

Skill Learning and Inference Framework for Skilligent Robot

Sang Hyoung Lee and Il Hong Suh

Abstract—To achieve a certain task, a skilligent robot should be able to learn the skills embedded in that task. Furthermore, the robot should be able to infer such skills to handle uncertainties and perturbations, since most robot tasks are usually daily-life tasks that include many unexpected situations. Therefore, we propose a unified skill learning and inference framework. The framework includes six processing modules: 1) a human demonstration process, 2) an autonomous segmentation process, 3) a dynamic movement primitive learning process, 4) a Bayesian network learning process, 5) a motivation graph construction process, and 6) a skill-inferring process. Based on the framework, the robot learns and infers situation-adequate and goal-oriented skills to handle uncertainties and human perturbations. To show the validity of our framework, some experimental results are illustrated using a robot arm that performs a ‘tea service’ task.

I. INTRODUCTION

It is a challenge for robots to handle unexpected situations using previously learned skills. Here, skills are defined as motion primitives and their meanings (i.e., activation conditions) embedded in tasks, since a task is usually characterized by a sequential combination of motion primitives. Therefore, a skilligent robot should learn and recombine such skills under uncertainties and perturbations [1].

To learn and infer such skills in order to handle unexpected situations, the robot should possess the following three abilities: i) Learning motion primitives: The robot should be able to learn motion primitives embedded in a task. Here, motion primitives have to be formalized to guarantee goal achievement under uncertainties and perturbations. ii) Learning the meanings of motion primitives: The robot should be able to learn the relationships between task-relevant motion primitives and task-relevant entities for activating the motion primitives. iii) Recombining skills: The robot should be able to sequentially infer skills in accordance with current and goal situations in a dynamic environment.

In this paper, to acquire the three abilities, the robot first obtains training data that include motion trajectories and trajectories of task-relevant entities by human demonstrations, as shown in Fig. 1(a). Next, an autonomous segmentation process is used for learning the motion primitives embedded in the continuous motion trajectories. To date,

*This work was supported by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government(MEST) (NRF-M1AXA003-2011-0028353)

S. H. Lee is with the Education Center for Network-based Intelligent Robotics, Hanyang University, Seoul, Korea zelog@hanyang.ac.kr

I. H. Suh is with the Department of Computer Science and Engineering, Hanyang University, Seoul, Korea ihsuh@hanyang.ac.kr, All correspondence should be addressed to I. H. Suh

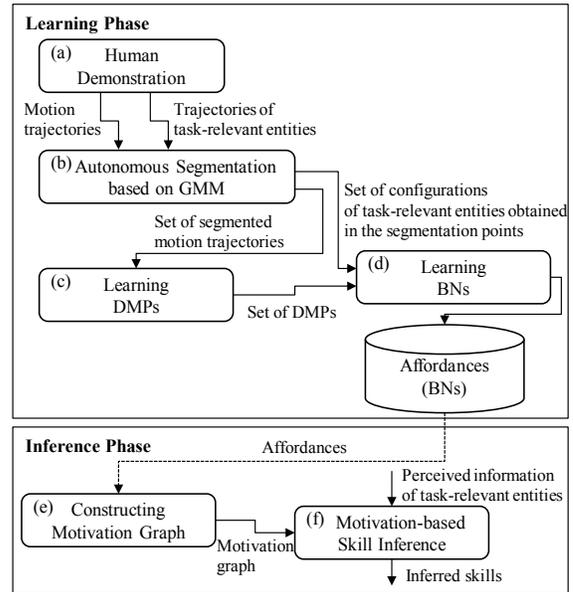


Fig. 1. Six processes for the three abilities of the skill learning and inference framework: (a) human demonstration process, (b) autonomous segmentation process, (c) motion primitive learning process, (d) probabilistic affordance learning process, (e) motivation graph construction process, and (f) motivation-based skill inference process.

many researchers have proposed autonomous segmentation approaches [2]–[5]. Even though these approaches learn unknown motion primitives, they still face certain constraints that are predefined or tuned, such as fixed intervals, window sizes, fixed times, and threshold values. It is not easy to predefine and tune such constraints according to the types of tasks and training data. It is, therefore, important to autonomously learn the motion primitives embedded in such tasks without predefining and/or tuning such parameters. To do this, segmentation points are estimated from a Gaussian Mixture Model (GMM) learned using motion trajectories, as shown in Fig. 1(b). Here, the motion trajectories and the trajectories of task-relevant entities are segmented by the estimated segmentation points.

To represent the motion primitives, the segmented motion trajectories are formalized as Dynamic Movement Primitives (DMPs) proposed in [6], as shown in Fig. 1(c). The DMPs guarantee the convergence of their goals under various uncertainties and perturbations such as changes in the initial and goal configurations as well as human intentions. To handle these uncertainties, the probabilistic affordances are formalized to learn meanings that can activate the motion

primitives, as shown in Fig. 1(d). The probabilistic affordances can be represented as Bayesian Networks (BNs) using the configuration of task-relevant entities and the labels of DMPs as in our earlier work [7].

In the decision phase, to achieve the assigned robot tasks, the skills are combined as a sequence that satisfies the current and goal configurations. For this, the probabilistic affordances are arranged based on a task-dependent motivation graph that presents a nominal sequence of the given task, as shown in Fig. 1(e). Finally, the situation-adequate and goal-oriented skills are sequentially inferred based on the motivation values calculated from the motivation graph and affordances as a behavior-based control approach. Based on these motivation values, it is possible to infer fully connected transitions between skills without designing or learning their transition models.

To date, there has been a significant amount of research into the inference of sequential combination of skills using predefined reactive plans for achieving given tasks [8]–[10]. As one branch of this research, motivation has been used to recommend skills based on current internal states of a robot. Existing approaches have used motivation to implicitly infer fully connected transitions of internal states by perceiving a current stimulus. Thus, the approaches can infer various sequential combinations of skills based on the transitions of internal states according to the given situations. However, in these approaches, it is difficult to infer goal-oriented skills using the implicit transitions of internal states. Furthermore, it is not easy to infer skills using motivation in the real world, as the environment includes various sensors and action noise in addition to limited perception. In Fig. 1(f), the motivation-based skill inference process infers goal-oriented and situation-adequate skills based on the motivation values.

The main contributions of this paper are (i) a unified skill learning and inference framework to handle uncertainties and perturbations, (ii) a probabilistic representation of skills using BNs, (iii) a methodology of clustering motion primitives based on effect equivalence, and (iv) an experimental validation using a daily-life task performed by a robot arm.

The rest of this paper is organized as follows: Section II-A revisits the autonomous segmentation process and the learning process of DMPs. Section II-B presents the details of learning probabilistic affordances and inferring skills using affordances and a motivation graph. Section III presents the experimental results for a robot arm performing a “tea service” robot task. Section IV discusses the proposed framework. Finally, Section V presents our conclusions and plans for future research.

II. SKILL LEARNING AND INFERENCE FRAMEWORK

A. Learning Motion Primitives

1) *Autonomous Segmentation Process*: The continuous motion trajectories of a robot arm, $\mathbf{X} \in \mathbb{R}^{(D+1) \times N}$, are first extracted from human demonstrations. Here, $(D+1)$ denotes a D -dimensional spatial variable and a one-dimensional temporal variable, and N is the length of the trajectories. In the autonomous segmentation process, it is important to

divide the trajectories reasonably. In this context, all the Gaussians of a GMM provide important information on segmentation of the motion trajectories [11], because each Gaussian distribution involved in a GMM encodes a portion that indicates a quasi-linear segment in hyperspace. The non-linear motion trajectories demonstrated by humans can be characterized better when the GMM is modeled by as many Gaussians as possible without overfitting. The Bayesian Information Criterion (BIC) algorithm can estimate the number of Gaussians while resolving the overfitting problem based on the criterion of minimum description length. In the BIC algorithm, however, the number of Gaussians depends on the dimensionality of the motion trajectories. The GMM tends to contain many Gaussians in the low-dimensional spaces reduced by Principal Component Analysis (PCA) under the assumption that essential motion trajectory information is not eliminated. As a result, the non-linear motion trajectories are represented better by the GMM estimated in the low-dimensional space, as it uses more Gaussians than the original space [12].

In this context, PCA transforms the motion trajectories, except for the temporal variable, as the low-dimensional variable, $\Psi \in \mathbb{R}^{(D'+1) \times N}$. Here, D' denotes the D' -dimensional spatial variable transformed by PCA. The dimensionality of the transformation matrix is adaptively determined in the range 0.9–1.0 for the sum of the eigenvalues while automatically estimating the number of Gaussians using the BIC algorithm. This is because the GMM should be modeled using as many Gaussians as possible within the range in which the essential information of the motion trajectories is not eliminated, as mentioned earlier.

The GMM is modeled using the motion trajectories $\Psi \in \mathbb{R}^{(D'+1) \times N}$ in the reduced dimensional space based on the BIC and Expectation-Maximization algorithms. The GMM is defined as

$$P(\Psi) = \sum_{i=1}^K w_i \cdot N(\Psi | \mu_i, \Sigma_i), \quad (1)$$

where w_i , μ_i , and Σ_i refer to the priors, means, and covariances, respectively, of the i^{th} Gaussian. Here, the GMM is initialized using the k-means clustering algorithm.

Based on the GMM, the segmentation points are estimated in temporally overlapping points between two consecutive Gaussians that are temporally adjacent in a GMM. Before estimating these points, the means and covariances of the GMM are divided into temporal and spatial components. The mean and covariance matrices of the i^{th} Gaussian are represented as

$$\mu_i = (\mu_{i,t} \mu_{i,\Psi'}), \quad (2)$$

$$\Sigma_i = \begin{pmatrix} \Sigma_{i,t} & \Sigma_{i,t\Psi'} \\ \Sigma_{i,\Psi't} & \Sigma_{i,\Psi'} \end{pmatrix}, \quad (3)$$

where t and Ψ' refer to the one-dimensional temporal variable and D' -dimensional spatial variable in the $(D'+1)$ -dimensional variable Ψ . The temporally overlapping points are extracted by estimating the weights of the Gaussians

along the time component of the GMM. In other words, the temporally overlapping points are detected by the intersections of the weights calculated by

$$h_i(t) = \frac{w_i N(t; \mu_{i,t}, \Sigma_{i,t})}{\sum_{k=1}^K w_k N(t; \mu_{k,t}, \Sigma_{k,t})}, \quad (4)$$

where i and K refer to the index and the total number of Gaussians, respectively. As a result, K segmentation points are estimated from the GMM.

2) *Motion Primitive Learning Process*: The set of segmented motion trajectories is formalized as DMPs for guaranteeing their goal achievement, as mentioned earlier. A DMP is represented by integrating the set of second-order differential equations as

$$\tau \dot{v} = K(g - x) - Dv - K(g - x_0)s + Kf(s), \quad (5)$$

$$\tau \dot{x} = v, \quad (6)$$

where x , v , x_0 , and g denote the current position, current velocity, initial position, and goal position, respectively. Here, τ , K , and D are a temporal scaling constant, spring constant, and damping constant, respectively. Finally, $f(s)$ is an external force term for controlling the motion trajectories defined as a set of non-linear equations. The non-linear equations are defined as

$$f(s) = s \frac{\sum_i w_i \psi_i(s)}{\sum_i \psi_i(s)}, \quad (7)$$

where $\psi_i(s)$ is the i^{th} Gaussian basis function, and w_i is a learnable weight. Here, $\psi_i(s)$ is also defined as

$$\psi_i(s) = e^{-h_i(s-c_i)^2}, \quad (8)$$

where c_i and h_i are the center and the width of the i^{th} Gaussian basis function. Finally, s is a monotonically decreasing function ranging from 1 to 0 during the execution of the motion primitive as

$$\tau \dot{s} = -\alpha s, \quad (9)$$

where α and τ are constants for adjusting the rate of change. The external force is estimated by regulating the weight w_i of (7) in order to generate complex trajectories to achieve the goal of the motion primitive. The weights are estimated by locally weighted regression, as this is a well-known regression problem. As a result, DMPs guarantee the achievement of their goals against perturbations and changes in a dynamic environment.

B. Skill Learning and Inference

The motion primitives are represented using only the segmented motion trajectories. To achieve a task, the motion primitives should be activated according to their current and goal configurations. Therefore, a skillful robot should be able to infer situation-adequate and goal-oriented motion primitives from their set to deal with the task in the current situation. Such a motion primitive is one whose effect is the closest to achieving the goal situations among all the motion primitives that can be executed in the current situation.

To infer the motion primitives sequentially, the robot first has to learn the relationships between task-relevant entities and motion primitives, and it then has to infer the sequence of the motion primitives with respect to situation-adequateness and goal-orientedness under uncertainties and perturbations.

In this context, a probabilistic affordance is formulated as a BN using the configurations of task-relevant entities acquired by the segmentation process and DMPs. To learn the BN, the configurations of the task-relevant entities need to be clustered based on certain criteria of the motion primitives. Affordances are then arranged based on a motivation graph acquired from a nominal sequence disregarding any uncertainties. In the skill inference process, the robot calculates and propagates motivation values using the probabilistic affordances and a motivation value propagation algorithm. The task is finally achieved by sequentially inferring the motion primitives based on the motivation values.

1) *Effect-based Clustering*: Affordance is a property that indicates possibilities for a motion primitive, perceived in a direct and immediate way without sensory processing. Among many other definitions, Sahin *et al.* defined affordance as an acquired relationship between a motion primitive and a task-relevant entity in the environment, such that the application of motion primitives to entities generates a certain effect [13]. To learn probabilistic affordances, there should be substantial training data. The training data are acquired by multiple human demonstrations, and the set of training data needs to be clustered according to its characteristics. ‘‘Effect’’ can be used as an excellent criterion for clustering the training data. Clustering is the grouping of training data that generate the same effect by the motion primitives (i.e., effect equivalence).

To cluster the training data based on effect, let us look at the training data. The training data are defined as a set of three-tuples segmented in the autonomous segmentation process. A set of three-tuples is defined as $\mathbb{T} = \{\mathbf{T}_1, \dots, \mathbf{T}_N\}$. Here, N indicates the number of training data. A three-tuple is defined as

$$\mathbf{T}_i = \langle \mathbf{Z}_i, \mathbf{A}_i, \mathbf{Y}_i \rangle, \quad (10)$$

where \mathbf{Z}_i and \mathbf{Y}_i are sets of variables that represent the configurations of task-relevant entities perceived in the segmentation points before and after the motion primitives \mathbf{A}_i acquired by the i^{th} segmentation point, and they are defined as $\mathbf{Z}_i = \{\mathbf{z}_i^1, \dots, \mathbf{z}_i^M\}$ and $\mathbf{Y}_i = \{\mathbf{y}_i^1, \dots, \mathbf{y}_i^M\}$. Here, M is the number of variables that represent the configurations of task-relevant entities, \mathbf{A}_i is defined as \mathbf{a}_i , which is a variable, since a robot can execute only a single motion primitive at a time. The configurations of task-relevant entities associated with the motion primitives that have the same effects are clustered according to effect-based clustering. For this, the effect \mathbf{E}_i can be calculated by various strategies as

$$\mathbf{E}_i := f(\mathbf{Y}_i, \mathbf{Z}_i), \quad (11)$$

where f is a function for calculating the effect using \mathbf{Y}_i and \mathbf{Z}_i . In this paper, the function f is defined as operator \ominus , which calculates the difference by subtracting \mathbf{Z}_i

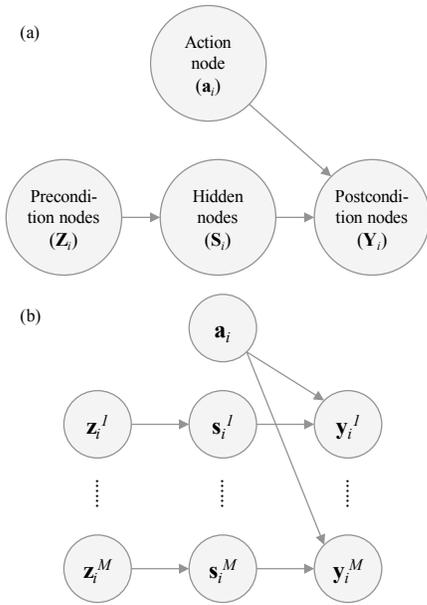


Fig. 2. Representations of an affordance: (a) causation of an affordance and (b) Bayesian network representing a probabilistic affordance.

from \mathbf{Y}_i . Effect \mathbf{E}_i is a set of effect values calculated as $\mathbf{E}_i = \{\mathbf{e}_i^1, \dots, \mathbf{e}_i^M\} = \{\mathbf{y}_i^1 - \mathbf{z}_i^1, \dots, \mathbf{y}_i^M - \mathbf{z}_i^M\}$. In addition, the effect values are substituted in the direction of the effect values, because using directions can improve the generality in comparing the similarities of effects over wider ranges. For example, let us consider an example in which a robot approaches an object from an initial position that is different from the goal position. The robot may execute the other motion primitives for approaching the object, such as stretching its arm by 20 or 30 cm. Although the motion primitives are physically different, their effects are the same (i.e., closed) when using the direction of effect values. In this case, the direction of the effect is calculated as

$$\mathbf{e}_i^j = \begin{cases} 1 & \text{if } \mathbf{y}_i^j - \mathbf{z}_i^j > 0 \\ -1 & \text{if } \mathbf{y}_i^j - \mathbf{z}_i^j < 0 \\ 0 & \text{if } \mathbf{y}_i^j - \mathbf{z}_i^j = 0 \end{cases} \quad (12)$$

where \mathbf{z}_i^j and \mathbf{y}_i^j are the variables that represent the i^{th} are the variables that represent the j^{th} training data. As a result, the training data are finally clustered by exactly comparing the set of directions.

2) *Probabilistic Affordance Learning Process*: Probabilistic affordances are learned using a set of clustered training data. To represent the affordances as BNs, a relationship between precondition (\mathbf{Z}_i), motion primitive (\mathbf{a}_i), and post-condition (\mathbf{Y}_i) is formulated. To facilitate this, the causation between variables has to be determined, as shown in Fig. 2. Hidden variables \mathbf{s}_i are inserted in the representation of BN since the hidden variables can include the sensitivity (or uncertainties) of sensors for perceiving the information about task-relevant entities. These are defined as $\mathbf{S}_i = \{\mathbf{s}_i^1, \dots, \mathbf{s}_i^M\}$. Here, the precondition and hidden variables are mutually independent of variable \mathbf{a}_i , and the post-condition variables

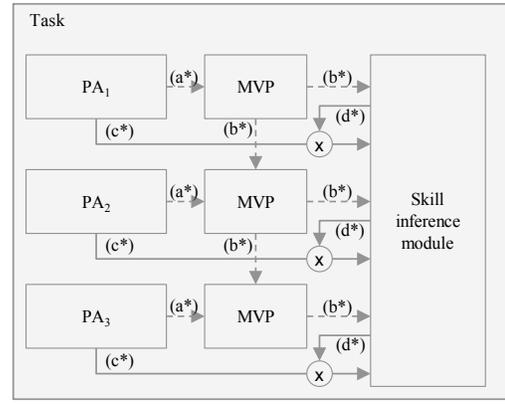


Fig. 3. Illustration of a motivation graph: (a) action probabilities of probabilistic affordances, (b) motivation values, (c) labels of DMPs, and (d) triggering signals for DMPs. Here, “PA” and “MVP” indicate probabilistic affordances and motivation value propagation modules.

are dependent on the hidden and action variables. Finally, the hidden variables are only dependent on the precondition variables. That is, the BN assumes that \mathbf{z}_i^j , \mathbf{s}_i^j , and \mathbf{y}_i^j are independent of all other \mathbf{z}_i^k , \mathbf{s}_i^k , and \mathbf{y}_i^k for $j \neq k$, and that \mathbf{y}_i^j is only dependent on \mathbf{z}_i^j and \mathbf{a}_i as shown in Fig. 2(b). Finally, \mathbf{s}_i^j is only dependent on \mathbf{z}_i^j . In this paper, all affordances are represented using the same BN structure, as shown in Fig. 2.

In BN, the parameters are learned using the training data that are clustered according to their effects. These consist of $P(\mathbf{Z}_i)$, $P(\mathbf{a}_i)$, $P(\mathbf{S}_i|\mathbf{Z}_i)$, and $P(\mathbf{Y}_i|\mathbf{a}_i, \mathbf{S}_i)$ as per the structure of Fig. 2. $P(\mathbf{Z}_i)$ and $P(\mathbf{a}_i)$ are learned as conditional probability tables or probability distributions using the frequencies of \mathbf{Z}_i and \mathbf{a}_i in the training data. In $P(\mathbf{Y}_i|\mathbf{a}_i, \mathbf{S}_i)$, the variables \mathbf{S}_i and \mathbf{Y}_i can also be discrete (e.g., contact sensor) or continuous (e.g., distance sensor). In addition, \mathbf{a}_i is a discrete variable that has the labels of DMPs. The BN that includes both discrete and continuous variables is referred to as a hybrid BN. The most common choice to represent a hybrid BN is the linear Gaussian distribution, in which the child has a Gaussian distribution whose mean μ varies linearly with the value of the parent, and whose standard deviation σ is fixed. When \mathbf{s}_i^j and \mathbf{y}_i^j are continuous variables, the distribution of $P(\mathbf{y}_i^j|\mathbf{a}_i, \mathbf{s}_i^j)$ can be expressed as

$$\begin{aligned} \hat{m}_i^j &= P(\mathbf{y}_i^j|\mathbf{a}_i = a, \mathbf{s}_i^j) = N(a \cdot \mathbf{s}_i^j + b, \sigma^2)(\mathbf{y}_i^j) \\ &= \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\mathbf{y}_i^j - (a\mathbf{s}_i^j + b)}{\sigma}\right)^2}, \end{aligned} \quad (13)$$

where \mathbf{s}_i^j and \mathbf{y}_i^j are the j^{th} hidden and post-condition variable in the i^{th} cluster, respectively, and \mathbf{a}_i is a discrete variable that represents the labels of the DMPs in the i^{th} cluster (or affordance). $P(\mathbf{y}_i^j|\mathbf{a}_i = a, \mathbf{s}_i^j)$ can specify all motion primitives a , because \mathbf{a}_i is a discrete variable that is handled by explicit enumeration. The parameters of (13) are individually defined as

$$a = 1, b = \frac{\sum_{i=1}^N (\mathbf{y}_i^j - \mathbf{s}_i^j)}{N}, \sigma = \frac{\sum_{i=1}^N ((\mathbf{y}_i^j - \mathbf{s}_i^j) - \mu)^2}{N}, \quad (14)$$

where N is the number of training data.

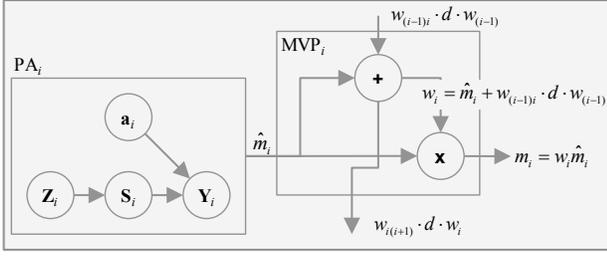


Fig. 4. Visualization of motivation value propagation algorithm.

Finally, in $P(S_i|Z_i)$, S_i and Z_i can also discrete or continuous variables. When both z_i^j and s_i^j are continuous variables, the distributions are assumed to be Gaussian distributions.

3) *Inference Process based on Probabilistic Affordance and Motivation Graph*: The robot can infer motion primitives in a given situation using the affordances as they provide probability values for all the motion primitives. However, the affordance is not suitable for accomplishing a task that requires the motion primitives to be performed in sequence. To achieve a task, the robot should be able to infer situation-adequate and goal-oriented motion primitives. For this, the affordances are arranged in a sequential structure (i.e., motivation graph), as shown in Fig. 3. Motivation values are calculated using the affordances and a motivation value propagation algorithm based on the motivation graph.

The motivation graph can be extracted using two methods: First, the affordances have relationships between preconditions, motion primitives, and post-conditions. Hence, the graph for achieving a given task can be extracted from the similarities between the preconditions/post-conditions of each affordance and the goal situation of the task. Next, the motivation graph can be generated by extracting the most frequent sequence of the motion primitives for achieving the tasks in the sequence that is represented by the task. In this paper, the motivation graph is acquired using the second method for calculating the frequency of transitions between the affordances in the training data.

A motivation value propagation algorithm for calculating motivation values is shown in Fig. 4. Here, the i^{th} affordance outputs an action probability \hat{m}_i to the Motivation Value Propagation (MVP) module. The MVP module propagates the motivation values calculated by the action probabilities of affordances. The motivation value of the i^{th} MVP is defined as

$$m_i = w_i \cdot \hat{m}_i, \quad (15)$$

where w_i is the weighting value for regulating action probability \hat{m}_i (here, $\hat{m}_i = \prod_{j=1}^M \hat{m}_i^j$) according to goal-orientedness. Note that m_i is increased or decreased from \hat{m}_i by the weight w_i . Here, w_i is defined as

$$w_i = \hat{m}_i + w_{(i-1)i} \cdot d \cdot w_{(i-1)}, \quad (16)$$

where \hat{m}_i is an action probability of the i^{th} affordance and $w_{(i-1)i}$ is the weighting value that represents the relationship between the $(i-1)^{th}$ and the i^{th} affordances, $w_{(i-1)}$ is the



Fig. 5. Illustrations of a robot arm and a motion capture system: (a) Katana robot arm and four objects (i.e., cup, kettle, teabag, and human hand) and (b) Experimental environment containing a robot arm and a motion capture system.

weighting value of the $(i-1)^{th}$ affordance, and d is a decay factor. This algorithm is similar to the spreading activation algorithm that propagates activation values of limited source nodes to all associated nodes based on already-learned weighting values. However, in the MVP algorithm, the motivation values are all source nodes, and the weighting values are calculated during runtime. The weight w_i is determined by the action probability and motivation value of the upper affordances. Moreover, the algorithm satisfies situation-adequateness as well as goal-orientedness [14].

This algorithm tends to increase the motivation values of reliable motion primitives and leads to comparatively more goal-oriented motion primitives in the current situation.

III. EXPERIMENTAL RESULTS

To validate the skill learning and inference framework, a “tea service” task was performed using a robot arm. In the task, a robot performs the task using a nominal sequence as follows: the robot first places a teabag inside a cup after picking it up. Next, it pours water into the cup using a kettle. Finally, the cup of tea is delivered to a human.

The motion (i.e., joint and end-effector) trajectories of Katana (developed by Neuronics) were recorded at 25 Hz. Moreover, the trajectories of task-relevant entities were acquired by using twelve V100:R2 motion capture cameras developed by Optitrack, as shown in Fig. 5. The training data were extracted from 50 demonstrations using a kinesthetic teaching method. In detail, five demonstrations were executed for ten different initial and goal configurations of the robot and four objects (i.e., cup, teabag, kettle, and human hand). The task-relevant entities were selected as follows: 1) Weight: This information is measured as the weight taken by the arm of the robot. 2) Contact: This information is measured by whether the robot was in contact with an object. 3) Relative distance: This information is measured by the motion-capture system to recognize the distance between the robot and entities. These entities were defined as the variables of BNs, as shown in Fig. 6. Of the three types of variables, two were defined as discrete variables, with the distance variables being continuous. Before learning the parameters of the BNs, the training data acquired using 50 demonstrations were first segmented by the autonomous segmentation process, and each collection of training data was divided into 14 segments. As a result, 700 (=50 demonstrations \times 14

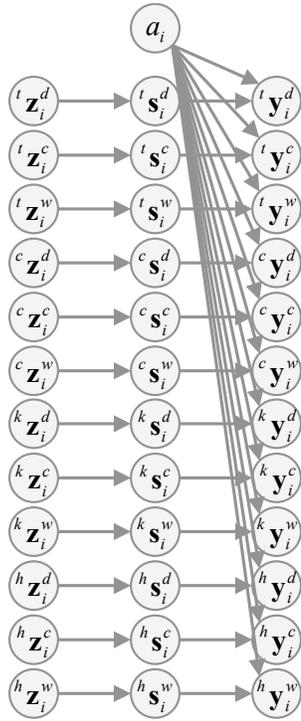


Fig. 6. Structure of the probabilistic affordances in executing the tea service task. The structure of all affordances is the same. Here, left superscripts t , c , k , and h indicate a teabag, a cup, a kettle, and a human hand, respectively, and right superscripts d , c , and w indicate the relative distance, contact, and weight, respectively. Finally, right subscripts i indicate the i^{th} affordance.

segments) training data were acquired in the segmentation process.

To learn motion primitives, 14 DMPs were formulated using the 14 segmented motion trajectories obtained in a representative training data as shown in Fig. 7. Next, the 700 pieces of training data were clustered using the effect-based clustering method. Table I lists the results of clustering the training data based on the effects. Although the training data were clustered into 31 groups, there were no clusters with meanings that are incompliant with each other. Even though the training data of [ApproachingTeabag] were divided into six groups, for example, they were not included in groups that contain different meanings. These results are sufficient for learning the probabilistic affordances. The meanings were attached to identify the effects, but the robot did not need to know the semantics to achieve the task.

The parameters of the affordances were learned based on the network structure and the clustered training data. Here, the number of training data was determined in the case that the results of Bayesian inference between preconditions/post-conditions of the affordances and the goal situations of the task are greater than 85%. The robot computed and propagated motivation values according to the motivation graph as shown in Fig. 8 using (13) and (15). Ultimately, the robot selected situation-adequate and goal-oriented motion primitives. Fig. 7 shows the results of the experiments where the task is executed based on the affordances, motivation graph, and motivation value propagation algorithm according

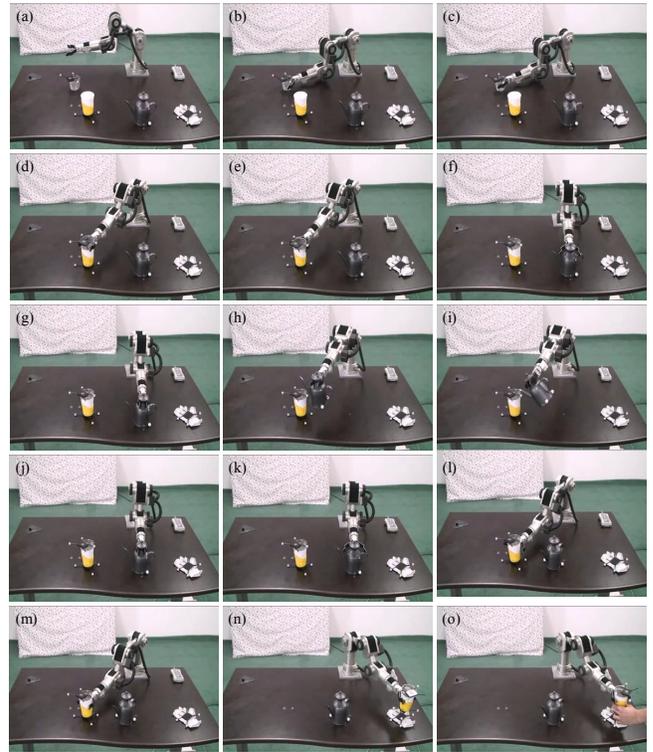


Fig. 7. Illustrations of fourteen DMPs: (a) initial configuration, and DMPs (b) [ApproachingTeabag], (c) [GraspingTeabag], (d) [DeliveringTeabag], (e) [ReleasingTeabag], (f) [ApproachingKettle], (g) [GraspingKettle], (h) [DeliveringKettle], (i) [TiltingKettle], (j) [PlacingKettle], (k) [ReleasingKettle], (l) [ApproachingCup], (m) [GraspingCup], (n) [DeliveringCup], and (o) [ReleasingCup](i.e., goal configuration).

to the nominal sequence of the task. To achieve the tasks, each affordance was finished after satisfying the goal of the selected motion primitives, because the motion primitive is formalized as a DMP.

Quantitative results for these affordances are presented in Fig. 9(a), which shows the success rates according to the number of training data. The affordances were executed with success rates greater than 90%. About 10% of failures occurred because the objects escape from the workspace of the robot arm. Fig. 9(b) shows the success rates of the task when using motivation graphs to arrange affordances, which affects the feasibility of accomplishing the task. In Fig. 9(b), the bars indicate the results of the following motivation graphs. Type 1 (nominal graph): b)→c)→d)→e)→f)→g)→h)→i)→j)→k)→l)→m)→n)→o), Type 2 (abnormal): b)→c)→d)→e)→j)→k)→l)→m)→f)→g)→h)→i)→o), Type 3 (abnormal): n)→b)→m)→c)→l)→d)→k)→e)→j)→f)→i)→g)→h)→o), and Type 4 (abnormal): o)→n)→m)→l)→k)→j)→i)→h)→g)→f)→e)→d)→c)→b)→a) (see Fig. 7 to understand the meanings of a)–o)). Even though the performances were significantly affected by the arrangement of the affordances according to the motivation graph, the task could be achieved the higher than the averaged 75% when even using abnormal motivation graphs.

Several experiments were conducted to verify the frame-

TABLE I
RESULTS OF CLUSTERING TRAINING USING EFFECTS AND THE
MEANINGS OF 31 CLUSTERS

No. of clusters	# of training data contained in the clusters	Patterns of effects < $c_{e1}^w, h_{e1}^w, c_{e2}^w, h_{e2}^w, c_{e3}^w, h_{e3}^w, c_{e4}^w, h_{e4}^w, c_{e5}^w, h_{e5}^w, c_{e6}^w, h_{e6}^w, c_{e7}^w, h_{e7}^w, c_{e8}^w, h_{e8}^w, c_{e9}^w, h_{e9}^w, c_{e10}^w, h_{e10}^w$ >	Labels
1	25	< -1, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0, 0 >	[ApproachingTeabag]
2	50	< 0, 1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0 >	[GraspingTeabag]
3	15	< 0, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0 >	[DeliveringTeabag]
4	50	< 0, -1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 >	[ReleasingTeabag]
5	45	< 1, 0, 0, 1, 0, 0, -1, 0, 0, -1, 0, 0 >	[ApproachingKettle]
6	5	< 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0 >	[GraspingKettle]
7	40	< -1, 0, 0, -1, 0, 0, 0, 0, 0, 1, 0, 0 >	[DeliveringKettle]
8	50	< 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0 >	[TiltingKettle]
9	40	< 1, 0, 0, 1, 0, 0, 0, 0, 0, -1, 0, 0 >	[PlacingKettle]
10	50	< 0, 0, 0, 0, 0, 0, 0, -1, 1, 0, 0, 0 >	[ReleasingKettle]
11	5(teabag), 45(cup)	< -1, 0, 0, -1, 0, 0, 1, 0, 0, 1, 0, 0 >	[ApproachingTeabag]
12	50	< 0, 0, 0, 0, 1, -1, 0, 0, 0, 0, 0, 0 >	[GraspingCup]
13	20	< 0, 0, 0, 0, 0, 0, -1, 0, 0, -1, 0, 0 >	[DeliveringCup]
14	50	< 0, 0, 0, -1, 1, 0, 0, 0, 0, 0, 0, 0 >	[ReleasingCup]
15	10	< 0, 0, 0, -1, 0, 0, 1, 0, 0, 1, 0, 0 >	[DeliveringTeabag]
16	45	< 0, 0, 0, 0, 0, 0, 0, 1, -1, 0, 0, 0 >	[GraspingKettle]
17	15	< 0, 0, 0, 0, 0, 1, -1, 0, 0, -1, 0, 0 >	[DeliveringCup]
18	5	< -1, 0, 0, 1, 0, 0, -1, 0, 0, -1, 0, 0 >	[ApproachingTeabag]
19	10	< 0, 0, 1, -1, 0, 0, 1, 0, 0, 1, 0, 0 >	[DeliveringTeabag]
20	10	< 0, 0, 1, -1, 0, 0, -1, 0, 0, -1, 0, 0 >	[DeliveringTeabag]
21	5	< -1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0 >	[DeliveringKettle]
22	5	< 1, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0 >	[PlacingKettle]
23	10	< 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0 >	[DeliveringCup]
24	5	< -1, 0, 0, -1, 0, 0, -1, 0, 0, -1, 0, 0 >	[ApproachingTeabag]
25	5(teabag), 5(cup)	< -1, 0, 0, -1, 0, 0, 1, 0, 0, -1, 0, 0 >	[ApproachingTeabag]
26	5	< 0, 0, 1, -1, 0, 0, 1, 0, 0, -1, 0, 0 >	[DeliveringTeabag]
27	5	< 1, 0, 0, 1, 0, 0, -1, 0, 0, 1, 0, 0 >	[ApproachingKettle]
28	5	< -1, 0, 0, -1, 0, 0, 0, 0, 0, -1, 0, 0 >	[DeliveringKettle]
29	5	< 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0 >	[PlacingKettle]
30	5	< 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, -1, 0, 0 >	[DeliveringCup]
31	5	< -1, 0, 0, 0, 0, 0, -1, 0, 0, -1, 0, 0 >	[ApproachingTeabag]

Here, the effects are calculated from the directions of differences by subtracting Z from Y .

work for slightly different situations. This refers to situations that are perceived by the robot as entities being located in initial and goal configurations different from those during learning, although the goal of the task is the same. In particular, the experiments were performed to verify the generation of various sequences of motion primitives in situations with various human disturbances and other exceptional cases. Additional experiments were performed as follows: 1) a human snatched the teabag from the robot arm while it was approaching the cup, 2) a human moved directly toward the cup of tea while the robot was preparing it, and 3) a human inserted a teabag into the cup while the robot was approaching the teabag.

The affordances recommended motion primitives using Bayesian inferences under limited perception when some sensors (particularly the touch and weight sensors) could not be used in the additional experiments. The robot successfully executed the tasks in all the situations by making fully connected transitions between the affordances based



Fig. 8. Motivation graph for executing tea service task. Here, fourteen affordances are used for achieving the task.

on the motivation values generated by combining Bayesian inference. Fig. 10 depicts these situations. These tasks were executed with success rates of about 90% despite slightly different conditions and several human interventions.

IV. DISCUSSION

Even though in the autonomous segmentation process in this study, joint motion trajectories were used to estimate segmentation points, motion trajectories of various types (e.g., end-effector, force, torque etc.) can be used in the process, as validated in our previous work [11]. The segmentation process can be also executed without any constraints on the types of variables or tasks, since segmentation points are estimated in the GMM domain. In addition, the segmentation results were found to be similar (about 90%) to the results of manual segmentation.

In motion primitive learning, the DMPs possess the following advantages: 1) The DMPs can achieve their goals despite changes in goal positions, initial positions, or both. 2) The motion trajectories can be temporally scaled by adjusting a variable. In spite of these advantages, it is difficult to learn an external force when the motion trajectories to be learned are too long. Therefore, DMPs should certainly be learned using the segmented motion trajectories.

Situation-adequate motion primitives were well estimated although the affordances were formulated using naïve BNs. However, the structure of BN may be learned in case of several tasks. Especially, BNs will have to learn their structures to represent tasks that have causal relations between task-relevant entities.

Finally, some tasks may be achieved using situation-adequate motion primitives. However, most tasks would not be achievable, since some oscillation in inferring the motion primitives usually occurs as a result of the actions being selected without consideration of their goal-orientedness. Therefore, goal-orientedness is an important property that a robot uses to infer a suitable action to achieve a given task.

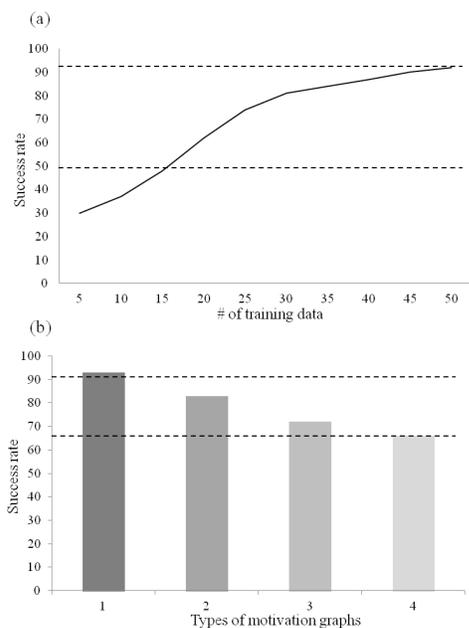


Fig. 9. Quantitative results of the affordances and motivation graphs in the different initial and goal configurations: (a) success rates according to the number of training data used in the affordance [ApproachingTeabag] for the action approaching a teabag in the different configurations of task-relevant entities, and (b) success rates according to the types of motivation graphs. Here, 1, 2, 3, and 4 show the success rates of the different motivation graphs.

However, it is very difficult to design goal-orientedness in cases that need rare or inexperienced sequences. Moreover, it is not easy to achieve the given tasks under uncertain environmental conditions even using well-planned sequences. In this study, using the skill inference process, the robot inferred situation-adequate and goal-oriented motion primitives to achieve a daily-life task under slightly different initial positions of itself and the entities and human interventions, without designing all sequences of skills. Additionally, the robot could compute action probabilities using the Bayesian inference algorithm, even under limited perception, and quickly executed skills using the affordance-based motivation values.

V. CONCLUSION

In this paper, we proposed a unified skill learning and inference framework to handle uncertainties and perturbations. Based on the framework, it is possible for a robot to possess the following three abilities: i) learning motion primitives, ii) learning the meanings of motion primitives, and iii) recombining skills. The framework was designed with autonomous segmentation, skill (i.e., motion primitive and its meaning) learning, and skill inference. To validate the framework, we provided experimental evaluations using a daily-life task performed by a robot arm. In future works, we intend to extract formal rules from BNs to create innovative sequential combinations of skills for achieving novel tasks, in a way similar to how humans create a number of unique sentences using words and grammar rules.

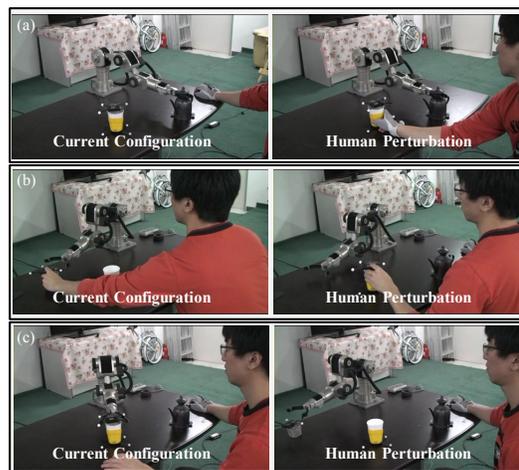


Fig. 10. Illustrations of human perturbations in the additional experiments: (a) a human directly moves to the cup while the robot is pouring water, (b) a human places the teabag into the cup while the robot is approaching the teabag to grasp it, and (c) a human snatches the teabag from the robot while it is placing the teabag into the cup.

REFERENCES

- [1] S. H. Lee and I. H. Suh, "Bayesian Network-based Behavior Control for Skilligent Robots," in Proc. of Intl. Conference on Robotics and Automation (ICRA), 2009, pp. 2910-2916.
- [2] T. Asfour, P. Azad, F. Gyarfas, and R. Dillmann, "Imitation Learning of Dual-arm Manipulation Tasks in Humanoid Robots," International Journal of Humanoid Robotics, vol. 5, no. 2, 2008, pp. 183-202.
- [3] D. Kulic, W. Takano, and Y. Nakamura, "Online Segmentation and Clustering from Continuous Observation of Whole Body Motions," IEEE Transactions on Robotics, vol. 25, no. 5, 2009, pp. 1158-1166.
- [4] M. Mühlig, M. Gienger, and J. J. Steil, "Human-robot Interaction for Learning and Adaptation of Object Movements," in Proc. of IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS), 2010.
- [5] V. Kruger, D. Herzog, S. Baby, A. Ude, and D. Kragic, "Learning Actions from Observations," IEEE Robotics & Automation Magazine, vol. 17, no. 2, 2010, pp. 30-43.
- [6] H. Hoffmann, P. Pastor, D. H. Park, and S. Schaal, "Biologically-inspired Dynamic Systems for Movement Generation: Automatic Real-time Goal Adaptation and Obstacle Avoidance," in Proc. of IEEE Intl. Conf. on Robotics and Automation (ICRA), 2009, pp. 2587-2592.
- [7] S. H. Lee and I. H. Suh, "Skill Learning and Inference Framework," The Sixth Conference on Artificial General Intelligence (AGI 2013), LNAI 7999, pp.196-205, 2013.
- [8] W. Heaven, D. Sykes, J. Magee, and J. Kramer, "A Case Study in Goal-driven Architectural Adaptation," Software Engineering for Self-Adaptive Systems, 2009, pp. 109-127.
- [9] M. Scheutz and V. Andronache, "Architectural Mechanisms for Dynamic Changes of Behavior Selection Strategies in Behavior-based Systems," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 34, no. 6, 2004, pp. 2377-2395.
- [10] J. Jaafar, E. McKenzie, and A. Smail, "A Fuzzy Action Selection Method for Virtual Agent Navigation in Unknown Virtual Environments," in Proc. of Intl. Conf. on Fuzzy Systems, 2007, pp. 1-6.
- [11] Z. Ghahramani and M. I. Jordan, "Supervised Learning from Incomplete Data via an EM approach," in Advances in Neural Information Processing Systems, 1994.
- [12] S. H. Lee, I. H. Suh, S. Calinon, and R. Johansson, "Learning Basic Skills by Autonomous Segmentation of Humanoid Motion Trajectories," in Proc. of IEEE Intl. Conf. on Humanoid Robots, 2012.
- [13] E. Sahin, M. Çakmak, M. R. Doğar, E. Uğur, and G. Üçoluk, "To afford or Not to Afford: A New Formalization of Affordances toward Affordance-based Robot Control," Adaptive Behavior, vol. 15, no. 4, 2007, pp. 447-472.
- [14] S. H. Lee and I. H. Suh, "Motivation-based Dependable Behavior Selection Using Probabilistic Affordance," Advanced Robotics, vol. 26, no.8-9, 2012, 897-921.