level repetitive motion planning and its experimental verification on a six-link planar robot manipulator. *IEEE Trans. Control Sys. Technol.* 21, 906–914. doi: 10.1109/TCST.2012.2190142

Liu, Z., Chen, C., Zhang, Y., and Chen, C. L. (2015). Adaptive neural control for dual-arm coordination of humanoid robot with unknown nonlinearities in output mechanism. *IEEE Trans. Cybern.* 45, 507–518. doi: 10.1109/TCYB.2014.2329931

Luo, J., Zhang, J., Xie, Z., Zhang, X., Xiao, L., and Su, X. (2017). "Acceleration-level inverse-free g2 scheme for inverse kinematics path tracking of robot manipulators," in *Proceedings of the 36th Chinese Control Conference,* Vol. 1 (Dalian), 6804–6809.

Reynoso-Mora, P., Chen, W., and Tomizuka, M. (2016). A convex relaxation for the time-optimal trajectory planning of robotic manipulators along predetermined geometric paths. *Opt. Control Applic. Methods* 37, 1263–1281. doi: 10.1002/oca.2234

Shin, S., and Kim, C. (2015). Human-like motion generation and control for humanoid's dual arm object manipulation. *IEEE Trans. Indus. Electron.* 62, 2265–2276. doi: 10.1109/TIE.2014.2353017

Sun, N., Fang, Y., Chen, H., Fu, Y., and Lu, B. (2018). Nonlinear stabilizing control for ship-mounted cranes with ship roll and heave movements: design, analysis, and experiments. *IEEE Trans. Sys. Man Cybern. Sys.* 48, 1–13. doi: 10.1109/TSMC.2017.2700393

Sun, N., Fang, Y., Chen, H., and Lu, B. (2017). Amplitude-saturated nonlinear output feedback antiswing control for underactuated cranes with double-pendulum cargo dynamics. *IEEE Trans. Indus. Electron.* 64, 2135–2146. doi: 10.1109/TIE.2016.2623258

Tewari, A. (2002). *Modern Control Design With MATLAB and SIMULINK[M].* New Jersey, NJ: John Wiley and Sons, Ltd.

Toshani, H., and Farrokhi, M. (2014). Real-time inverse kinematics of redundant manipulators using neural networks and quadratic programming: a lyapunov-based approach. *Robot. Auton. Sys.* 62, 766–781. doi: 10.1016/j.robot.2014.02.005

Tran, T. T., Ge, S. S., and He, W. (2015). "Adaptive control for a robotic manipulator with uncertainties and input saturations," in *Proceedings of IEEE International Conference on Mechatronics and Automation*, Vol. 1, 1525–1530.

Wang, Y., Yan, X., He, L., Tan, H., and Zhang, Y. (2015). "Inverse-free solution of z1g1 type to acceleration-level inverse kinematics of redundant robot manipulators," in *Proceedings of the 7th International Conference on Advanced Computational Intelligence*, Vol. 1 (Mount Wuyi), 57–62.

Xiao, Y., Fan, Z., Li, W., Chen, S., Zhao, L., and Xie, H. (2016). "A manipulator design optimization based on constrained multi-objective evolutionary algorithms," in *Proceedings of International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration*, Vol. 1 (Wuhan), 199–205.

Zhang, P., Yan, Z., and Wang, J. (2014). "Obstacle and singularity avoidance for kinematically redundant manipulators based on neurodynamic optimization,." in *Proceedings of the 5th International Conference on Intelligent Control and Information Processing*, Vol. 1 (Dalian), 460–465.

Zhang, Y., Ge, S. S., and Lee, T. H. (2004). A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators. *IEEE Trans. Sys. Man Cybern. B* 34, 2126–2132. doi: 10.1109/TSMCB.2004.830347

Zhang, Y., Li, S., Gui, J., and Luo, X. (2018). Velocity-level control with compliance to acceleration-level constraints: a novel scheme for manipulator redundancy resolution. *IEEE Trans. Indus. Inform.* 14, 921–930. doi: 10.1109/TII.2017.2737363

Zhang, Y., Lv, X., Li, Z., Yang, Z., and Chen, K. (2008). Repetitive motion planning of pa10 robot arm subject to joint physical limits and using lvi-based primalcdual neural network. *Mechatronics* 18, 475–485. doi: 10.1016/j.mechatronics.2008.04.005

Zhang, Y., and Zhang, Z. (2013a). *Repetitive Motion Planning and Control of Redundant Robot Manipulators*. Berlin: Heidelberg: Springer .

Zhang, Z., Li, Z., Zhang, Y., Luo, Y., and Li, Y. (2015). Neural-dynamic-method-based dual-arm cmg scheme with time-varying constraints applied to humanoid robots. *IEEE Trans. Neural Netw. Learn. Sys.* 26, 3251–3262. doi: 10.1109/TNNLS.2015.2469147

Zhang, Z., and Zhang, Y. (2012). Acceleration-level cyclic-motion generation of constrained redundant robots tracking different paths. *IEEE Trans. Sys. Man Cybern. B* 42, 1257–1269. doi: 10.1109/TSMCB.2012.2189003

Zhang, Z., and Zhang, Y. (2013b). Repetitive motion planning and control on redundant robot manipulators with push-rod-type joints. *J. Dyn. Sys. Measur. Control* 135, 44–45. doi: 10.1115/1.4007608

Zheng, Y., and Luh, J. (1986). "Joint torques for control of two coordinated moving robots," in *Proceedings of IEEE International Conference on Robotics and Automation* (San Francisco, CA), 1375–1380.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read
for greatest visibility
and readership

FAST PUBLICATION
Around 90 days
from submission
to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative,
and constructive
peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers
acknowledged by name
on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: info@frontiersin.org | +41 21 510 17 00

REPRODUCIBILITY OF RESEARCH
Support open data
and methods to enhance
research reproducibility

DIGITAL PUBLISHING
Articles designed
for optimal readership
across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics
track visibility across
digital media

EXTENSIVE PROMOTION
Marketing
and promotion
of impactful research

LOOP RESEARCH NETWORK
Our network
increases your
article's readership