Task-Based Robot Grasp Planning Using Probabilistic Inference

Dan Song, Carl Henrik Ek, Kai Huebner, *Member, IEEE*, and Danica Kragic, *Senior Member, IEEE*

***Abstract*—Grasping and manipulating everyday objects in a goal-directed manner is an important ability of a service robot. The robot needs to reason about task requirements and ground these in the sensorimotor information. Grasping and interaction with objects are challenging in real-world scenarios, where sensorimotor uncertainty is prevalent. This paper presents a probabilistic framework for the representation and modeling of robot-grasping tasks. The framework consists of Gaussian mixture models for generic data discretization, and discrete Bayesian networks for encoding the probabilistic relations among various task-relevant variables, including object and action features as well as task constraints. We evaluate the framework using a grasp database generated in a simulated environment including a human and two robot hand models. The generative modeling approach allows the prediction of grasping tasks given uncertain sensory data, as well as object and grasp selection in a task-oriented manner. Furthermore, the graphical model framework provides insights into dependencies between variables and features relevant for object grasping.**

***Index Terms*—Cognitive human–robot interaction, grasping, learning and adaptive systems, probabilistic graphical models, recognition.**

I. Introduction

The TRANSFER of information between a teacher (human/robot) and a student (robot) requires a common knowledge representation. When the human and the robot student have identical motor and sensory capabilities, the task may be simply to transform the action of one to the other by changing the frame of reference. However, such transfer is not commonly possible given that the embodiments and associated capabilities often differ. In such cases, direct copy of an action parameterized specific to a human teacher's body (such as wrist orientation and finger joint angles) will not guarantee the success of the task intended, and sometimes is even not feasible due to distinct kinematic structure of the manipulators. This is especially difficult in the scenarios of object grasping and manipulation because human and robot often have very different structures and sensorimotor capabilities of the hands. This is commonly referred as the "correspondence problem" in the imitation learning literature [1]. The way to address it is to focus on achieving the aimed goal of a task with any means that the learner is capable of, while getting around direct imitation at the "action level" [2], which aims to copy detailed action parameters, such as joint angles and end-effector (hand) configurations and its trajectories. This is an approach commonly referred to as "goal-directed imitation." Goal-directed imitation is an advanced learning strategy inbuilt in many biological systems, such as human and primates [3], and has been widely adopted in teaching robots complex motor tasks [4]–[8].

In this paper, we study how to teach robots to do complex tasks involving object grasping and manipulation through *goal-directed grasp imitation*. We represent the object-grasping action and its outcomes (the tasks it can afford) at a level that facilitates a more natural, higher level, and goal-directed grasp transfer. When less information is transferred, i.e., only the goal of the grasp action, the student is less constrained by the teacher in how to perform the task. However, this means that the student needs to recognize the goal of the teacher's grasp action, and the student needs to know how to achieve the goal through planning grasps with its own embodiment. The former phase requires the robot to build an internal model encoding the links of observed human actions and their intended task, i.e., *the action understanding of the human teacher*. The later phase requires the robot to model its own sensorimotor capabilities and the relation to the tasks (or goals) it wants to achieve, i.e., *the action representation of the robot learner*.

We take a data-driven approach to address both phases of operation mentioned above; therefore, to solve the problems of grasp knowledge transfer between a teacher and a learner. We first acquire a large set of task-labeled grasps performed by a human teacher and a robot learner with different embodiments (hands). For each hand, we use probabilistic graphical models—Bayesian networks (BNs) [9]—to encode variations of all the variables (including task, and other object and grasp action variables). The learned BNs encode task-oriented object-grasping capabilities specific to each hand embodiment. The grasp transfer between the teacher and the learner can be done in a goal-directed manner by emulating the intended task of the human demonstrator. The initial development of this model was presented in [10]–[14].

A. Problem Formulation

Our model implements goal-inference and goal-based grasp planning suitable for an artificial agent with a specific embodiment. The goal inference relates to estimating the intention of the human teacher by observing an execution of a grasping task. The grasp planning involves decisions over object selection, and grasp type so that the requirements posed by the task are fulfilled.

Manuscript received December 14, 2013; revised July 20, 2014 and January 5, 2015; accepted March 2, 2015. Date of publication April 2, 2015; date of current version June 3, 2015. This paper was recommended for publication by Associate Editor J. Peters and Editor D. Fox upon evaluation of the reviewers' comments. The material in this paper was presented at the IROS 2010, ICRA 2011, and IROS 2011, and ICRA 2013. This work was supported by EU IST-FP7-IP GRASP, and Swedish Foundation for Strategic Research.

The authors are with the KTH—Royal Institute of Technology, 11428 Stockholm, Sweden (e-mail: dsong@kth.se; chek@kth.se; khubner@kth.se; danik@kth.se).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2015.2409912

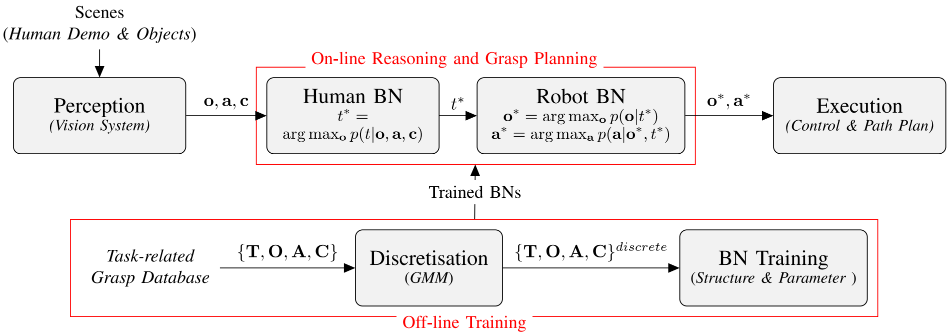
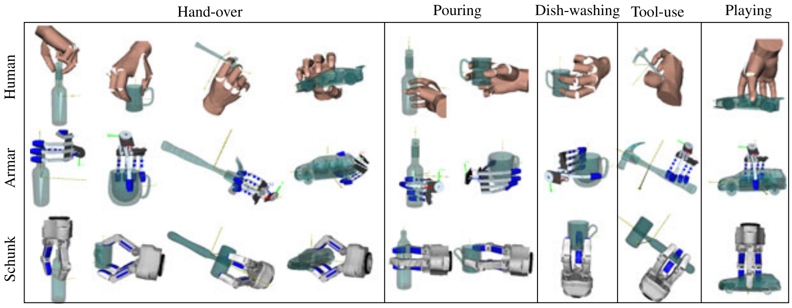

Scenes
(*Human Demo & Objects*)
Perception
(*Vision System*)
$\mathbf{o}, \mathbf{a}, \mathbf{c}$
On-line Reasoning and Grasp Planning
Human BN
$t^* = \arg\max_{\mathbf{o}} p(t|\mathbf{o}, \mathbf{a}, \mathbf{c})$
t^*
Robot BN
$\mathbf{o}^* = \arg\max_{\mathbf{o}} p(\mathbf{o}|t^*)$
$\mathbf{a}^* = \arg\max_{\mathbf{a}} p(\mathbf{a}|\mathbf{o}^*, t^*)$
$\mathbf{o}^*, \mathbf{a}^*$
Execution
(*Control & Path Plan*)
Trained BNs
Task-related Grasp Database
$\{\mathbf{T}, \mathbf{O}, \mathbf{A}, \mathbf{C}\}$
Discretisation
(*GMM*)
$\{\mathbf{T}, \mathbf{O}, \mathbf{A}, \mathbf{C}\}^{discrete}$
BN Training
(*Structure & Parameter*)
Off-line Training

Fig. 1. System diagram. The top row shows an online process of goal-directed grasp imitation consisting of a vision-based perception subsystem, a BN-based reasoning subsystem, and a grasp execution subsystem. The bottom row shows the offline training of the BNs. The scope of this paper is within the red rectangles.



Task Requirements on Objects and Grasps

Tasks	The object	The grasp should
Hand-over	can be of any kind (all 4 object classes)	leave enough uncovered surface or handle for safe regrasp
Pouring	should be pourable (bottle, mug)	not cover opening part for pouring out the liquid
Tool-use	should be a tool (hammer)	be on the handle for applying the function of the tools
Dish-washing	is washable in dishwasher (mug)	allow placing the object upside down into the dishwasher
Playing	should be a toy (sedan toy car)	allow playing with the toy

Fig. 2. Left: Task-constrained grasp examples. For each task and each object class that affords the task, one preferred grasp configuration is shown for all three hands. Right: Constraints considered when generating task-related grasp database.

1) Notations: Before going into the formulation, we first clarify the use of notations. Commonly, we use the upper case letters, such as T, O, A, C to represent variables or a group of variables, and the corresponding lower case letters to represent their instantiations, such as $t, \mathbf{o}, \mathbf{a}, \mathbf{c}$. The bold face of the lower case letter indicates that the represented variable is a vector instead of a scalar value. A set of data with N instantiations will then be represented by the bold upper case letters $\mathbf{T}, \mathbf{O}, \mathbf{A}, \mathbf{C}$, where, for example, $\mathbf{C} = (\mathbf{c}_i, i = 1, \ldots, N)$.

In this paper, we use O to represent a group of variables describing object category, size, and shape-related features, A a group of variables that parameterize a grasp action such, as hand position, orientation, and articulation at the moment of grasping, and C a group of constraint variables that parameterize the object-grasp configuration, such as the free volume on the grasped object that is not covered by the hand. We use notation $\mathcal{OG}$ to represent an object-grasp configuration illustrated in Fig. 2.

2) Probabilistic Modeling: In Fig. 2, a set of example $\mathcal{OG}$s are listed for different hands according to the task they afford. Each $\mathcal{OG}$ can be described by the set of aforementioned O, A, C features and the task(s) T that the grasp can afford. Our model aims to encode the interdependencies between the O, A, C, and T for different hands.

Goal inference can be expressed as finding the mapping $f_1 : \{\mathbf{o}, \mathbf{a}, \mathbf{c}\} \rightarrow t$, object selection $f_2 : t \rightarrow \mathbf{o}$, and grasp planning $f_3 : \{t, \mathbf{o}\} \rightarrow \mathbf{a}$. In terms of modeling the mappings, the simplest approach would be to assume f_1, f_2, and f_3 to take functional form for tasks in the task set. This would imply that $\{\mathbf{o}, \mathbf{a}, \mathbf{c}\}$ deterministically assesses the task t, which in turn assesses the object $\mathbf{o}$ according to f_1 and f_3. However, this scenario is not a particularly realistic one since different tasks require different settings on different subsets of O, A, C features. Furthermore, for a given task, it is often not a single, but rather a set of grasps that can be considered as good. For example, many side grasps around a bottle are good for pouring. In other words, we aim to find a distribution of good grasps rather than a single deterministic mapping. This is also beneficial during real-world grasp execution, where one single grasp may not be feasible due to various factors, such as occlusion or internal collisions.

Thus, we adopt a probabilistic modeling approach capable of handling a multimodal problem. If we model the joint distribution of all variables $p(\mathbf{o}, \mathbf{a}, \mathbf{c}, t)$, we can infer the conditional probabilistic distributions $p(t|\mathbf{o}, \mathbf{a}, \mathbf{c})$ for goal inference, and $p(\mathbf{o}|t)$ and $p(\mathbf{a}|\mathbf{o}, t)$ for goal-directed object selection and grasp planning, respectively.

In this study, we use a directed graphical model, BN, to model $p(\mathbf{o}, \mathbf{a}, \mathbf{c}, t)$. Fig. 1 illustrates the entire system. In the offline learning phase, we obtain hand-specific BNs for the human teacher and robot learner. During the process of goal-directed grasp imitation, the robot first uses its perception system to extract $\mathbf{o}, \mathbf{a}, \mathbf{c}$ from visual observation of the human demonstration. The goal inference is then the process that determines the most likely task using human hand-specific BN (or BN^H) $t^* = \arg\max_{\mathbf{t}} p(t|\mathbf{o}, \mathbf{a}, \mathbf{c}, \mathrm{BN}^H)$. Grasp planning is a process that determines the most appropriate object $\mathbf{o}^* = \arg\max_{\mathbf{o}} p(\mathbf{o}|t^*, \mathrm{BN}^R)$, and then the most appropriate grasp $\mathbf{a}^* = \arg\max_{\mathbf{a}} p(\mathbf{a}|\mathbf{o}^*, t^*, \mathrm{BN}^R)$ for executing the given task, using robot hand-specific BN (or BN^R). Finally, a grasp execution system performs the planned grasp on a real robot platform. Note that the scope of this paper is the grasp reasoning and planning systems.

B. Contributions

The main contributions of this study are as follows: 1) a human-assisted method for acquiring manually annotated task-related grasps, 2) a methodology for learning probabilistic relationships between various task-, object-, and action-related features, 3) an embodiment-specific concept of affordance, which maps symbolic representations of task requirements to the continuous object and grasp action parameters. Furthermore, using a probabilistic framework, we can easily extend the object and action spaces, and allow 4) flexible learning of novel tasks and adaptation in uncertain environments. Finally, our model can be applied to a goal-directed imitation framework, which allows 5) a robot to learn from humans despite differences in their embodiments. Our initial findings have been reported in [10]–[13]. The application of the modeling framework to predict human intention in visual observations has been reported in [14]. Except for summarizing the previous work, this paper includes the following extensions: 1) a more challenging dataset including more tasks and hands; 2) the evaluation of two discretization methods for learning data representation; and 3) a more detailed evaluation of the proposed modeling framework. This paper dedicates a large space for designing and evaluating discretization methods because it is the key to learn the structure of interaction between a large set of different sensory modalities. A good structure is the core of a good model for the reasoning system.

II. Related Work

The work presented in this paper relates to a wide range of works in robotics and machine learning. To limit the overview, we identify three separate topics that are central to our work: imitation learning, affordances, and probabilistic modeling and their applications in robot grasping.

A. Goal-Directed Imitation Learning

Imitation learning is an effective approach for teaching robots simple tasks [15]. Most applications of imitation learning are focused on learning the entire sequence of complex tasks (behavior learning) [16], learning full-body locomotion [17], or motions of the arm [18] and the end effector [19]. When applied to grasping problems, one important focus is on learning the trajectories of the end effector, i.e., how to reach to the grasping position (e.g., [20]). Direct mapping of grasping configuration from teacher's to learner's hands is difficult. The challenge is imposed by both the kinematic differences of the actor that is the hands, the "correspondence problem" [21], and the physical interaction with the objects. Several approaches include mapping in a lower dimensional or "synergy" space, such as "virtual fingers" [22], mapping through classification of grasps, "grasp taxonomy" [23], through empirical encoding "master motor maps" [24], or through a database of demonstrated grasps on a set of primitive object shapes [25]. Work in [25] addressed grasp mapping through picking the demonstrated grasps applied on the object with similar primitive shapes (boxes, cylinders, or spheres). Schmidts *et al.* [26] incorporated both motion and force data to learn grasping skills that showed better generalization compared with encoding kinematic relations only. These works all focused on teaching robots to stably grasp the objects, which cannot guarantee the success of the grasping tasks. This is because the success of a grasp relies on both good physical interaction with the object (stable grasps), and on satisfying the high-level goal of the subsequent manipulation actions (the goal of the grasping task).

To solve the correspondence problem in imitation learning, a number of works have proposed solutions based on different levels of imitation from direct action copy to higher level imitation [1], [2]. The highest level is the effect level, where the learner emulates the intention or goal of the teacher using the action that conforms to its own embodiment. Goal-directed imitation has been explored by a number of robotic researchers [6]–[8]. This is particularly useful for learning difficult motor skills that require intensive cognitive guidance's, such as grasping tasks. Works in [10] and [12] applied such a concept to achieve human-to-robot knowledge transfer for the purpose of goal-directed grasping.

Goal-directed imitation is also supported by the learning paradigm based on internal models [27]–[30]. Wolpert and Kawato [27] proposed a modular approach based on multiple pairs of inverse and forward internal models to explain motor learning and control in biological systems. In [28], it is shown that the internal models that represent the brain circuitry subserving sensorimotor control also participate in action recognition. They are used to predict the goal of observed behavior, and activate the correct actions to maintain or achieve the "goal" state. Inspired by this, the authors proposed a robotic architecture for imitation and learning. The work in [29] and [30] instantiated such internal models through BNs, a probabilistic modeling approach that links motor activities and their consequences. A later work in [31] extended the use of an internal model to the domain of visual-manual tasks. Song *et al.* [14] are the first to show that the same BN model used to implement goal-directed grasp imitation can also be used for recognizing visual observed human demonstration on a set of house-hold objects.

B. Affordances

Gibson proposed the concept of affordances that defines action possibilities of an individual agent in the environment, which depends on its action capabilities [32]. Affordances are, therefore, the fundamental concept for agents (artificial or human) to act in a complex environment, to interact with each other and with objects in a flexible manner. Montesano *et al.* [8] modeled this affordance on a robot in simple object manipulation tasks. The authors adopt a self-supervised developmental approach, where the robot first explores its sensory motor capabilities, and then interacts with objects to learn their affordances. The affordances being modeled are measured as the salient changes in the agent's sensory channels, which are interpreted as effects of specific actions applied on objects. As an example, an effect of poking a ball is making it roll. Later, the work in [33] associates the learned affordance model to learning of words, automatically linking the meaning of the words to the properties of the objects and the manipulative actions on them.

Although learning through exploration is an important step for a robot to discover its own motor ability, another necessary step to achieve goal-directed behavior is to link this immediate motor act and its effects (as to poke the ball and let it roll) to the conceptual goal of an assigned task (as to provide the ball to a child). While trial-and-error-based exploration can be seen as inefficient to solve such goal learning problems, human supervision is helpful. Our work reported in [10] and [12] takes this step to model the affordances at a higher level concept, the grasp task, that is to be provided by human tutors. As an example, how an agent grasps an object determines whether this object-hand configuration can afford the subsequent actions, such as to pour out water or to handover to another agent. The affordance is to be recognized, and not perceived directly as in [8] and [33].

C. Probabilistic Modeling of Affordances

Observations from sensory streams in grasping are often uncertain and corrupted by the significant noise. The data are often composed of several different views or modalities each being high dimensional on its own making the concatenated observation space extremely high dimensional. Furthermore, in terms of estimation, there is often more than one correct answer, an example being that there are many different possibilities of how to grasp a cup in order to pour from it. Each of these characteristics on its own makes a problem more difficult; therefore, putting them all together provides an even more challenging scenario. In this paper, we will take a probabilistic approach to model task-based grasping.

Probabilistic models naturally deal with uncertainty and have been successfully applied across many different robotic domains, such as control [34], [35], motion transfer [36], and affordances [8], [33]. We are interested in inferring grasps from several different combinations of observations, and we expect the estimation task to be highly multimodal, i.e., that there are many possible grasps that afford a specific task. A probabilistic approach from a generative perspective parameterizes the variations in the data. This makes it ideally suited for the task as we can perform inference based on partial observations. In this missing data problem, we do not rely on assumptions about the distribution of the inference task; therefore, being able to encode ambiguities. The high dimensionality of the data we wish to model requires infeasible amounts of data to circumvent the curse of dimensionality. However, the concept of affordances is underpinned by a hierarchical structure, as specific knowledge provides information about other parts of the state space. Such structures lie at the heart of probabilistic models described by conditional distributions. More formally, when a set of data can be described using conditional distributions, this can be exploited to learn compact and efficient models by investigating conditional independence. To that end, we will use a directed factorized probabilistic model known as BN [9]. This model is used to encode the statistical dependencies between object attributes, grasp actions, and a set of task constraints; therefore, linking the symbolic tasks to quantified constraints. The factorization described by the structure of the observations will not only reduce the requirements on the training data and speed up learning, but significantly simplify inference as well.

In our previous work [10], we applied this approach to grasping using a small set of sensory streams with good results. However, this model required that the hierarchy (the network structure) needs to be specified *a priori*. This is difficult because it requires complete knowledge of conditional relationships in the data. We improved the model by first learning a compact data representation, which then allows the hierarchical structure to be learned from the data [11], [13].

III. Data Representation and Generation

Our approach is data driven. In order to achieve a useful and expressive model, the data used to create the model need to reflect the variations we expect to encounter. In this section, we will describe and motivate the representation and collection process of the data that form the basis for the work presented in this paper.

As briefly mentioned in Section I, the features describing each $\mathcal{OG}$ (as exemplified in Fig. 2) are divided into three subsets: *object features* (O) from the object representation, *action features* (A) from the planned grasps, and *constraint features* (C) resulting from the complementation of both, i.e., the hand-object configuration $\mathcal{OG}$. Each $\mathcal{OG}$ is visualized in a simulation environment to a human tutor who associates it with a binary task label ($T \in \Re^L$), indicating which tasks in the task set this grasp affords.

A. Feature Representation

1) Task: In our notation, a *task t* refers to a "basic task" that involves grasping or manipulation of a single object. According to [37], such a basic task can be called as a *manipulation segment*, which starts and ends with both hands free and the object at the stationary state. These manipulation segments are the building blocks for complex manipulation tasks. Although there may be an infinite number of complex tasks, we assume the basic building blocks form a finite set of object manipulation tasks. We therefore choose our task representation at the level of manipulation segments as each of them has an independent goal directly constraining how to grasp an object.

2) Object Features: An object feature set $O = \{O_1, \ldots, O_{n_O}\}$ specifies the attributes (e.g., size) and/or categorical (e.g., class) information of an object. The features in O are not necessarily independent. The same attribute, such as shape, can be represented by different variables dependent on the capabilities of the perceptual system and the current object knowledge. For instance, eccentricity and convexity can be estimated from any kind of point cloud or mesh, while 3-D shape representations like Zernike descriptors [38] can be used when a complete and dense 3-D model of an object is available, i.e., when the object is known. Although apparently redundant, a system-dependent object representation offers flexibility in generalization across possibly different vision systems, which can provide various levels of object knowledge.

Note that the O features are object intrinsic properties. We do not consider state changes of the object, e.g., its pose. Such a representation is beneficial because it allows us to build a complete model over what kind of grasps afford a given task, without being distracted by issues such as reachability. This encourages a modular design of the robot manipulation systems, where our system addresses a goal-directed grasp planning and problems, such as inverse kinematics and collision detection should be addressed by a path planning system.

3) Action Features: An action feature set $A = \{A_1, \ldots, A_{n_A}\}$ describes the *object-centered kinematic* grasp features, which may be the direct outputs of a grasp planner. A may include properties like grasp position, orientation, hand approach vector, or the grasp-finger configuration.

The action set is a static feature set. We do not look at time sequences of the data that record how hand is approaching the object, and how fingers close around the object. Instead, only data for a single time frame are used. This frame is chosen manually, for example, when finger close around the object and establish a stable grasp.

Such a representation is chosen for the following reasons. First, this is consistent with the output of most grasp planner systems. Therefore, we can use our system to constrain the selection of one grasp from a set of planned grasps. Second, such a “goal state” of the “reaching and grasping” process is already very informative to determine what one can do (i.e., the task) after lifting the object. The temporal data beforehand are relatively redundant. Finally, a static data are much more compact; thus, more efficient to model than temporal data, and structural learning algorithms in BN can be applied in a straightforward manner.

4) Constraint Features: Finally, constraint feature set $C = \{C_1, \ldots, C_{n_C}\}$ specifies a set of *constraint variables,* which are defined by human experts. They can be considered to be an abstraction of the combined O, A features, and have direct links to the possible tasks under consideration. For example, one may define the enclosure of the object volume as a constraint feature, which defines the constraints of a handover task. Such a feature obviously depends on both O (size and shape) and A (grasp position and finger configuration) features. However, it is only a 1-D variable; thus, provides a much more compact representation than using a set of O, A features. They can be used to quantitatively interpret the “goal” or the “requirements” of a given task.

B. Feature Generation

Fig. 3 shows the schematic of the data generation process. For each hand-object pair, we generate a set of stable grasps using a grasp planner system—*Box Approximation, Decomposition, and Grasping* (BADGr) [39]. BADGr is used to extract the set of $\{O, A, C\}$ features for each $\mathcal{OG}$ configuration. These features are described in Fig. 9.

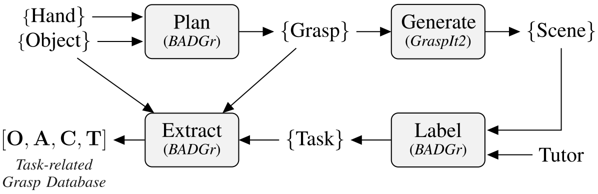


Fig. 3. Schematic diagram for generating a task-related grasp database.

To provide task labels for this set of stable grasps, each $\mathcal{OG}$ is visualized as a 3-D scene in GraspIt!. The human tutor then selects among a set of possible tasks the affordable task(s) for this $\mathcal{OG}$. If this grasp instance is labeled to be valid for at least one task, a data instance will be included in the final complete dataset $[\mathbf{O}, \mathbf{A}, \mathbf{C}, \mathbf{T}]$ for training and testing.

IV. Methodology

As we stated in problem formulation, we use BNs [9] to model the joint distribution $p(O, A, C, T)$ for the purpose of goal-inference and goal-directed grasp planning. The former is essentially a task classification problem, i.e., to classify T by inferring $p(T|X)$, where $X \subseteq \{O, A, C\}$. BN is a generative modeling framework, which allows us to learn a single model from which we are capable to perform inference given any partial view of the data. However, generative models have the disadvantage compared with discriminative models that it is not obvious how to perform feature selection. This means that the model has to represent all the variations in the data, even those that are not relevant for the estimation tasks that we are interested in. In order to evaluate the classification performance, we will compare the generative model with a discriminative approach, specifically, we will use kernel logistic regression (KLR). In this section, we provide an overview of the two modeling approaches.

A. Kernel Logistic Regression

Given a class variable T and a input feature set X, KLR models the probability of the class variable $p(T|X)$ through a weighted sum of the similarities (kernels $\mathcal{K}$) between a testing point $\mathbf{x}$ and each training point $\mathbf{x}_i$ [40]

$$p(\mathbf{t}|\mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp\left\{ - \sum_{i=1}^{N} w_i \mathcal{K}(\mathbf{x}, \mathbf{x}_i) \right\}} \qquad (1)$$

where N is the number of training data points.

In this paper, we choose $\mathcal{K}$ to be a Gaussian kernel. Training a KLR model is to find the weight vector $\mathbf{w}$ that maximizes the regularized probability of the data

$$-\sum_{i=1}^{n} \log p(t_i|\mathbf{x}_i; w_i) + \eta\, \mathrm{trace}(\mathbf{w}\mathbf{K}\mathbf{w}^T) \qquad (2)$$

where t_i is one training point of class variable T, $\mathbf{K}$ is the kernel Gram matrix, with $K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, and η is the regularization constant. During training, the kernel bandwidth parameters and η are chosen by cross validation.

B. Bayesian Networks

A BN [9] is a directed graphical model that encodes the joint distribution of a set of random variables $V = \{V_1, V_2, \ldots, V_m\}$ in a network structure. Each node in the network represents one variable, and the directed arcs represent conditional (in)dependencies. Given a structure of the network S and a set of local conditional probability distributions (CPDs) of each variable V_i, the joint distribution of all the variables can be decomposed as

$$p(\mathbf{v}) = p(\mathbf{v}|\boldsymbol{\theta}, S) = \prod_{i=1}^{m} p(\mathbf{v}_i|\mathbf{pa}_i, \boldsymbol{\theta}_i, S) \tag{3}$$

where $\mathbf{pa}_i$ denotes the parents of node V_i, and the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$ specifies the CPDs. Learning a BN includes discovering from a dataset $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$: 1) how one variable depends on others ($\boldsymbol{\theta}$), and 2) what the conditional in-dependencies between different variables are (S). The former is an instance of parameter learning and the latter of structure learning. Various algorithms and techniques have been developed to learn a BN in different model and data conditions.

BN models the joint distribution of task and a set of task-relevant variables, i.e., $V = \{T, X\}$, where $X \subseteq \{O, A, C\}$. To correctly describe a manipulation task, both high-level conceptual information and continuous low-level sensorimotor variables are needed. The variables in this study are both discrete (e.g., $task$) and continuous (most O, A, C features). The continuous features, such as hand-grasp configuration can be high dimensional with complex probabilistic distributions.

Learning BN structures from both continuous and discrete data are difficult, particularly when continuous data are high dimensional and sampled from complex distributions. Most algorithms for structure learning only work with discrete variables. Therefore, a common approach is to convert the mixed modeling scenario into a completely discrete one by discretizing the continuous variables [41]. In Section V, we will introduce two different discretization approaches based on self-organizing maps (SOM) [42] and Gaussian process latent variable models (GP-LVM) [43] for this purpose. Given the discretized data, we use a greedy search algorithm to find the network structure (the directed acyclic graph, or DAG) in a neighborhood of graphs that maximizes the network score (Bayesian information criterion [44]). The search is local and in the space of DAGs; therefore, the effectiveness of the algorithm relies on the initial DAG. As suggested by [45], we use another simpler algorithm, the maximum weight spanning tree [46], to find an oriented tree structure as the initial DAG. Once the structure is determined, the conditional probability tables are updated sequentially using a standard Bayesian parameter updating scheme.

V. Data Discretization

We propose two discretization algorithms for our high dimensional continuous problem based on SOM [42] and GP-LVM [43] (see Fig. 4). Both approaches result in a Gaussian mixture model (GMM) representing the density of the input space. This probability representation can "smooth out" the boundaries in the initial clusters; hence, provide a "soft" discretization similar to [47]. We will explain them in detail in the following section.

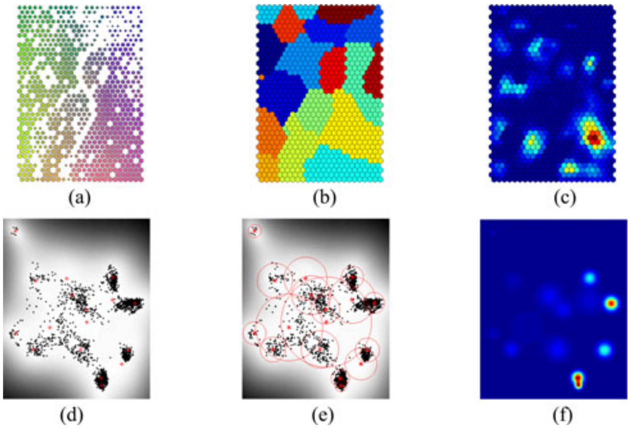


Fig. 4. Two discretization methods illustrated on 20-D data of human hand final grasp configuration $fcon$. Top row is SOM-based method, and bottom row is the GP-LVM-based method. (a) SOM with the size of the unit determined by number of original training data that is mapped to that unit. (b) Clustering results on the SOM. (c) Density map from the trained GMM. The GP-LVM plots show the location of the latent points and the grayscale value encodes the posterior probability in the observed space associated with the latent location.

A. Discretization Using Self-Organizing Map

In this approach, we first use SOM to efficiently cluster the high-dimensional data Y to automatically define the boundaries of discrete intervals. From the clustering results, we then obtain a GMM to represent the density of the input space. For a set of N observed high-dimensional values $Y \in \Re^{N\times D}$, we first use a SOM-based clustering approach as in [48] to form K clusters. The principle is to use SOM to project Y to a set of prototypes (map units) that are then combined to form final clusters.

SOM [42] is often used in vector quantization and visualization of high-dimensional data and consists of a regular usually 2-D grid of M map units. Each unit i is represented by a prototype vector $\mathbf{w}_i$ that has the same dimensionality as the original data. The units are connected by neighborhood relations discovered from original training data. The SOM can thus be interpreted as a topology preserving mapping from input space Y onto the 2-D map units $\mathbf{W} \in \Re^{M\times D}$ resembling the density of the original data. Given this property, the problem of clustering on the original N data points can be reformulated into clustering on their mapped M prototype vectors on the SOM. Since M is much smaller than N, clustering is more efficient in computation.

Fig. 4 top row illustrates this process. First, a large set of prototypes—much larger than the expected number of clusters K—is formed using the SOM [see Fig. 4(a)]. Given a predefined number of clusters K, clustering is performed on $\mathbf{W}$ using the K-means algorithm. To determine K, we use the Davies–Bouldin validity index [49], defined as

$$\mathcal{I}_{db} \triangleq \frac{1}{2}\frac{1}{K}\sum_{k=1}^{K}\max_{l\neq k}\left\{\frac{s_k + s_l}{d_{kl}}\right\} \tag{4}$$

where s_k is intracluster distance, and d_{kl} the intercluster distance, K is the minimizer of $\mathcal{I}_{db}$. Fig. 4(b) shows the clustering result on the set of prototypes.

Each data point of the original dataset Y is assigned to the cluster of its prototype $\mathbf{w}_i$. This result provides the initial clustering of the data for learning of a GMM

$$p(\mathbf{y}) \propto \sum_{k=1}^{K} \lambda_k \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}_k, \Sigma_k) \tag{5}$$

where $\boldsymbol{\mu}_k \in \Re^D$ and $\Sigma_k \in \Re^{D \times D}$ are the mean and covariance of each Gaussian component, and λ_k are the mixing weights. The parameters of the mixture model are learned using a standard EM approach. The learned model encodes the density distribution in the original data space [see Fig. 4(c)].

B. Latent Space Discretization

The latent space discretization is also a two-stage process. First, we learn a new low-dimensional representation of the data over which its density can be easily modeled. Second, a mixture model is used to model the density of the data in the new representation. The latent representation is learned using a sparse approximation of the GP-LVM [43] with the incorporation of a novel prior model that encourages the data to be clustered.

1) Sparse Gaussian Process Latent Variable Model: In the GP-LVM framework, the observed data are assumed to have been generated from a low-dimensional latent variable through a functional mapping with the added Gaussian noise $y_d = f_d(X) + \epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, where $_d$ indicates the dimension. In order to find a solution to this general and ill-constrained formulation, one needs to include additional priors to regularize the solution space. The novelty of the GP-LVM is that it uses a flexible Gaussian process (GP) [50] priors to model this mapping. Formulating the likelihood of the data and marginalizing out the mapping leads to the marginal likelihood of the model

$$p(\mathbf{Y}|\mathbf{X}, \Phi) = \prod_{d=1}^{D} \int p(\mathbf{y}_d|\mathbf{f}_d) p(\mathbf{f}_d|\mathbf{X}, \Phi) df \tag{6}$$

where Φ are the hyperparameters of the model, which characterizes the prior. A solution to the latent representation can be found by maximizing the marginal likelihood for the observed data. However, performing this maximization is an expensive optimization procedure with $\mathcal{O}(N^3)$ complexity, which significantly reduces the applicability of the model. Therefore, an augumented model was formulated in [51], where the latent space is approximated using a set of sparse latent *inducing* variables, which significantly reduces the complexity. In [52], a variational framework was developed to efficiently learn the latent location X, inducing inputs X_u, as well as the hyperparameters of this sparse model. Fig. 4(d) shows the 2-D latent space learned from the hand-grasp configuration data, where black dots are the latent locations of training data X, red stars depict the inducing inputs X_u.

The intuition behind the inducing variables $\mathbf{X_u}$ is that the two latent function values $\mathbf{f}$ and $\mathbf{u}$ can only communicate through $\mathbf{X_u}$; therefore, *induce* the dependencies between the outputs [51]. In other words, $\mathbf{X_u}$ compose a sparse set of "supporting points" that effectively explain the observed data in a more compact manner. This means that the inducing input $\mathbf{X_u}$ can be interpreted as a sparse representation of the full latent representation $\mathbf{X}$, which is a low-dimensional parameterization of the observed data $\mathbf{Y}$.

2) Discretization Using Sparse Gaussian Process Latent Variable Model: Dimensionality reduction is a severely ill-constrained task. The GP-LVM, as described above searches for a solution by using a $\mathcal{GP}$ to encode a preference over the generative mapping f. However, it is important to note that this is just one of infinitely many possible solutions. In this paper, we are interested in a representation that represents the data in a discretized manner with good accuracy. Using the framework above, we could learn such a representation by encoding this preference as prior distribution over $\mathbf{X}$. To encode this, we would need to know which cluster each data point should be associated with, i.e., "a chicken and an egg problem." However, rather than encoding this directly, we can use the sparse model and place a prior on the inducing points, which encourages each $\mathbf{x_u}$ to be independent. This means that each latent point $\mathbf{x}$ is generated by a single inducing point $\mathbf{x_u}$.

A prior that will encourage such behavior can be created by penalizing the L_1 norm of the off-diagonal elements of the inner product matrix computed between the inducing points

$$p(X_u|\theta_u, \beta_u) = \mathcal{N}(\sqrt{D(X_u, \theta_u)}|0, \beta_u^{-1}) \tag{7}$$

$$D(X_u, \theta_u) = \sum_{ij}^{M} (1 - \delta_{ij}) k_u(\mathbf{x}_{ui}, \mathbf{x}_{uj}, \theta_u)$$

where δ_{ij} is the Dirac delta function. If the function $k_u(\mathbf{x}_{ui}, \mathbf{x}_{uj}, \theta_u)$ is smooth and monotonically decreasing with respect to $||\mathbf{x}_{ui} - \mathbf{x}_{uj}||$, the distribution will encourage a representation with well-separated clusters. The parameters β_u and θ_u control the strength of the prior and the smoothness of the kernel, respectively. In specific, we will use a radial-basis function, where θ_u controls the width of the function that relates to the strength of the prior.

In the second step, one could directly use the inducing points as cluster centers to discretize the data. However, in order to reduce the number of hyperparameters in the model, the parameters of the covariance function are shared among the inducing points. This means that in order to increase the representational power of the model, we fix the location of the inducing points and learn an independent covariance for each of the centers. In effect, this implies learning a GMM with the inducing points as means in the representation provided through the GP-LVM

$$p(\mathbf{x}) \propto \sum_{k=1}^{K} \lambda_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k^{-1}) \tag{8}$$

where λ_k are the mixing weights. Note that in the latent space discretization, the cluster centers $\boldsymbol{\mu}_k \in \Re^q$ are fixed at the inducing points, i.e., $\boldsymbol{\mu}_k = \mathbf{x}_{uk}$. We choose a spherical covariance function for each Gaussian component because this is consistent with the RBF kernel used in the inducing prior. Covariance matrix for each component is thus $\Sigma_k = I\sigma_k$, where $I \in \Re^{q\times q}$ is the identity matrix, and σ_k is the variance in each dimension of the latent space. The parameters λ_k and σ_k are optimized through the standard expectation maximization approach. Fig. 4(e) shows the learned GMM model with the red circles representing one standard deviation of the Gaussian components. Fig. 4(f) shows the density distribution of the latent space.

a) Inference: A trained BN defines a factorization of the joint distribution $p(T, O, A, C)$ that adheres to the discovered conditional dependencies. The availability of such a BN enables the computation of the marginal probability density of a group of variables by conditioning on the rest. This is commonly performed by applying the junction tree algorithm [53]. The result of the latter is a set of probabilities, for all the discrete states of a variable Z, conditioned on the observed evidence e, $p_k = p(Z = k|e)$. If the queried variable Z is originally a discrete variable, such as $task$, p_ks provide a complete probability information that allows us to classify tasks given observed object-grasp configurations. If, however, Z is a continuous variable, a further step is required to recover the expected continuous value $E(Z)$ in its observation space.

Estimating a continuous representation from soft evidence is an “inverse problem” in soft discretization as it recovers the continuous value from discrete states. This is also known as defuzzification in fuzzy set theory, which does not have a simple solution [54]. The simplest and most common approach for interpreting the output of a discrete BN is to use the *most likely state* of the inferred variable as output, and then take the mean of that state represented in the one Gaussian component in the GMM model, i.e., $E(Z) = \mathbf{u}_k$, where $k = \arg\max_k(p_k)$. In this approach, however, there is a danger of tremendous information loss, especially when p_k does not have a strong preferred single state, but it spreads out to multiple states. In such cases, the most likely solution disregards all the other states that also have high likelihood. An alternative method would be to estimate the expected value using a weighted sum of the component centers that takes probabilities of all the states into account. The expected value of variable Z is then defined as

$$E(Z) = \sum_{k=1}^{K} p(Z = k|e)\mathbf{u}_k = \sum_{k=1}^{K} p_k \mathbf{u}_k. \tag{9}$$

It is important to note that the recovered continuous representation $E(Z)$ using (9) is in the space where GMM discretization model is build on. Therefore, if Z is a high-dimensional continuous variable, such as the hand-grasp configuration and its data are discretized with the latent space discretization approach, then a further step is required to recover the original representation in its observation space through the standard GP mapping.

VI. Experimental Details

We design experiments to evaluate the proposed BN-based grasp modeling approach. The goals of the experiments are as follows.

1) To evaluate task classification performances of the proposed generative modeling approach using BNs (see Section VI-C).

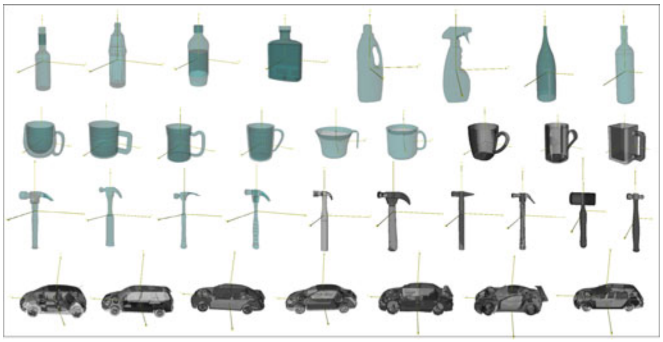


Fig. 5. Experiment object models.

TABLE I
Total Number of $\mathcal{OG}$s and Sample Sizes of Each Task for the Three-Hand Models

	Total	Handover	Pouring	Tool Use	Dish Washing	Playing
Human	4760	996*2	860*2	986*2	457*2	101*2
Armar	6767	1488*2	1565*2	565*2	767*2	66*2
Schunk	7865	2033*2	2135*2	960*2	759*2	296*2

For each task, there are equal amount of positive and negative data, therefore the sample size is N_{positive}*2.

2) To evaluate the ability of the BNs to encode task-specific requirements on grasping parameters (see Section VI-D).

3) To evaluate the two proposed discretization methods on the high-dimensional data (see Section VI-E).

4) To illustrate the application of the proposed approach in a goal-directed grasp imitation scenario (see Section VI-F).

The focus of the experiments is to evaluate learning algorithms underlying the reasoning system. The execution on real robots using the proposed system has been done in [55].

A. Data Collection

We evaluate our grasp modeling framework on three hands: Human hand (20 DoFs), Armar humanoid hand (11 DoFs) [56], and three-finger Schunk dexterous hand (7 DoFs) [57]. For each hand, we generate a large set of stable grasp data on the objects of four different classes: bottle, mug, hammer, and sedan (toy car) (see object models in Fig. 5). The total numbers of $\mathcal{OG}$'s are listed in Table I. For each $\mathcal{OG}$, a human tutor observes the 3-D configuration in a grasp simulator and provides labels of one or a set of afforded tasks from the task set (handover, pouring, tool use, dish washing, playing). The semantics of the five tasks are described in Fig. 2. It is important to note that the task requirements on the grasped object and the grasp configurations are defined subjectively by the human user. Such task preferences can change from one user to another, or adapt over time to different situations. The BN-based modeling approach allows such knowledge updating in a life-long learning process of the artificial agent.

For each task, we extract balanced positive and negative examples from the total dataset, and this will be the basis for training task-specific BNs. Table I lists the sample sizes for each task for the three-hand models. We note that there is an

```
Algorithm VI.1: FIND ($\mathcal{X}'$, $\mathbf{X}$, $\mathbf{X}_d$) GIVEN ($\mathcal{X}$, $\mathbf{T}$, $\mathbf{X}$)
  where $\mathcal{X} = \{O, A, C\}$ is the feature set
  $\mathbf{T} \in \Re^{N \times 1}$ is the task labels for $N$ data points
  $\mathbf{X} \in \Re^{N \times G}$ is the data for all variables in $\mathcal{X}$
  $\mathcal{X}' \subset \mathcal{X}$ is the selected subset of variables
  $\mathbf{X}_d \in \Re^{N \times n}$ is the discrete representation of $\mathbf{X}$

  1) Feature Selection
  $\mathcal{X}' \leftarrow$ Feature Selection Algorithm, HITON($\mathcal{X}, \mathbf{T}, \mathbf{X}$)
  $\mathbf{X} \leftarrow$ Update to only include selected variables in $\mathcal{X}'$

  2) Data Discretization
  for $X \in \mathcal{X}'$, and $\mathbf{x} \in \Re^{N \times D}$ being the data of $X$
    if $X$ is a discrete variable, i.e. $obcl$
      then $\mathbf{x}_d \leftarrow \mathbf{x}$; no discretization needed
      else if $X$ is a continuous variable and $Dim(X) = 1$
      then
        $\mathbf{x}_d \leftarrow \mathbf{x}$; equally divided and modeled with GMM
        number of Gaussian components is determined using BIC
      else if $X \in \{size, npos, dir, coc\}$
      then $\mathbf{x}_d \leftarrow \mathbf{x}$; using SOM-based discretization
      else if $X$ is $fcon$
      then
        if using SOM-based discretization
          then $\mathbf{x}_d \leftarrow \mathbf{x}$;
          else if using GP-LVM-based discretization
          then
            $\mathbf{x}_d \leftarrow \mathbf{x}_l \leftarrow \mathbf{x}$;
            where $\mathbf{x}_l \in \Re^{N \times q}$ is low-dimensional data
            $\mathbf{X} \leftarrow$ Update $fcon$ data with $\mathbf{x}_l$
```

Fig. 6. Algorithm VI.1 for preprocessing the data.

```
Algorithm VI.2: FIND ($BN$, $ROC_{BN}$, $KLR$, $ROC_{KLR}$) GIVEN ($\mathbf{T}, \mathbf{X}, \mathbf{X}_d, \mathcal{X}'$)
  where $\mathbf{T} \in \Re^{N \times 1}$ is the task labels for $N$ data points
  $\mathbf{X} \in \Re^{N \times G}$ is the data for selected variables in $\mathcal{X}'$
  $\mathbf{X}_d \in \Re^{N \times n}$ is the discrete representation of $\mathbf{X}$

  $M = 30$; $\leftarrow$ user specify number of trials
  for $i = 1, 2, \ldots M$
    $idx_{train}, idx_{test} \leftarrow$ 20% Hold-out Split ($\mathbf{T}$)

    1) Train-Test KLR model with full observation, i.e. all $\mathcal{X}'$
    $\theta_* \leftarrow$ 10-trial cross-validation to optimize KLR parameter($idx_{train}$)
    $KLR_i^f \leftarrow \mathbf{T}(idx_{train}), \mathbf{X}(idx_{train}), \theta_*$;
    $ROC_{KLRi}^f \leftarrow KLR_i^f, \mathbf{T}(idx_{test}), \mathbf{X}(idx_{test})$;

    2) Train-Test KLR model with partial observation (no Constraint features)
    $\theta_* \leftarrow$ 10-trial cross-validation to optimize KLR parameter($idx_{train}$)
    $\mathbf{X}^p \leftarrow$ removing constraint features from $\mathbf{X}$
    $KLR_i^p \leftarrow \mathbf{T}(idx_{train}), \mathbf{X}^p(idx_{train}), \theta_*$;
    $ROC_{KLRi}^p \leftarrow KLR_i^p, \mathbf{T}(idx_{test}), \mathbf{X}^p(idx_{test})$;

    3) Train-Test BN model
    $BN_i \leftarrow \mathbf{T}(idx_{train}), \mathbf{X}_d(idx_{train})$;
    $ROC_{BNi}^f \leftarrow BN_i, \mathbf{T}(idx_{test}), \mathbf{X}_d(idx_{test})$;
    $\mathbf{X}_d^p \leftarrow$ removing constraint features from $\mathbf{X}_d$
    $ROC_{BNi}^p \leftarrow BN_i, \mathbf{T}(idx_{test}), \mathbf{X}_d^p(idx_{test})$;
```

Fig. 7. Algorithm VI.2 to train and test BN and KLR models for task classification.

imbalance in sample sizes across five tasks. This is explained by different numbers of objects that afford different tasks and how constrained a task's requirements is. For example, for playing task, only sedan can be grasped. For human and Armar hands, good grasps are those from the top with the thumb and index fingers facing the driving direction of the car (see Fig. 2). This is why the "playing" task on human and Armar hands has the least amount of data.

B. Data Preprocessing

The collected data are a set of $\mathcal{OG}$ configurations instantiated with a set of $\{O, A, C\}$ features together with the task labels T provided by human experts. Fig. 8 lists a set of features used in this paper. We note that, for the purpose of task classification, the listed features are not equally discriminative. Some variables may be very relevant for one task, but not relevant at all for another. To remove most redundant features and alleviate the effect of the curse of dimensionality, we implement a feature selection step during data preprocessing. Furthermore, most $\{O, A, C\}$ features are continuous-real-valued data and some of them are multivariate variables. As stated in Section IV-B, to learn the structure of the BN from data, we need to discretize the continuous data.

Fig. 6 shows the pseudocode for the two steps of data preprocessing. For feature selection, we applied the optimal variable selection algorithm HITON proposed by [58]. We select HITON as the feature selection method because it is sample efficient and it is based on inducing the Markov blanket of the target class variable, which is the minimum set of variables that make the other variables conditionally independent from the target variable. Implicitly, the variables in the Markov blanket are a set of variables that are most relevant to the class variable. HITON works by first inducing the Markov blanket of the target variable to be classified (in this paper it is task T). Then, support vector machine is used to further remove the unnecessary variables in the Markov blanket in a greedy hill-climbing fashion. The left side of Fig. 8 are the task-specific BNs for the three hands, where the selected variables are displayed. As expected different tasks and hands have resulted in different set of selected variables. For example, since *playing* with the toy car only requires the Schunk hand to grasp the car from the top, i.e., requirements on the unified grasp position $npos$, the feature selection algorithm does not select the orientation of the hand dir to be relevant. On the other hand, dir is included for both human-like hands (human and Armar) because *Playing* task also requires thumb and index finger to be facing the driving direction of the car, thus constrain the possible values of dir.

In the second step of preprocessing, the pseudocode in Fig. 6 specifies the data discretization methods used for each variable. For 1-D continuous variables, we first apply the equal width binning to cluster the continuous data into K intervals, and then train a GMM model based on this initial clustering. The free parameter K is selected from a given range of Ks that minimize the Bayesian information criteron. For multivariate variables $\{npos, dir, size, coc, fcon\}$, we apply SOM-based discretization. Number of cluster K is determined by minimizing the Davies–Bouldin index in (4). Only one variable $fcon$ is a high-dimensional variable that potentially has a lower dimensional

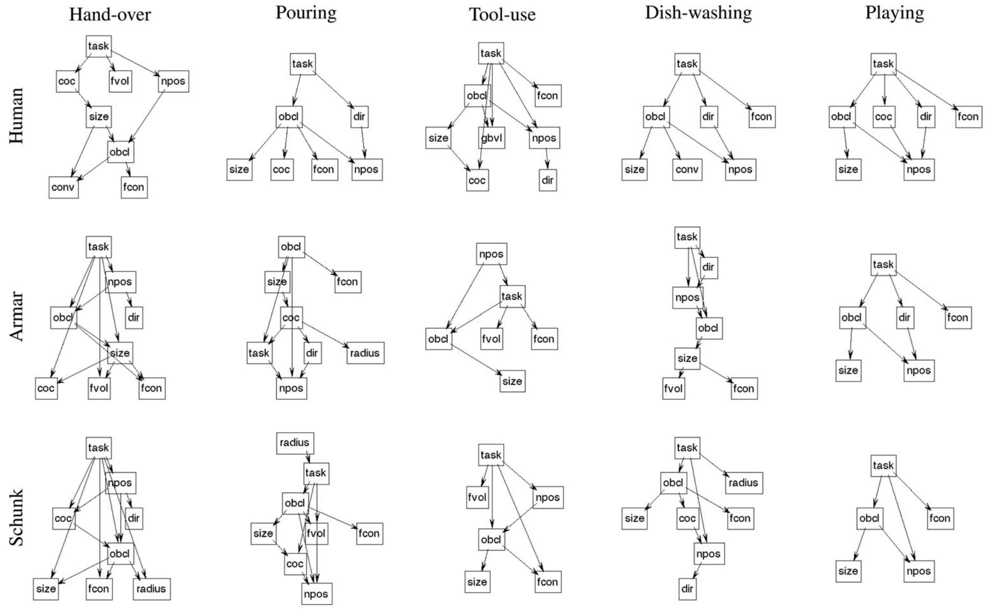


Fig. 8. DAGs of SOM discretization for three hands and five tasks.

	Name	Dim	Description	Definition
T	task		Binary task identifier	: A discrete class, describing a task like *playing*.
O_1	obcl		Object category	: A discrete class, describing an object category like *mug*.
O_2	size	3	Object dimensions	: describing the 3 object dimensions (width, height, depth).
O_3	conv	1	Object Convexity	: describing the object convexity, as the ratio between decomposition and bounding box volumes.
O_4	ecce	1	Object Eccentricity	: describing the object eccentricity, i.e. the ratio between minimum and maximimum object dimensions.
A_1	dir	4	Hand orientation	: describes the hand orientation as a Quaternion, normalized to the object's pose.
A_2	npos	3	Unified grasping position	: describes the position from which the grasp is triggered, projected on a unified sphere.
A_3	radius	1	Distance to object center	: describes the distance (in final configuration) from the palm position to the object center.
A_4	fcon	20/11/7	Finger configuration	: describes the joint configuration of the hand, dependent on its number of degrees-of-freedom.
C_1	fvol	1	Free volume	: describes a measure of free volume, computed from contact points, and shape decomposition.
C_2	coc	3	Center of contact	: describes the center of contact points in which the hand touches the object's surface.
C_3	gbvl	1	Grasped part volume	: describes the ratio of the object part that is in contact, compared to the overall volume of the object.

Fig. 9. Feature abbreviations, dimensionality, and descriptions. Note: $task$ and $obcl$ are discrete variables and therefore not assigned with any dimensionality.

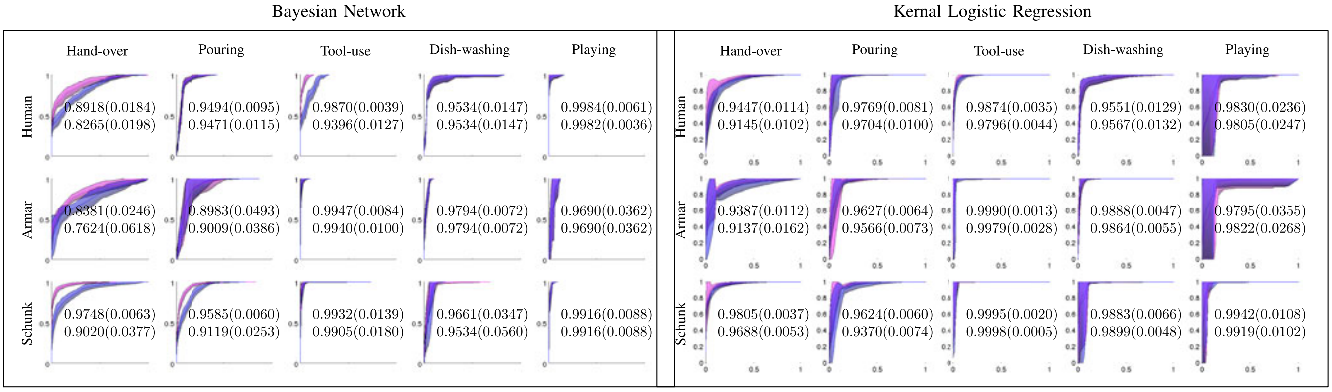


Fig. 10. Goal inference: The average ROC curves (true positive versus false positive rates) obtained with 30 trials of hold-out (20% testing) cross validation for the four tasks and three hands. The shaded region represents two standard deviation on true positive rate (*Y*-axis). Two colors represent two different observation conditions: pink—full; blue—partial, where constraint features, such as *free volume* and *grasped part shape* are missing. The numbers on the graph reports the average and standard deviations of AUCs of the full and partial observations, respectively. Note that sometimes the pink and blue curves show same results because C features were not included in the feature set (see selected features on the DAGs in Fig. 8).

intrinsic space. We, therefore, also apply the latent space discretization approach on the data of $fcon$. In Section VI-E, the two discretization methods will be evaluated by examining the reconstruction of $fcon$ in the original observation space.

The following sections will present the experiment using the preprocessed data.

C. Experiment I: Task Classification With Bayesian Networks and Kernel Logistic Regressions

For evaluating the classification performance, we compare the BNs with the discriminative model KLRs under two observation conditions: the full observation when the values of all selected variables are observed and when only given partial knowledge. To get class probabilities with two observation conditions, we need to train two separate KLR models, KLR^f model with a full feature set, KLR^p model with partial feature set; and one BN model with full feature set. We do not need to train a separate BN model with partial feature set because task probability can be inferred with partial observations using BNs. Note that in some situations, the selected feature set $\mathcal{X}'$ during first step of data preprocessing (see Fig. 6) does not contain C variables. In such cases, the two observation conditions are equivalent thus return same classification results.

Fig. 7 shows the algorithm for training and testing the BN and KLR models for the purpose of task classification. Thirty trials are performed in which each trial divides the total dataset into 20% of testing and 80% of training data. Note that this division of the total dataset is not division among different objects; therefore, the testing data may contain same and different objects from the training set. The experiment is designed to evaluate how the system can generalize not only on novel objects but on new grasping strategies operated on the same and novel objects as well.

For KLR models, we use the task label together with the continuous observations $[\mathbf{T}, \mathbf{X}]$. In each of the 30 trials, we first find the optimal parameters θ_* (the kernel bandwidth and the regularization term) for KLR through a grid search using a ten trials cross validation on the training set. Once θ_* is found, we train the KLR model with the total training set, and test on the 20% of testing set to estimate the performance of the classifiers.

For the BNs, the observations are discrete $[\mathbf{T}, \mathbf{X}_d]$, where $\mathbf{X}_d$ is obtained using the SOM-based discretization method. We do not search for the optimal parameters here, as the parameters that most affect the performance of the BNs are the resolutions of discretization for all the continuous variables. In addition searching for the optimal resolutions for all the variables will take prohibitively long time. In each of the 30 trials, we use the training data to first obtain the structure of the network, and then update the parameters (local conditional probability tables).

We use the area (AUC) under the receiver operator characteristic (ROC) curve as the performance metric for both KLRs and BNs. During testing, each classifier outputs the task probability given the specified observation condition $p(T|X)$, which are then thresholded at different levels to derive the ROC curves. Fig. 10 shows the average ROC curves over 30 trials under two different observation conditions (pink color for the full observation, and blue for the partial observation). The left columns in Fig. 10 show results from BNs, while the right from KLR models. The corresponding structures of the BNs are shown in Fig. 8.

In general, we observed that BN models have quite good classification performance especially under full observation conditions (AUCs ≤ 0.84). Among the five tasks, handover is the most challenging one for classification, especially when constraint features, such as $fvol$ and coc are not observed (AUCs is 0.76 for Armar hand). This is because handover task is the most "generous" task that has least requirements on O (all objects affort handover task) and A (one can grasp from any direction) variables. In addition, the only requirement is to leave enough uncovered surface for safe regrasp that is reflected mostly by the C variable free volume $fvol$. The BN structure for the handover task (see Fig. 8) reflects this fact, $task$ has direct connection to $fvol$.

KLR models report higher classification performance (AUCs > 0.91) for handover task in both full and partial observation conditions. This reflects the "side effect" of data discretization (for BN training), especially on C variables. Discretization always leads to loss of information. When the resolution is low (i.e., very few discrete states), the variance in the original continuous domain that is discriminative may be smoothed out. On the other hand, for the variables that are not discriminative, a high resolution will jeopardize the classification performance due to the curse of dimensionality. In the future, the granularity for discretizing each continuous variable should be optimized in the inner loop of the nested cross validataion similar to KLR models (see Fig. 7).

Additionally, we observed a large variance in true positive rates when classifying playing task with the KLR models. This may be due to the limited training data for this task (see Table I), but high-dimensional feature space (dimensionality is the sum of all selected O, A, C variables). However, such limitation is alleviated in the proposed BN-based modeling approach. Through data discretization, the high-dimensional feature space is compressed into a low-dimensional compact representation. Furthermore, in a learned BN, the less discriminative variables are further pruned through the conditional independence structures of the network. In summary, this experiment confirmed that the proposed BN-based grasp modeling approach can provide highly accurate and robust (with small variation in true-positive rate) task classification performance.

D. Experiment II: Inference on Unified Grasp Position

In addition to task classification, i.e., inferring $p(T|X)$, another important purpose of the proposed modeling approach is to obtain task conditioned distribution of the grasp parameters $p(X|T)$ so that grasp planning can be performed in a goal-directed manner. This conditional distribution should reflect the requirements of a given task in the original observation space of the grasping parameters. In this experiment, we examine if the learned BN models can successfully encode these requirements using an intuitive action variable, the unified grasp position $npos$.

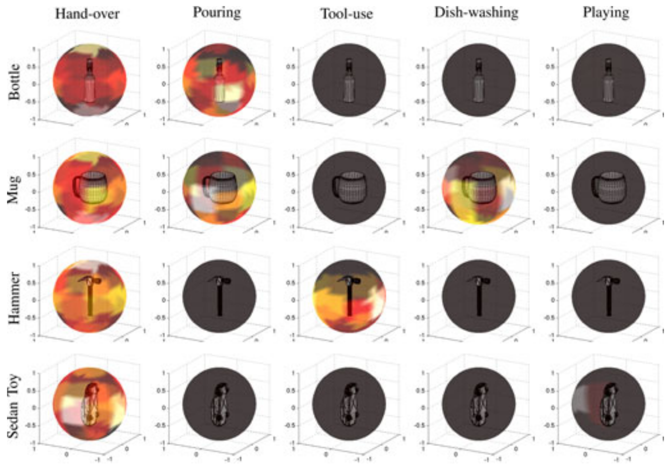


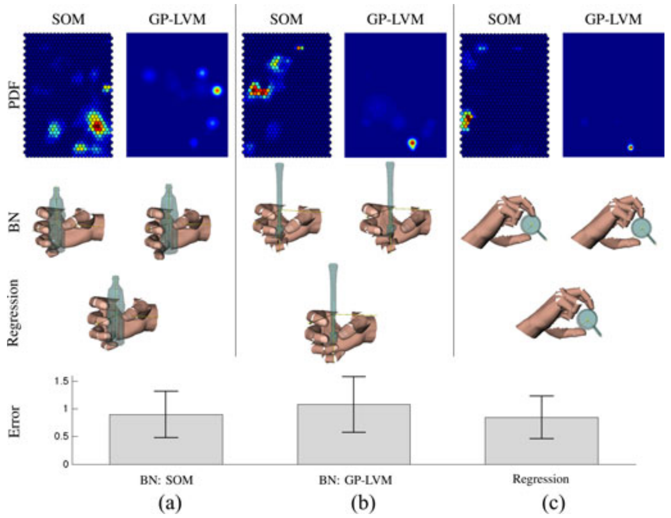
Fig. 11. Likelihood maps of unified grasp position conditioned on tasks and object classes $p(npos|task, obcl)$.

A data point of $npos$ is a 3-D vector that lies on a unit sphere around an object, and it indicates from which direction the hand approaches and grasps the object. It is important to note that the constraint of a given task is often encoded by combination of multiple features, e.g., one should not grasp this object from this position $npos$, in this orientation dir, and with this joint configuration $fcon$, etc. However, a distribution over the 4-D quaternion orientation dir and high-dimensional finger configurations $fcon$ are difficult to visualize. We therefore select $npos$ for this experiment.

We evenly sample a set of points $\mathbf{x}$ on the unified sphere, where $npos$ lies. For each sampled point, a conditional likelihood is obtained given the five tasks and the four object classes $p(\mathbf{x}|task, obcl)$ using the human hand BNs (the first row in Fig. 8). The values of likelihoods are normalized across all five tasks and four object classes.

Fig. 11 shows the resulting *likelihood maps,* where the lighter color indicates higher probability; therefore, a good grasp at this point. In general, the resulting maps successfully reflect the constraining effects of the tasks over both object selection and grasp planning. For example, for object selection, we observe pouring task exclude the hammer and the sedan toy car that do not afford to pour. This is reflected by the dark likelihood maps around them. Handover can be applied to all objects therefore no complete dark spheres for this task. For grasp planning, we observe dark regions on the top of the maps when pouring with bottles and mugs. This reflect the constraint of the pouring task, no grasp should block the top, opening part of the object. Similarly, when using the hammer as a tool, one should avoid the head of the hammer as it is the functional part. When the goal is to handover a hammer, the preferred grasps are the ones around the hammer head so that handle can be exposed for convenient regrasp. Similar constraining effects are also observed in the likelihood maps from the other two-hand models. However, due to space limits, we will not show them this paper.

In summary, this experiment demonstrated the strength of the proposed framework, by modeling the embodiment-specific task space using a probabilistic network, we have learned not only the affordances of the objects but the robots own motor capability associated with a task as well. Such motor capabilities are encoded in the continuous distributions of the grasp action parameters conditioned on the assigned task. Even though the KLR approach performs slightly better for most tasks, this difference in performance does not overcome the benefits of the BN approach such as being able to use a single model for all types of inference and providing an interpretability of the data.



(a) (b) (c)

Fig. 12. Predict final grasp configuration $fcon$: (a)–(c) case 1–3 shows the predicted grasp configurations given three cases of objects and hand poses. Row 1 shows the probability distribution conditioned on object and hand pose on the SOM and the latent space. Row 2 shows prediction of $fcon$ from the two distributions above. Row 3 shows the prediction of $fcon$ by GP regression. Row 4 shows the mean and std of the reconstruction errors (measured as Euclidean distance between predicted and true values) over 398 cases of prediction.

E. Prediction on Finger Configuration

Section V introduced two generative soft discretization approach for high-dimensional data to support learning of BN that models the conditional relationships of a set of variables. An ideal discretization should provide a compact representation of the data in the discrete space, and at the same time maintain, to a large extend, its original distribution in a high-dimensional space. The goal is then to recover the high-dimensional data of one variable given the observation of the other related variables provided the relations being accurately modeled by the BN. We used the Euclidean distance between the predicted and true values of the target variable as the error metric to evaluate the discretization methods $||\mathbf{y}_{\text{pred}} - \mathbf{y}_{\text{true}}||$.

The purpose of the experiment in this section is to examine the reconstruction accuracy using the high-dimensional variable, the finger configuration $fcon$ when grasping the object given the observed hand pose $\{npos, radius, dir\}$, and the object $size$ and the object class $obcl$. We used 80% (1594 cases) of the human handover dataset as the training data. We train two Naive Bayes networks each with the $fcon$ being discretized with one of the two discretizaiton methods. In the naive Bayes network, the target variable is $fcon$, and other attributes are $\{size, npos, radius, dir\}$, conditionally independent given the value of $fcon$. To compare this with a baseline approach, we also train a GP regression model with the input the continuous data of $\{size, npos, radius, dir\}$, and output the high-dimensional data of $fcon$.

Given the two trained BNs, we test the prediction of $fcon$ given observed $\{obcl, size\}$ and hand pose $\{npos, radius, dir\}$ using the 398 testing cases. Fig. 12(a)–(c) shows three sample cases. For each case, we first obtain the conditional distribution $p(fcon|size, npos, radius, dir)$ by inference in the BN. The corresponding probabilistic density in the continuous space is shown on row 1 of Fig. 12. The predicted value of the finger configuration in the original observation space $\mathbf{y}_{\text{pred}}$ (visualized in row 2 of Fig. 12) is then obtained using the weighted sum methods as explained in Section V-A. We also obtained the prediction on 398 testing cases using the trained GP regression model. The three regression results are shown in row 3 of Fig. 12.

By observing the three sample cases in Fig. 12, we can qualitatively evaluate the reconstruction results. Row 1 shows the conditional distribution of $fcon$. We see given different object and hand poses, $fcon$ has different PDF in the continuous space. In addition, high probable distribution sometimes (Case 1) spread over multiple states, indicating BNs are "not very certain," which state the $fcon$ belongs to. The weighted sum method takes into account this uncertainty and produces prediction in original space. When comparing the reconstructed grasps in rows 2 and 3, we see that the SOM-based discretization method results in finger configurations that conform slightly better to the object surface shapes than those from the GP-LVM-based latent space discretization. SOM method has similar performance to the baseline results from the regression method. The error metric plotted in row 4 provides a quantitative evaluation that is consistent with our observation above. Although SOM method has lower mean error than GP-LVM, the differences are not significant enough to conclude one method as superior to the other.

For runtime evaluation, with OS X. Snow Leopard, 2.4-GHz Intel Core 2 Duo processor, and an unoptimized MATLAB implementation on 1594 data points of the 20-D $fcon$, SOM-based discretization process (12.31 sec) is much faster than the GP-LVM-based latent space discretization (97.43 s). Most computational time (98%) is due to the training of the GP-LVM.

To interpret the results in this section, we need to carefully point out the general challenges in the problem of data reconstruction after discretization, and the particular challenges in predicting finger-grasp configurations based on limited observation of object attributes and hand poses. In general, to reconstruct real-valued vectors from probability membership assignments to the discrete states is a big challenge in vector quantization and defuzzification (in fuzzy logic). More finely spaced discretization intervals (i.e., more discrete states) will reduce errors in data reconstruction, however, at the price of increased computational cost. In addition, high-resolution discretization also increases the difficulty in BN training because the number of parameters (the entries of the conditional distribution tables) increases, which requires larger set of training data to model the valid conditional distributions.

In the experiment of predicting finger configurations, we have assumed that the finger-grasp configuration is completely determined by very gross object property ($size$) and the pose of the hand. This assumption is not that accurate because the detailed finger positions are subject to geometry of the contact surface on the object. This dependence is very local and is difficult to model with data-based method such as BN. As a result, the prediction from BNs is only a "gross" estimation of finger position, which can, in many cases, penetrate the surface of the object.

Nonetheless, the results are still satisfactory as an initial guess for subsequent optimization. For example, the PDF (row 1 in Fig. 12) can work as a hand pose prior similar to [59]. Such a prior can be combined with other physical constraints, such as object surface geometry and hand kinematics to optimize the synthesis of grasps and even improve the robustness of hand pose tracking.

F. Goal-Directed Grasp Imitation

Finally, we illustrate how the proposed modeling framework can be used for goal-directed grasp imitation on two robot platforms. To achieve this, a robot should be equipped with a perception system to extract O, A, C features from human demonstration, a BN-based reasoning system for goal-inference and goal-directed grasp planning, and a grasp execution system that plans and executes collision-free reaching motion to place the hand at the selected grasping pose. We assume the perception [60] and execution systems [55], [61] are available, and the focus of this illustration is to show how the reasoning system allows a goal-directed grasp imitation.

For each robot, the reasoning system consists of grasp models of two embodiments: one of the human teacher (BN^{Human}) and one of the robot itself (BN^{Robot}). The robot uses BN^{Human} to infer the grasp intention t^*. Given the estimated goal t^*, grasp imitation is accomplished by executing the grasping strategy (selecting the object to grasp and how to grasp it) according to the learned grasping policy suitable for the robot's own embodiment BN^{Robot}. Fig. 13 illustrates this process on two robot platforms. Note that this is not a real-time implementation but a simple flowchart to demonstrate how to use the BN-based grasp reasoning system. First, the robots observe the human demonstration, and extracts the object and action parameters $\mathbf{o}, \mathbf{a}, \mathbf{c}$. They both use a human-specific network BN^{Human} to infer the intention of the human, and estimate the most likely goal is $t^* =$ *pouring* with the mug. Second, the robots need to select the most suitable object for pouring task among the three novel objects: a hammer, a toy car, and a mug. This requires the use of the robots' own experiences encoded in $\text{BN}^{\text{Schunk}}$ and BN^{Armar} to infer $p(\mathbf{o}|t^*)$. The most likely object in both cases is the mug. Note that the inferred probability from two robot BNs are slightly different, which indicates their different experiences with these object categories. Finally, the robots need to search for the best grasp strategy on the mug that affords pouring, and execute it on their platforms. Assuming there are a set of preplanned grasp hypotheses parameterized by $\mathbf{a}$. The robot can then use $p(\mathbf{a}|t^*, \mathbf{o}^*)$ to select the most suitable one in order to pour with the selected mug. Note here, the optimal grasping policies (including selected object and grasp) may be

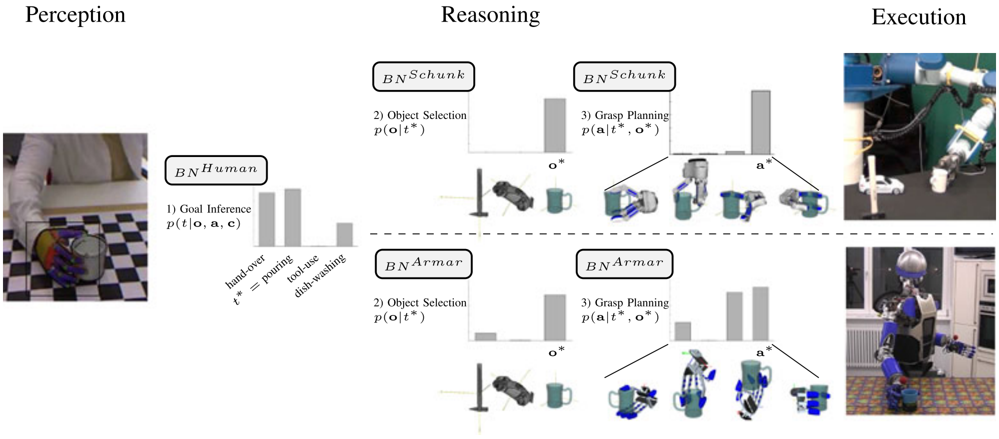


Fig. 13. Process of a goal-directed imitation. This is not a real-time implementation. Image of perception is human-grasp demonstration setup based on the tracking system in [60]. Execution images on the robot platforms Tombatossals (top) and Armar (bottom) are from [61] and [55], respectively. Armar images are generated with the proposed system.

different from the exact human demonstration. However, this is acceptable as long as the task goal is satisfied, and the action is compatible with the embodiment of the robot.

VII. Complexity

The complexity of the proposed method comes in two different parts: the training and the testing. In training, the bottlenecks of the complexity are on learning the discrete representation using the GP-LVM, and on learning the structure using the BN. For the GP-LVM, the complexity is $\mathcal{O}(Nm^2)$, where N is the number of data points, and m is the number of inducing points, or in our interpretation, the number of clusters. For the BN structure learning, an estimation of the worst-case complexity of the greedy search algorithm used in this paper is $\mathcal{O}(n^2 \cdot 2(n-1)!)$, where n is the number of the variables. Testing with GP-LVM is linear in the number of data points $\mathcal{O}(N)$. Testing with BN with the junction tree algorithm has the complexity of $\mathcal{O}(n \exp(W-1))$, where W is the size of largest clique created during inference. In our experiment, the inference with BNs takes a fraction of a second.

VIII. Conclusion

We have presented a framework for representing grasping in a task- and embodiment-specific setting. The approach allows for generalization over task, object, and action, which introduces a concept of embodiment-specific concept affordances. This allows us to perform and plan grasps across different embodiments as the specific affordances are independent across embodiments. We specifically exploit this to perform imitation learning, allowing several different robots to be "taught" by the same instructor and executing the same task, while respecting its unique constraints.

Our model is formulated in a Bayesian setting, allowing for inference from partially observed data and providing a notion of uncertainty. What enables this is a rich parameterization of the environment, integrating task, action, and object information within the same model. We have developed a method that learns an intermediate discrete representation from complex data, avoiding the challenges posed by the orginal high dimensionality.

The proposed framework scales well with the number of tasks since one can obtain an additional network whenever a new task is added. The scalability of the objects depends on the object representation used in the network. Since we do not use object instances, but rather object classes and geometrical properties that are able to handle the intraclass variabilities, the method should scale with the limited number of object classes that a robot usually encounters in its environment.

This study opens up a broad avenue for future research. First, the framework can be extended to incorporate other sensory signals such as tactile arrays [62], and force/torque sensors on the fingers. This way, the model can be used to infer stability of a grasp on the fly, and to provide important information for online task-oriented grasp adaptation. Second, the BNs presented here model the static grasp before lifting the object. This can be ambiguous in goal inference since one grasp configuration can potentially afford multiple tasks. One solution is to exploit the information contained in the dynamic manipulation of an object after lifting, e.g, by introducing dynamic BNs. The learned model can then be used for task recognition during human demonstration and action reproduction on robot platforms. Finally, with the generative models, the system can detect novel observations; hence, allows us to build an active learning system that self-updates and adapts to new situations.

References

[1] A. Alissandrakis, C. Nehaniv, and K. Dautenhahn, "Correspondence mapping induced state and action metrics for robotic imitation," *IEEE Trans. Syst., Man, Cybern., B Cybern.*, vol. 37, no. 2, pp. 299–307, Apr. 2007.

[2] A. Alissandrakis, C. Nehaniv, and K. Dautenhahn, "Imitation with alice: Learning to imitate corresponding actions across dissimilar embodiments," *IEEE Trans. Syst., Man, Cybern., A,Syst., Humans*, vol. 32, no. 4, pp. 482–496, Jul. 2002.

[3] A. N. Meltzoff, *Elements of a Developmental Theory of Imitation*. Cambridge, MA, USA: Cambridge Univ. Press, 2002, pp. 19–41.

[4] D. Verma and R. Rao, "Goal-based imitation as probabilistic inference over graphical models," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf, and J. Platt, Eds. Cambridge, MA, USA: MIT Press, 2006, pp. 1393–1400.

[5] M. R. Dogar, M. Cakmak, E. Ugur, and E. Sahin, "From primitive behaviors to goal-directed behavior using affordances," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 729–734.

[6] S. Calinon, F. Guenter, and A. Billard, "Goal -directed imitation in a humanoid robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 299–304.

[7] D. Verma and R. P. N. Rao, "Planning and acting in uncertain environments using probabilistic inference," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 2382–2387.

[8] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory–motor coordination to imitation," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 15–26, Feb. 2008.

[9] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 1988.

[10] D. Song, K. Huebner, V. Kyrki, and D. Kragic, "Lea rning task constraints for robot grasping using graphical models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 1579–1585.

[11] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Multivariate discretization for Bayesian network structure learning in robot grasping," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1944–1950.

[12] D. Song, C. H. Ek, K. Huebner, and D. Kragic, "Embodi ment-specific representation of robot grasping using graphical models and latent-space discretization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2011, pp. 980–986.

[13] C. H. Ek, D. Song, and D. Kragic, "Learning conditional structures in graphical models from a large set of observation streams through efficient discretisation," in *Proc. IEEE Int. Conf. Robot. Autom., Workshop Manipul. Uncertainty*, May 2011.

[14] D. Song, N. Kyriazis, I. Oikonomidis, C. Papazov, A. Argyros, D. Burschka, and D. Kragic, "Predicting human intention in visual observations of hand/object interactions," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 1608–1615.

[15] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Springer Handbook of Robotics*, New York, NY, USA: Springer, 2008, pp. 1371–1394.

[16] C. Breazeal, J. Gray, and M. Berlin, "An embodied cognition approach to mind reading skills for socially intelligent robots," *Int. J. Robot. Res.*, vol. 28, no. 5, pp. 656–680, May 2009.

[17] D. B. Grimes and R. Chalodhorn, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference, " in, *Robotics: Science and Systems*. Cambridge, MA, USA: MIT Press, 2006.

[18] A. Billard and S. Schaal, "Robust learning of arm trajectories through human demonstration," in *Proc. Int. Conf. Intell. Robots Syst.*, 2001, pp. 734–739.

[19] S. Bitzer, I. Havoutis, and S. Vijayakumar, "Synthesising novel movements through latent space modulation of scalable control policies," in *Proc. Int. Conf. Simul. Adapt. Behav.*, 2008, pp. 199–209.

[20] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 763–768.

[21] C. L. Nehaniv and K. Dautenhahn, *The Correspondence Problem*. Cambridge, MA, USA: MIT Press, 2002, pp. 41–61.

[22] S. Kang and K. Ikeuchi, "Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps," *IEEE Trans. Robot. Autom.*, vol. 13, no. 1, pp. 81–95, Feb. 1997.

[23] H. Kjellström, J. Romero, and D. Kragic, "Visual recognition of grasps for human-to-robot mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 3192–3199.

[24] P. Azad, T. Asfour, and R. Dillmann, "Toward an unified representation for imitation of human motion on humanoids," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 2558–2563.

[25] K. Hsiao and T. Lozano-Perez, "Imitation learning of whole-body grasps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 5657–5662.

[26] A. M. Schmidts, D. Lee, and A. Peer, "Imitation learning of human grasping skills from motion and force data," in *Proc. IEEE/RSJ Int. Conf Intell. Robots Syst.*, 2011, pp. 1002–1007.

[27] D. Wolpert and M. Kawato, "Multiple paired forward and inverse models for motor control," *Neural Netw.*, vol. 11, nos. 7/8, pp. 1317–1329, Oct. 1998.

[28] Y. Demiris and M. Johnson, "Distributed, predictive perception of actions: A biologically inspired robotics architecture for imitation and learning," *Connection Sci.*, vol. 15, no. 4, pp. 231–243, 2003.

[29] R. Rao, A. Shon, and A. Meltzoff, "A Bayesian model of imitation in infants and robots," in *Imitation and Social Learning in Robots, Humans, and Animals*. Cambridge, U.K.: Cambridge Univ. Press, 2004, pp. 217–247.

[30] D. B. Grimes and R. P. N. Rao, "Learning actions through imitation and exploration: Towards humanoid robots that learn from humans," in *Creating Brain-Like Intelligence* (Lecture Notes in Computer Science Series), vol. 5436. New York, Ny, USA: Springer, 2009, pp. 103–138.

[31] E. Oztop, D. Wolpert, and M. Kawato, "Mental state inference using visual control parameters," *Cogn. Brain Res.*, vol. 22, no. 2, pp. 129–151, 2005.

[32] J. G. Greeno, "Gibson's affordances," *Psychol. Rev.*, vol. 101, no. 2, pp. 336–342, 1994.

[33] V. Krunic, G. Salvi, A. Bernardino, L. Montesano, and J. S. Victor, "Affordance based word-to-meaning association," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 806–811.

[34] M. Toussaint, N. Plath, T. Lang, and N. Jetchev, "Integrated motor control, planning, grasping and high-level reasoning in a blocks world using probabilistic inference," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 385–391.

[35] D. Jain, L. Mösenlechner, and M. Beetz, "Equipping robot control programs with first-order probabilistic reasoning capabilities," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 3130–3135.

[36] A. Shon, K. Grochow, A. Hertzmann, and R. P. Rao, "Learning shared latent structure for image synthesis and robotic imitation," in *Advances in Neural Information Processing Systems*, 2005, pp. 1233–1240.

[37] R. Zöllner, M. Pardowitz, S. Knoop, and R. Dillmann, "Towards cognitive robots: Building hierarchical task representations of manipulations from human demonstration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2005, pp. 1535–1540.

[38] M. Novotni and R. Klein, "Shape retrieval using 3D zernike descriptors," *Comput. Aided Des.*, vol. 36, no. 11, pp. 1047–1062, 2004.

[39] K. Huebner, "BADGr: A toolbox for box-based approximation, decomposition and GRasping," *Robot. Auton. Syst.*, vol. 60, no. 3, pp. 367–376, 2012.

[40] M. Yamada, M. Sugiyama, and T. Matsui, "Semi-supervised speaker identification under covariate shift," *Signal Process.*, vol. 90, no. 8, pp. 2353–2361, 2010.

[41] L. D. Fu and I. Tsamardinos, "A comparison of Bayesian network learning algorithms from continuous data," in *Proc. AMIA Annu. Symp.*, 2005, p. 960.

[42] T. Kohonen, *Self-Organizing Maps*, vol. 30. Berlin, Germany: Springer, 1995.

[43] N. D. Lawrence, "Probabilistic non-linear principal component analysis with Gaussian process latent variable models," *J. Mach. Learn. Res.*, vol. 6, pp. 1783–1816, Nov. 2005.

[44] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.

[45] P. Leray and O. Francois, "BNT structure learning package: Documentation and experiments," Université de Rouen, Tech. Rep., Laboratoire PSI - INSA Rouen- FRE CNRS 2645, Mont-Saint-Aignan, France, 2006.

[46] C. Chow and C. Liu, "App roximating discrete probability distributions with dependence trees," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 3, pp. 462–467, May 1968.

[47] I. Ebert-Uphoff, "A probability-based approach to soft discretization for Bayesian networks," Georgia Inst. Technol., Tech. Res. Rep. GT-ME-2009-002, Atlanta, GA, USA, 2009.

[48] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map." *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.

[49] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[50] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. Cambridge, MA, USA: MIT Press, 2006.

[51] Q. Candela and C. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *J. Mach. Learn. Reason.*, vol. 6, pp. 1935–1959, 2005.

[52] M. Titsias, “Variational learning of inducing variables in sparse Gaussian processes,” in *Proc. Int. Conf. Airtif. Intell. Stat. Learn.*, 2009, pp. 567–574.

[53] C. Huang and A. Darwiche, “Inference in belief networks: A procedural guide,” *Int. J. Approx. Reason.*, vol. 15, pp. 225–263, 1994.

[54] W. V. Leekwijck and E. E. Kerre, “Defuzzification: Criteria and classification,” *Fuzzy Sets Syst.*, vol. 108, no. 2, pp. 159–178, 1999.

[55] J. Bohg, K. Welke, B. Leon, M. Do, D. Song, W. Wohlkinger, M. Madry, A. Aldoma, M. Przybylski, T. Asfour, H. Marti, and D. Kragic, “Task-based grasp adaptation on a humanoid robot,” in *Proc. Symp. Robot Control*, 2012, pp. 779–786.

[56] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, “Armar-III: An integrated humanoid platform for sensory-motor control,” in *Proc. Int. Conf. Hum. Robots*, 2006, pp. 169–175.

[57] C. Parlitz, “Hardware for industrial gripping at schunk GmbH & CoKG,” *Grasping Robot., Mechanisms Mach. Sci.*, vol. 10, pp. 363–384, 2013.

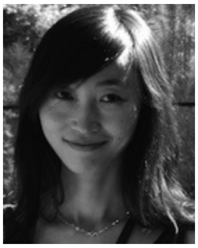
[58] C. F. Aliferis, I. Tsamardinos, and A. Statnikov, “HITON: A novel Markov blanket algorithm for optimal variable selection,” in *Proc. Amer. Med. Informat. Assoc. Annu. Symp.*, 2003, pp. 21–25.

[59] H. Hamer, J. Gall, T. Weise, and L. V. Gool, “An object-dependent hand pose prior from sparse training data,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 671–678.

[60] I. Oikonomidis, N. Kyriazis, and A. Argyros, “Efficient model-based 3D tracking of hand articulations using kinect,” in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 101-1–101-11.

[61] J. Felip and A. Morales, “Robust sensor-based grasp primitive for a three-finger robot hand,” in *Proc. Int. Conf. Intell. Robots Syst.*, 2009, pp. 1811–1816.

[62] Y. Bekiroglu, D. Song, L. Wang, and D. Kragic, “A probabilistic framework for task-oriented grasp stability assessment,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 3040–3047.



Dan Song received the B.S. degree in textile engineering from Donghua University, Shanghai, China, in 2000 and the Ph.D. degree in biomedical engineering from University of Southern California, Los Angeles, CA, USA, in 2008.

From 2008 to 2012 she was a Senior Researcher with the School of Computer Science and Communication, KTH—Royal Institute of Technology, Stockholm, Sweden. Since November 2012 she has been a Computer Vision Scientist for sports and broadcasting with ChyronHego Tracab, Stockholm.

Carl Henrik Ek received the M.Sc. degree in vehicle engineering from KTH—Royal Institute of Technology, Stockholm, Sweden, in 2005 and the Ph.D. degree in computer science from Oxford Brookes University, Oxford, U.K., in 2009.

He is an Assistant Professor with the School of Computer Science and Communication, KTH—Royal Institute of Technology. His research interest includes machine learning with applications to robotics and computer vision; specifically, he is interested in developing probabilistic models to learn efficient representations of data.

Kai Huebner (M’11) received the M.S. degree in computer science from University of Bielefeld, Bielefeld, Germany, in 2001 and the Ph.D. degree in computer science from University of Bremen, Bremen, Germany, in 2006.

From 2007 to 2010 he was a Senior Researcher with the School of Computer Science and Communication, KTH—Royal Institute of Technology, Stockholm, Sweden. Since January 2011 he has been a Senior Algorithm Developer in computer vision for eye-tracking applications with Tobii Technology AB, Stockholm.

Danica Kragic (SM’12) received the M.Sc. degree in mechanical engineering from TU Rijeka, Rijeka, Croatia, in 1995 and the Ph.D. in computer science from the KTH—Royal Institute of Technology, Stockholm, Sweden, in 2001.

She is a Professor with the School of Computer Science and Communication, KTH—Royal Institute of Technology. Her research interests include computer vision and robotics.

Dr. Kragic is a Member of the Swedish Royal Academy of Sciences and Swedish Young Academy. She received the 2007 IEEE Robotics and Automation Society Early Academic Career Award.