



# SERKET: An Architecture for Connecting Stochastic Models to Realize a Large-Scale Cognitive Model

Tomoaki Nakamura<sup>1\*</sup>, Takayuki Nagai<sup>1</sup> and Tadahiro Taniguchi<sup>2</sup>

<sup>1</sup> Department of Mechanical Engineering and Intelligent Systems, University of Electro-Communications, Tokyo, Japan,

<sup>2</sup> Department of Information Science and Engineering, Ritsumeikan University, Shiga, Japan

## 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. Neurobot.* 12:25. doi: 10.3389/fnbot.2018.00025

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

## 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<sup>1</sup>), 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

<sup>1</sup>Symbol emergence in robotics focuses on the real and noisy environment, and the  $\epsilon$  in Serket 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.

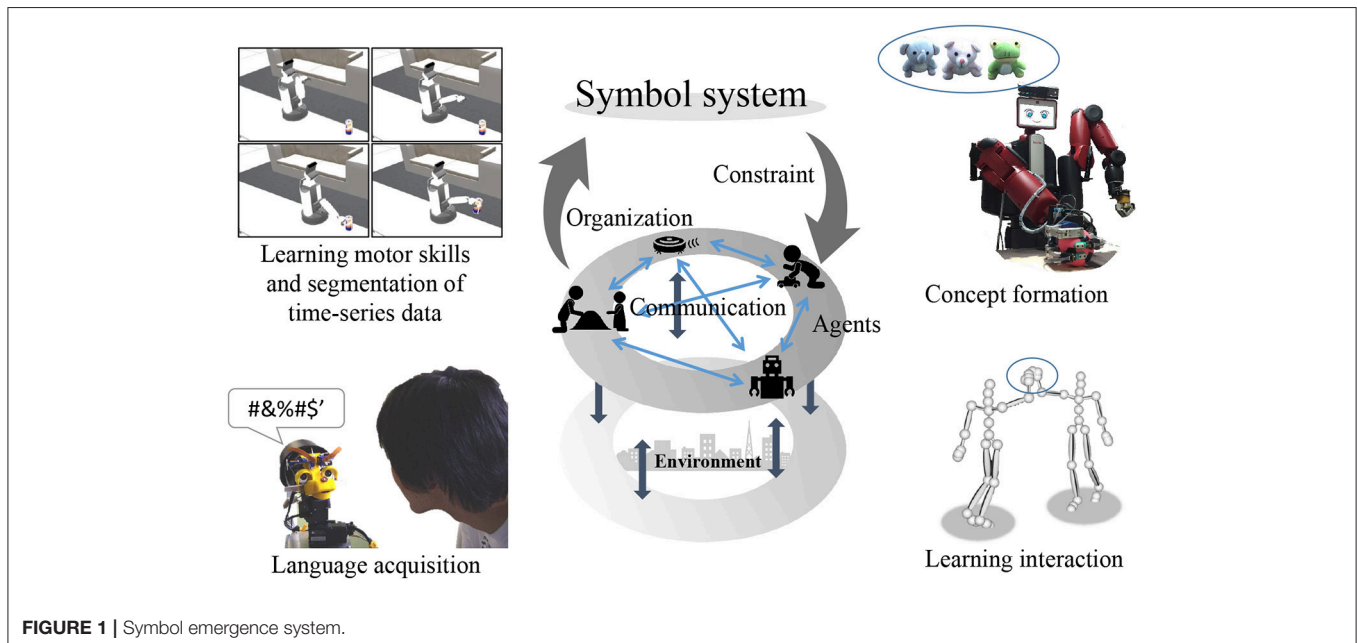
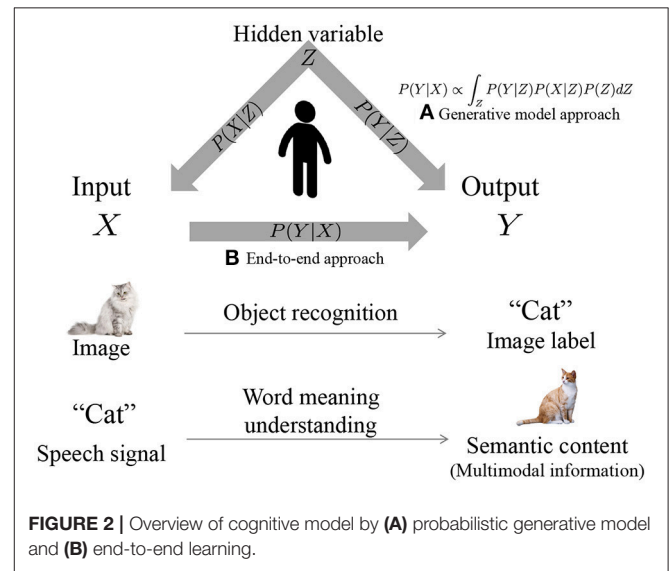
Frameworks of deep neural networks (DNNs) such as TensorFlow (Abadi et al., 2016), Keras (Chollet, 2015), and Chainer (Tokui et al., 2015) have been developed. These frameworks make it possible to construct DNN models and estimate their parameters easily. These frameworks are one of the reasons why DNNs 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



$\text{argmax}_Y P(Y|X)$ . 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:

$$P(Y|X) \propto P(Y, X) \tag{1}$$

$$= \int_Z P(Y|Z)P(X|Z)P(Z)dZ. \tag{2}$$

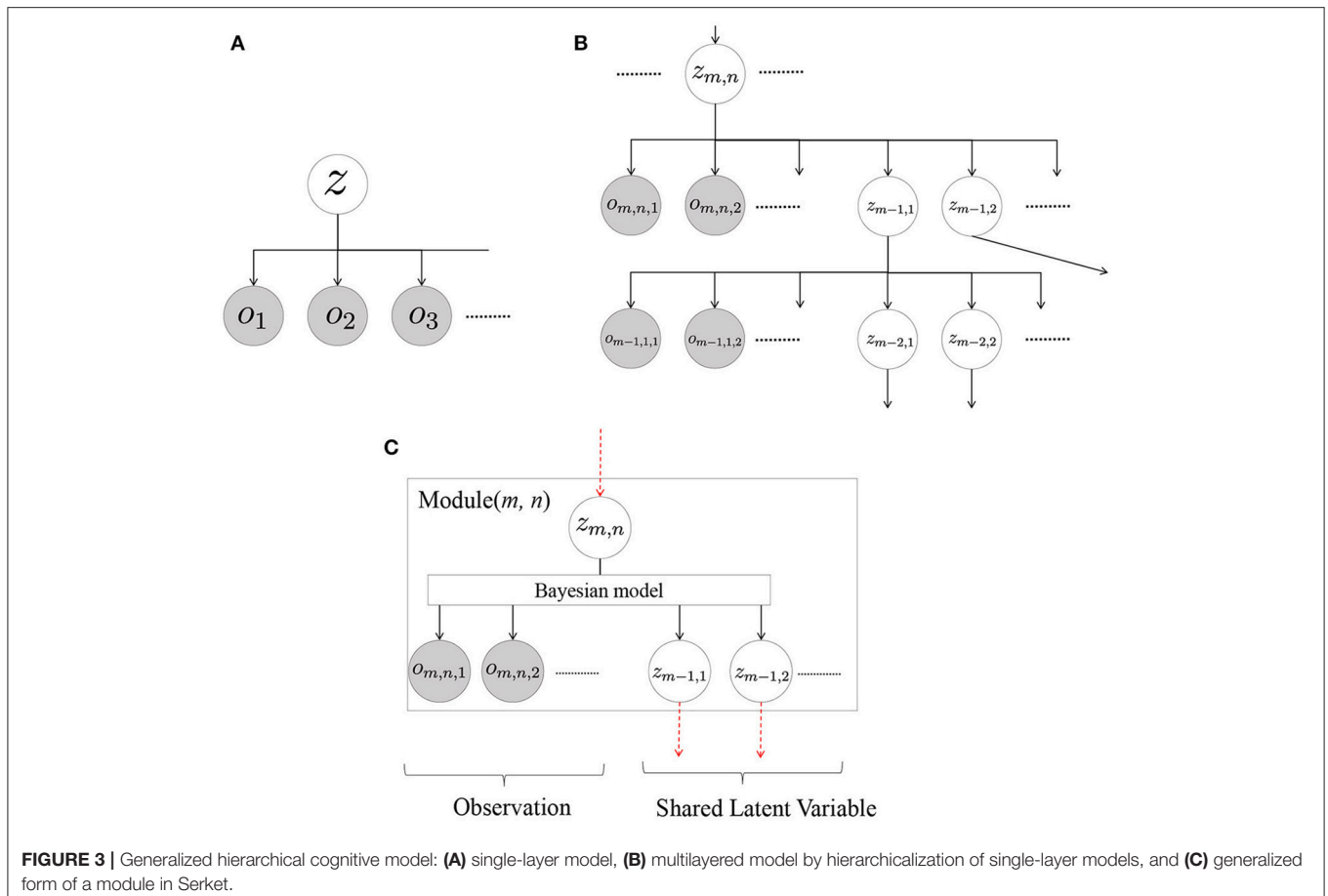
This is multimodal latent Dirichlet allocation (MLDA) (Blei and Jordan, 2003; Nakamura et al., 2009; Putthividhy et al., 2010), the details of which are described in the Appendix. However, MLDA is based on the important assumption that the observed variables  $X$  and  $Y$  are conditionally independent 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.

$$P(\mathbf{o}_1, \mathbf{o}_2, \dots) = \int_z P(z)\Pi_n P(\mathbf{o}_n|z)dz. \tag{3}$$

As mentioned earlier, it is assumed that all observations  $\mathbf{o}_1, \mathbf{o}_2, \dots$  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  $\mathbf{o}_1, \mathbf{o}_2, \dots$ , 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



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:

$$P(\mathbf{O}|z_{M,1}, z_{M,2}, \dots) = \prod_m^M \prod_n^{\bar{N}_m} \int_{z_{m,n}} P(z_{m,n}) \prod_i^{N_m} P(o_{m,n,i}|z_{m,n}) \prod_{n'}^{\bar{N}_{m-1}} P(z_{m-1,n'}|z_{m,n}) dz_{m,n}, \quad (4)$$

where  $\mathbf{O}$  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; Lallec 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.

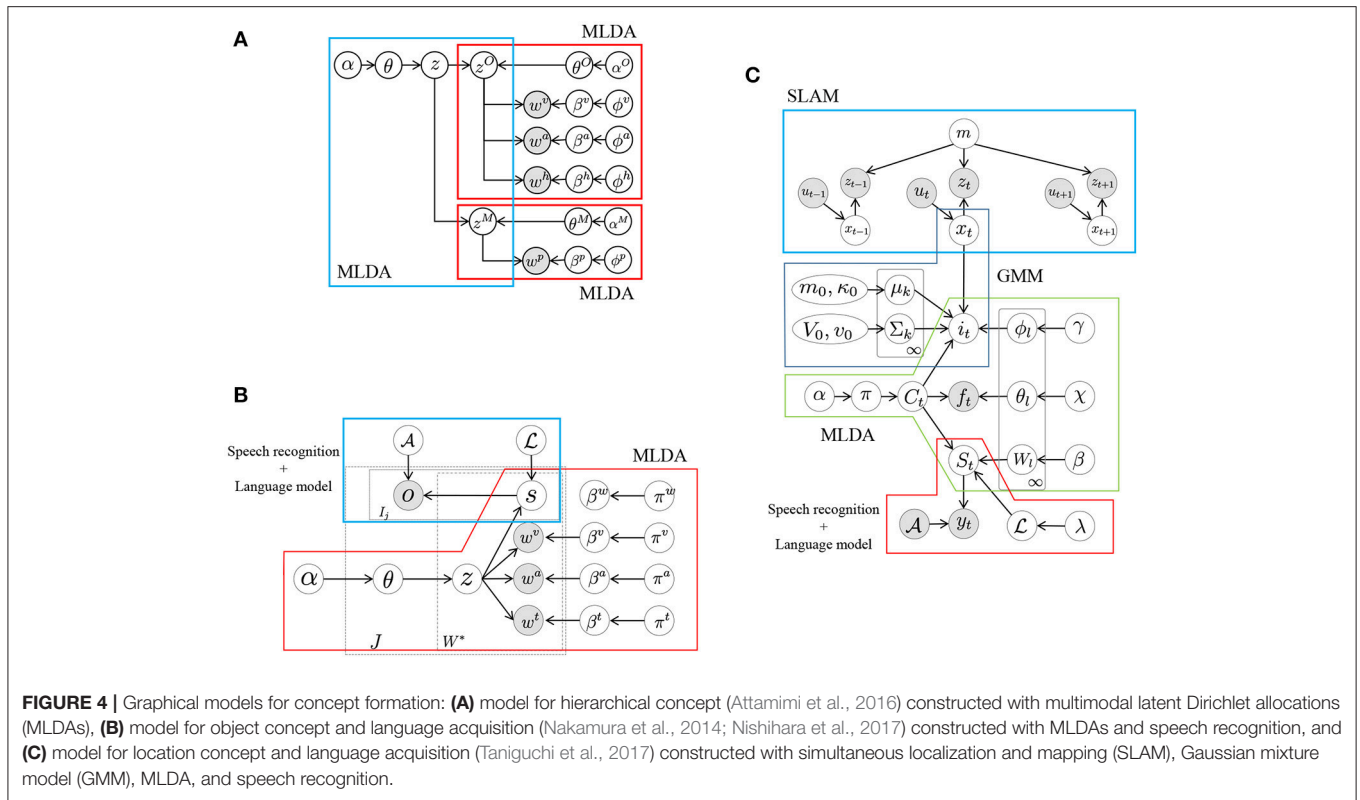
Recently, DNNs have made notable advances in many areas such as object recognition (He et al., 2015), object detection (Redmon et al., 2016), speech recognition (Amodei

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 MLDA models 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.



### 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  $\mathbf{o}_{m,n,*}$ , 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  $(m - 1, 2)$ ,  $\dots$  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

$$P(z_{m-1,i} | z_{m,n}, \mathbf{o}_{m,n,1}, \mathbf{o}_{m,n,2}, \dots, z_{m-1}). \quad (5)$$

2. The module can send the following probability by leveraging one of the methods explained in the next section:

$$P(z_{m-1,i} | z_{m,n}, \mathbf{o}_{m,n,1}, \mathbf{o}_{m,n,2}, \dots, z_{m-1}). \quad (6)$$

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:

$$P(z_{m,n} | z_{m+1,j}, \mathbf{o}_{m+1,j,1}, \mathbf{o}_{m+1,j,2}, \dots, z_m). \quad (7)$$

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:

$$z_1 \sim P(z_1|o, z_2) \tag{8}$$

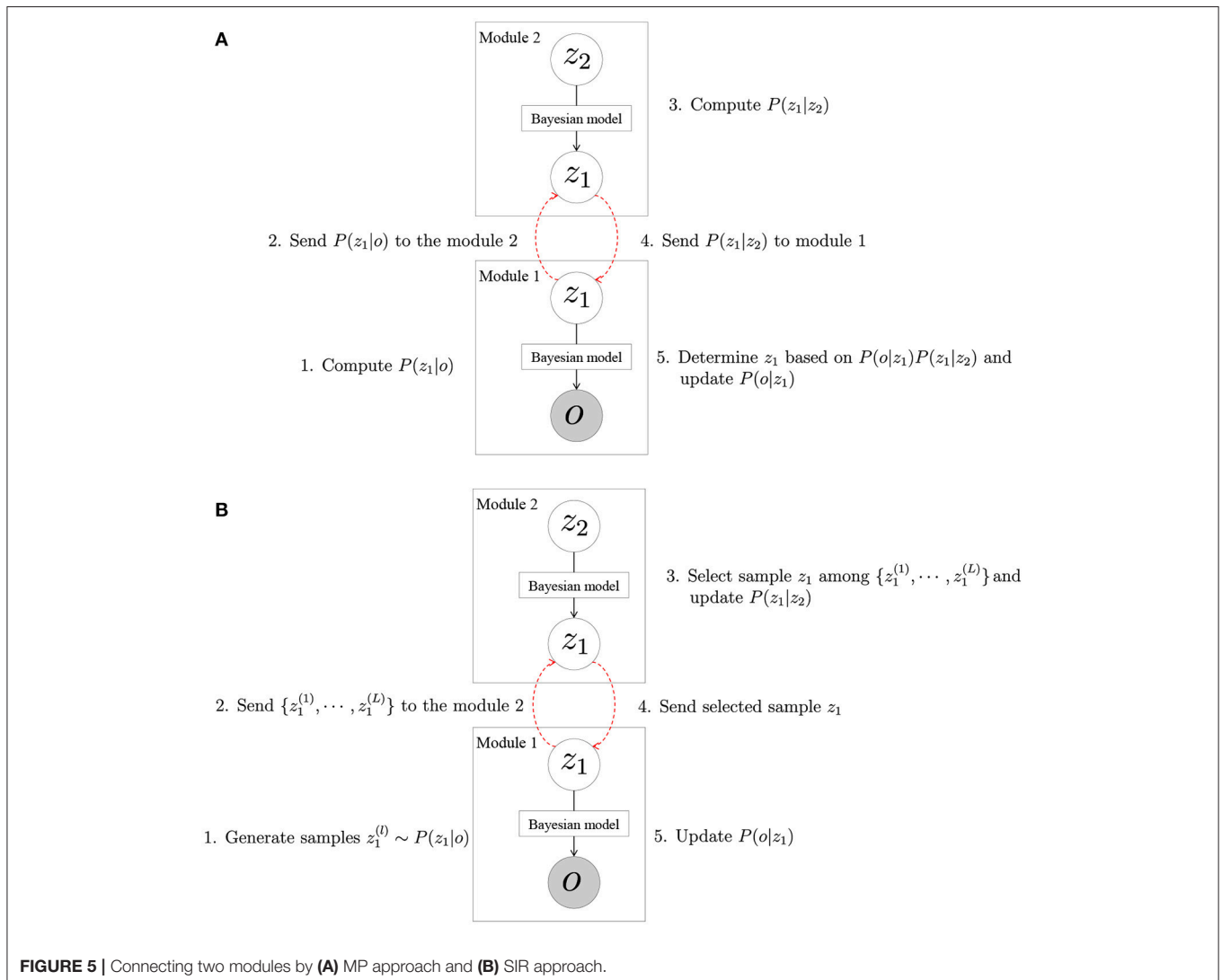
$$\propto P(z_1|o)P(z_1|z_2). \tag{9}$$

In this equation,  $P(o|z_1)$  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  $P(z_1|o)$  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,  $P(z_1|o)$  is computed.
2.  $P(z_1|o)$  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  $P(z_1|o)$ .
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  $P(o|z_1)$  are updated.

Thus, in the case when the latent variable is infinite and discrete, the modules are learned by sending and receiving the parameters



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:

$$z_1 \sim P(z_1|\mathbf{o}, z_2) \quad (10)$$

$$\propto P(z_1|\mathbf{o})P(z_1|z_2). \quad (11)$$

However, when the value of  $z_1$  is infinite or continuous, module 2 cannot send  $P(z_1|z_2)$  to module 1. Therefore,  $P(z_1|\mathbf{o})$  is first approximated by  $L$  samples  $\{z_1^{(l)} : l = 1, \dots, L\}$ :

$$z_1^{(l)} \sim P(z_1|\mathbf{o}). \quad (12)$$

This approximation is equivalent to approximating  $P(z_1|\mathbf{o})$  by the following  $\tilde{P}(z_1|\mathbf{o})$ :

$$P(z_1|\mathbf{o}) \approx \tilde{P}(z_1|\mathbf{o}) = \frac{1}{L} \sum_l \delta(z_1, z_1^{(l)}), \quad (13)$$

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  $P(z_1|z_2)$ :

$$z_1 \sim P(z_1 \in \{z_1^{(1)}, \dots, z_1^{(L)}\} | z_2). \quad (14)$$

This procedure is equivalent to sampling from the following distribution, which is an approximation of Equation (11):

$$z_1 \sim P(z_1|z_2)\tilde{P}(z_1|\mathbf{o}). \quad (15)$$

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  $A(z, z^*)$ :

$$A(z, z^*) = \min(1, \alpha) \quad (16)$$

$$\alpha = \frac{P(z^*)Q(z|z^*)}{P(z)Q(z^*|z)}, \quad (17)$$

where  $P(z)$  represents the target distribution from which the samples are generated.

The model parameters in **Figure 5** can be estimated by considering  $P(z_1|\mathbf{o})$  and  $P(z_1|z_2, \mathbf{o})$  as the proposal distribution and target distribution, respectively.  $P(z_1|z_2, \mathbf{o})$  can be transformed into

$$P(z_1|z_2, \mathbf{o}) \propto P(z_1|\mathbf{o})P(z_1|z_2)P(z_2). \quad (18)$$

Therefore,  $\alpha$  in Equation (16) becomes

$$\alpha = \frac{P(z^*)Q(z|z^*)}{P(z)Q(z^*|z)} = \frac{P(z_1^*|z_2, \mathbf{o})}{P(z_1|z_2, \mathbf{o})} \cdot \frac{P(z_1|\mathbf{o})}{P(z_1^*|\mathbf{o})} \quad (19)$$

$$= \frac{P(z_1^*|\mathbf{o})P(z_1^*|z_2)P(z_2)}{P(z_1|\mathbf{o})P(z_1|z_2)P(z_2)} \cdot \frac{P(z_1|\mathbf{o})}{P(z_1^*|\mathbf{o})} = \frac{P(z_1^*|z_2)}{P(z_1|z_2)}, \quad (20)$$

Hence, the proposal distribution  $P(z_1|\mathbf{o})$  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 three MLDA. In the lower-level MLDA, object categories  $z^O$  can be formed from multimodal information  $\mathbf{w}^v$ ,  $\mathbf{w}^a$ , and  $\mathbf{w}^h$  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 MLDA.

First, in the two MLDA shown in **Figures 6A,B**, the probabilities  $P(z_j^O | w_j^v, w_j^a, w_j^h)$  and  $P(z_j^M | w_j^p)$  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.

$$\hat{z}_{jn}^O \sim P(z_j^O | w_j^v, w_j^a, w_j^h), \tag{21}$$

$$\hat{z}_{jn}^M \sim P(z_j^M | w_j^p). \tag{22}$$

where  $w_j^v, w_j^a, w_j^h$ , and  $w_j^p$  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.

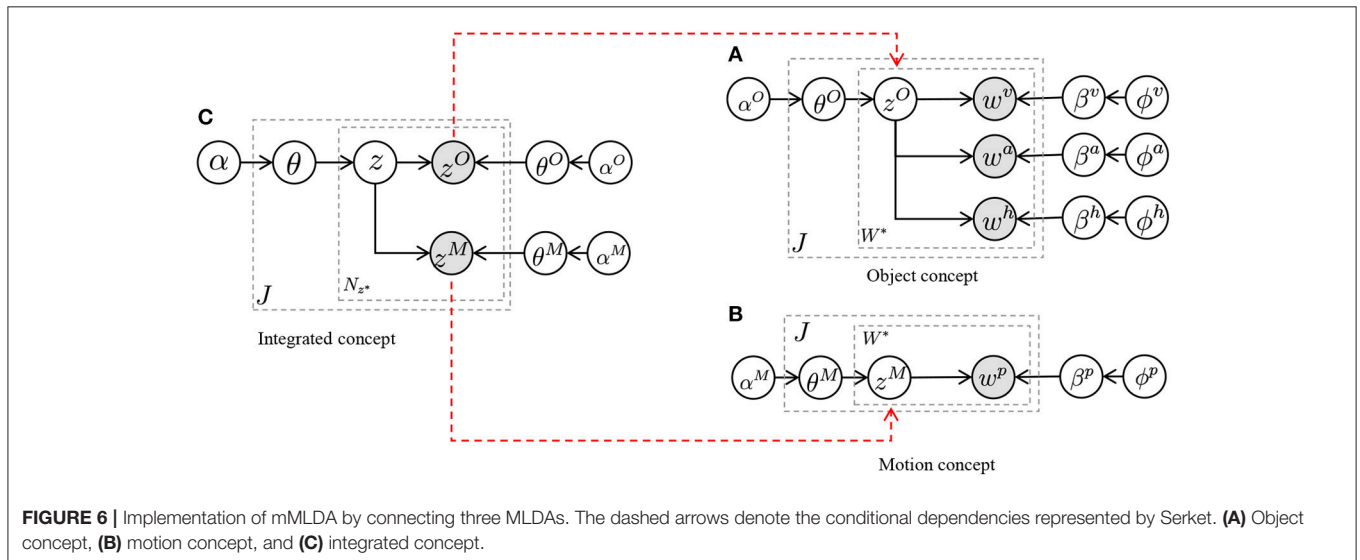
$$z_{jmn}^C \sim P(z^C | W^m, Z_{-jmn}) P(z^C | \hat{z}_j^M, \hat{z}_j^O), \tag{25}$$

where  $W^m$  represents all the information of modality  $m$ , and  $Z_{-jmn}$  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  $P(z^C | W^m, Z_{-jmn})$  in the normal MLDA, they were also sampled using  $P(z^C | \hat{z}_j^M, \hat{z}_j^O)$ . Therefore, all the latent variables were learned in a complementary manner. From the sampled variables, the parameters of  $P(z_j^O | w_j^v, w_j^a, w_j^h)$  and  $P(z_j^M | w_j^p)$  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



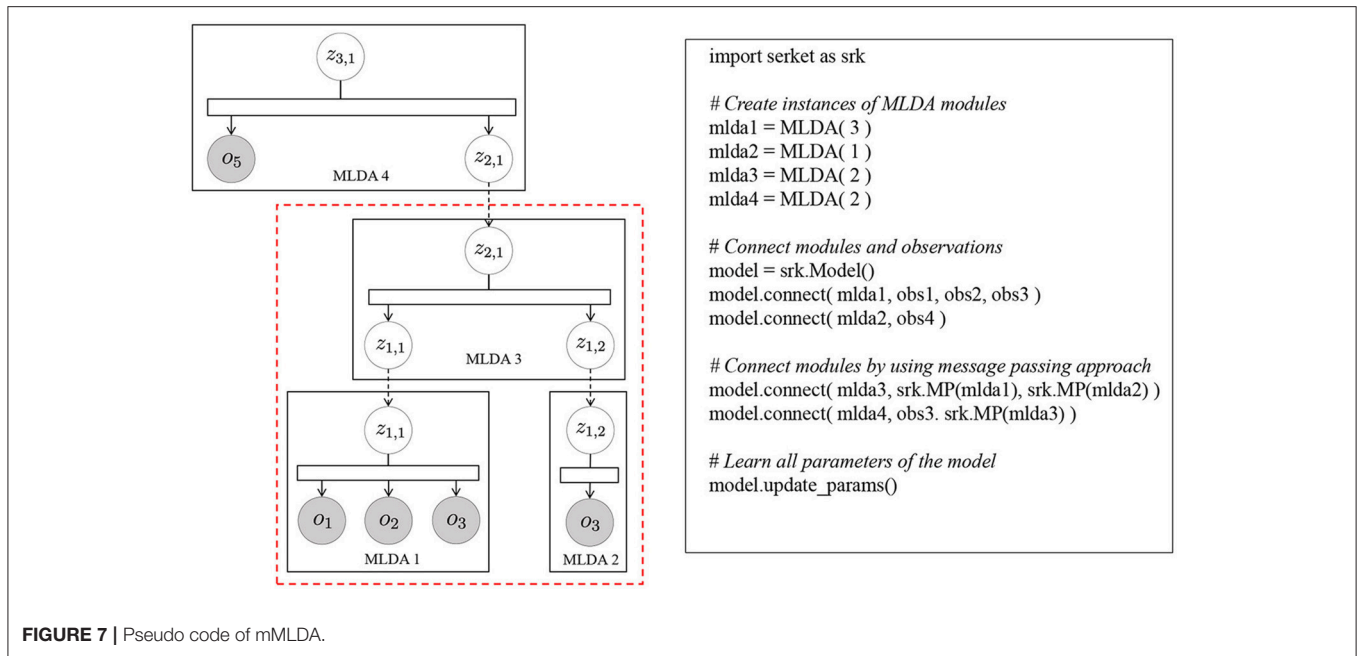


FIGURE 7 | Pseudo code of mMLDA.

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 MLDA 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:

$$z_{5,1} \sim P(z|\theta_5) \tag{26}$$

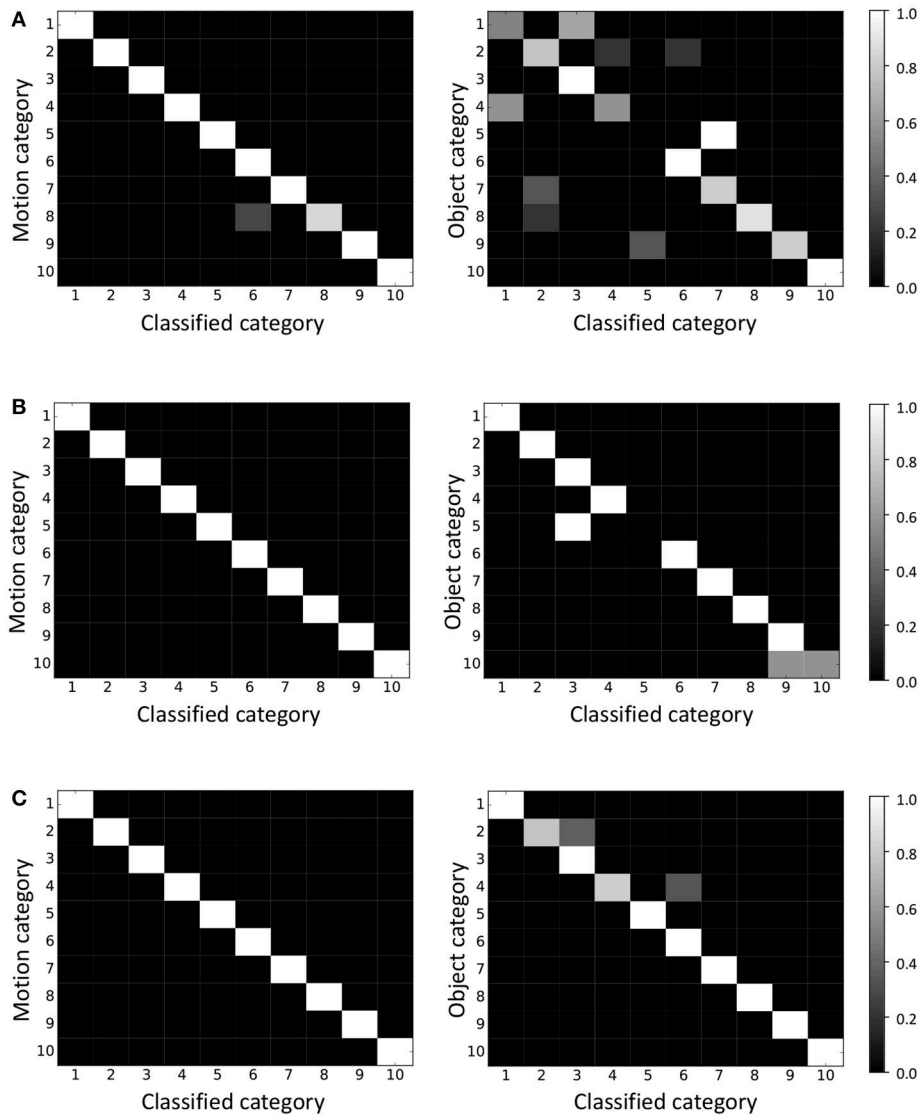
$$o_5 \sim P(o|\phi_{z_{5,1}}) \tag{27}$$

for  $m = 4$  to 1:

$$z_{m,1} \sim P(z|z_{m+1,1}, \theta_m) \tag{28}$$

$$o_m \sim P(o|\phi_{z_{m,1}}) \tag{29}$$

where  $m$  denote the index of hierarchies, and the number of categories of all modules was 10.  $\theta_m$  and  $\phi_z$  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.

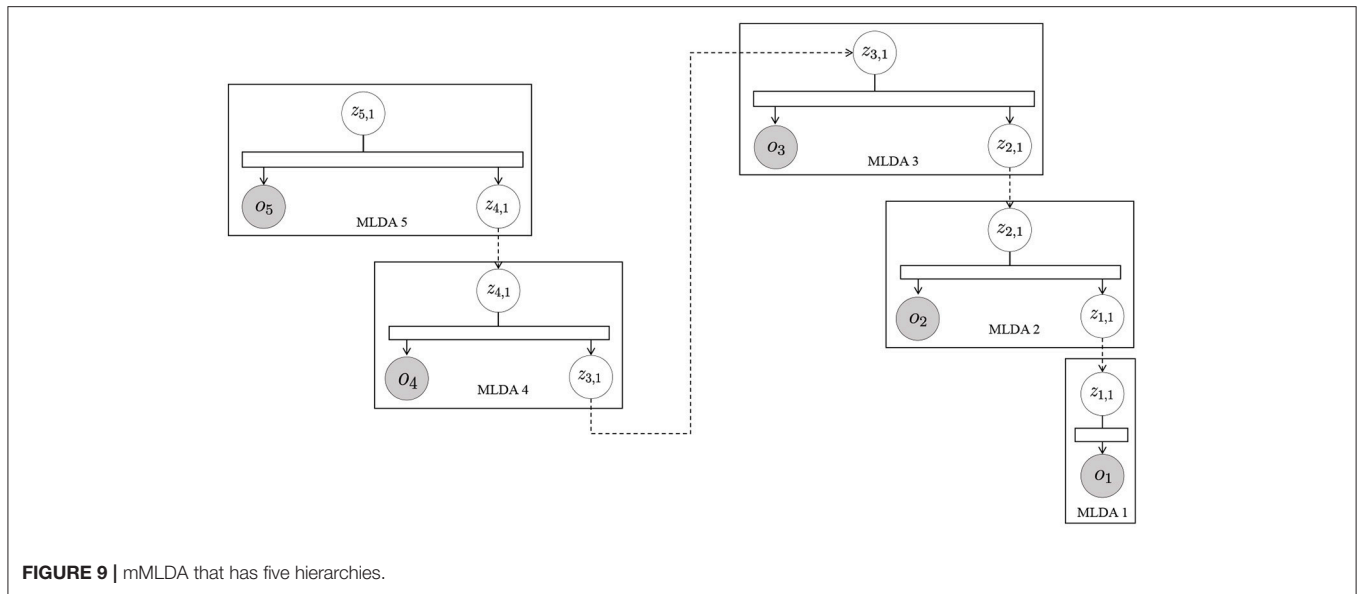


FIGURE 9 | mMLDA that has five hierarchies.

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  $o$  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  $o$  but is accurately recognized by utilizing the inferred words  $s$  from the multimodal information perceived by the robot.

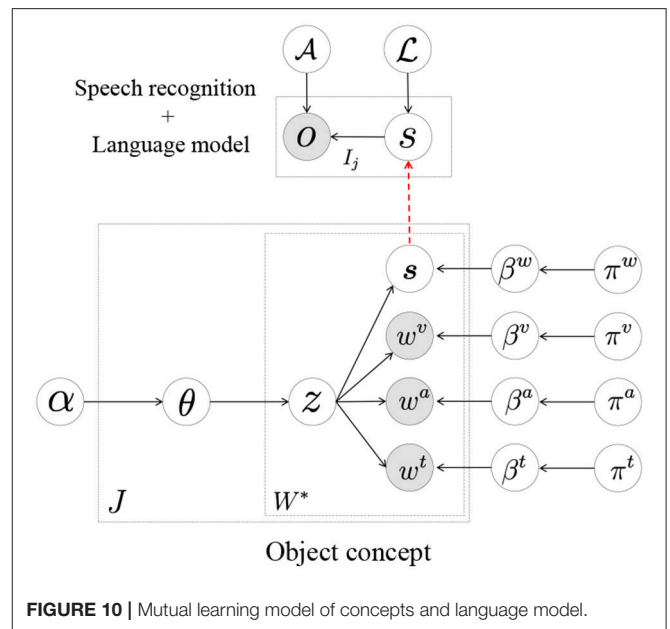


FIGURE 10 | Mutual learning model of concepts and language model.

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  $O$  are recognized by using a language model whose parameter is  $\mathcal{L}$  and the probability  $P(S|O, A, L)$  that the word sequences  $S$  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.  $s_j^{(l)}$  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  $\mathbf{S} = \{s_j^{(l)} | 1 \leq l \leq L, 1 \leq j \leq J\}$  can be obtained.

$$\mathbf{S} \sim P(\mathbf{S} | \mathbf{S}', \mathcal{L}). \quad (30)$$

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  $\Theta$ :

$$\hat{s}_j \sim P(s_j^{(l)} | \mathbf{w}_j^v, \mathbf{w}_j^a, \mathbf{w}_j^t, \Theta). \quad (31)$$

The selected samples  $\hat{s}_j$  are considered as words that can represent multimodal information. Then, the MLDA parameters are updated using a set of these words  $\hat{\mathbf{S}} = \{\hat{s}_j | 1 \leq j \leq J\}$  and a set of multimodal information  $\mathbf{W}^v, \mathbf{W}^a, \mathbf{W}^t$  by utilizing Gibbs sampling.

$$\Theta = \operatorname{argmax} P(\hat{\mathbf{S}}, \mathbf{W}^v, \mathbf{W}^a, \mathbf{W}^t | \Theta). \quad (32)$$

Moreover,  $\hat{\mathbf{S}}$  is sent to the speech recognition model, and the parameter  $\mathcal{L}$  of the language model is updated.

$$\mathcal{L} = \operatorname{argmax} P(\hat{\mathbf{S}}' | \mathcal{L}), \quad (33)$$

where  $\hat{\mathbf{S}}'$  denotes strings obtained by connecting words in  $\hat{\mathbf{S}}$ . 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  $\mathbf{S}'_0$  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:

$$\mathbf{S}_0 \sim P(\mathbf{S} | \mathbf{S}'_0, \mathcal{L}). \quad (34)$$

Alternatively, method (b) was implemented by Serket, and the concepts and language model were learned mutually through the shared latent variable  $\Theta$ .

**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.

$$P = \frac{N_{TP}}{N_{TP} + N_{FP}}, \quad (35)$$

$$R = \frac{N_{TP}}{N_{TP} + N_{FN}}, \quad (36)$$

$$F = \frac{2RP}{R + P}, \quad (37)$$

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

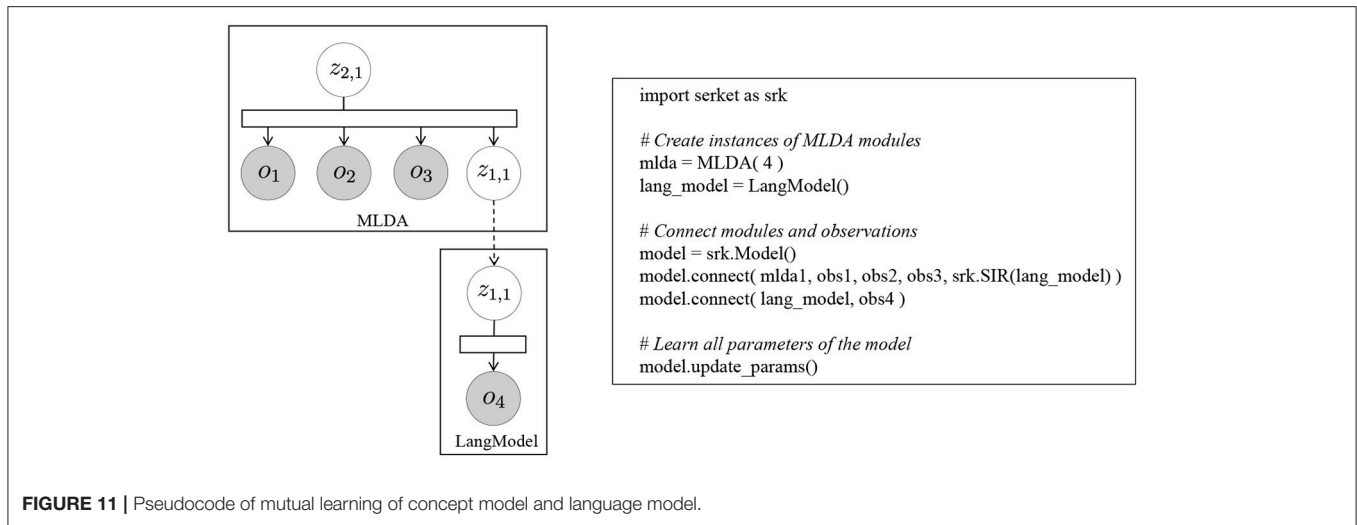


FIGURE 11 | Pseudocode of mutual learning of concept model and language model.

TABLE 3 | Accuracies of speech recognition, segmentation, and object classification.

Methods	(i) Speech recognition	(ii) Segmentation			(iii) Object classification
		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.

## 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: 10.1145/1667053.167056
- 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.
- 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>

- 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.
- Lallem, 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 mllda and nplym," 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.

Copyright © 2018 Nakamura, Nagai and Taniguchi. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.