Mirroring Human Manipulation Actions to Robots with Functional Equivalence

Xiaofeng Gao Hangxin Liu Shengming Zhang

Abstract

We propose to investigate a force-based, task-oriented mirroring technique for robot Learning from Demonstration (LfD). Instead of mimicking demonstrator's trajectory or matching their key-points poses, this technique accounts for the latent forces and the changes of object physical states to facilitate LfD. Given a data set of hand pose and force collected by a tactile glove for the task of opening a medicine bottle with safety lock, we plan to use reinforcement learning to acquire a policy that associates the change of physical states and the exerted forces. The learned associations would provide a robot the task planner of how to change the physical states for task completion. We would apply a physics-based simulation engine to emulate various robot motion primitives and mirrors primitives that are functionally equivalent to the human actions in terms of the similar forces and state changes. In the experiment, we want to demonstrate the proposed framework by successfully teaching a Baxter robot to open the medicine bottle.

1 Introduction

A human can quickly learn new skills by observing other individuals, thus expanding their repertoire swiftly to adapt to the everchanging environment. To emulate the similar learning process, the robotics community has been developing the framework of Learning from Demonstration (LfD).

However, most of the prior work within the LfD framework is hindered by their approaches to address the "correspondence problem", i.e. the differences of embodiments between human and robots. These approaches can be roughly categorized into two types: i) keypoint-based methods that match the extracted key-points between the demonstrator and the robot, and ii) trajectory-based methods that match the trajectory of the human hand/body from the demonstration to the robot's end-effector's motion space. We argue these two types of approaches fail to capture a deeper understanding of the physical world (forces, in particular) that a robot would interact with, making them incompetent to model beyond visually observable space.

Take for example opening a medicine bottle with a child-safety lock. By design, opening such bottles often requires to push or squeeze various parts, but it presents visually similar steps compared to opening a conventional bottle without a lock. If one only imitates the trajectory or the keypoint of the demonstrator, the learned model deployed on a robot would never be able to open such a bottle.

To address this issue, we propose to use force-based, task-oriented mirroring (mirroring) approaches that integrate both actions and manipulated objects instead of the key-points based matching or trajectory based approaches.

2 Related Work

Mirror Neurons or the mirror neuron system (MNS) has been found to play an important role in human action recognition, action understanding, and action imitations [1, 2]. Several computational models have been proposed, primarily studying grasping actions and validated by hand stimuli.

Using a simple neural network with one hidden layer, the MNS model [3] was designed to associate activities in canonical neurons (encode grasping in the premotor area) with visual inputs (encode an actual grasping action). Bonaiuto *et al.* [4] introduced the MNS2-I model that uses a Recurrent Neural Network to classify grasps based on the time series of hand-object relations. In these cases, the studies were strictly confined to hand relative position to an object and little embodiment difference was presented. MNS2-II model [5] is another extension that takes audiovisual information into account. In a parallel work, Ito *et al.* [6] used a Recurrent Neural Network with Parametric Biases (PB) to model MNS.

Although it went further than hand-object relation and allowed mirroring the body movements, it is still restricted to similar embodiments. In contrast, the proposed method can handle largely different embodiments for complex tasks by augmenting forces in addition to the observed action sequences.

Correspondence Mapping in LfD is usually addressed using two approaches: i) keypoint-based methods [7, 8], in which the mapping of key-points between the demonstrator and the robot is manually defined, and ii) trajectory-based methods, in which the robot's end-effector's trajectory is either directly mapped to demonstrator's trajectory [9, 10] or indirectly mapped using trajectory optimization methods [11].

These prior work only considered the visual information, and thus would fail in the aforementioned example of opening medicine bottle, which requires the reasoning of contact forces. Although some other work using kinesthetic teaching methods [12] was capable of incorporating forces into the demonstrations, it is difficult to transfer the demonstration to a different embodiment, especially for complex tasks.

Policy Search Methods use a wide range of human demonstrations as the initial policy to constrain the search space. Reinforcement Learning scheme is usually applied to derive a control policy for a robot, circumventing the correspondence problem. These methods have succeeded in robot's constrained reaching [13], locomotion [14], grasping [15], and controlling soft robot hand [16]. To avoid being confined by human demonstration, Levine *et al.* uses guided policy search [17, 18] for robot manipulations. These works, however, have not yet demonstrated successful applications in complex tasks with multiple steps of actions.

Inverse Reinforcement Learning (IRL) or inverse optimal control [19, 20, 21, 22] has gained increasing interests in robotics community. Although it alleviated the need for reward engineering by inferring the reward/objective function from demonstrations, IRL has not been shown to scale to the same complexity of tasks as direct imitation learning, since there may exist many optimal policies that can explain a set of given demonstrations [23]. This challenge is often magnified by task complexity, making it computationally highly expensive [24].

3 Representation

We represent the action sequence to execute a task by a structural grammar model called *Temporal And-Or Graph (T-AOG)* [25] (see Figure 1). An AOG is a directed graph which describes a stochastic context-free grammar (SCFG), encoding both a hierarchical and a compositional representation. Formally, an AOG is defined as a five-tuple $G = (S, V, R, P, \Sigma)$. Specifically,

- S is a start symbol that represents an event category (*e.g.* , opening a bottle).
- V is a set of nodes including non-terminal nodes V^{NT} and terminal nodes V^{T}: $V = V^{NT} \cup V^{T}$. The **non-terminal** nodes can be divided into And-nodes and Or-nodes: $V^{NT} = V^{AND} \cup V^{OR}$. An **And-node** represents the compositional relations: a node v is an And-node if the entity represented by v can be decomposed into multiple parts represented by its child nodes. An **Or-node** indicates the alternative configuration among its child nodes: a node v is an Or-node if the entity represented by v has multiple mutually exclusive configurations represented by its child nodes. The **terminal** nodes V^{T} are the entities that cannot be further decomposed or do not have different configurations. For a T-AOG, the terminal nodes represent the *hand-object interaction (hoi)* units [26]. An hoi unit encodes motion primitives a_i that an agent can perform (*e.g.* grasping, twisting), and how the force F_i produced by such a primitive causes the changes of physical states on the object.
- $R = \{r : \alpha \to \beta\}$ is a set of production rules that represent the top-down sampling process from a parent node α to its child nodes β.
- $P : p(r) = p(\beta|\alpha)$ is the probability associated with each production rule.

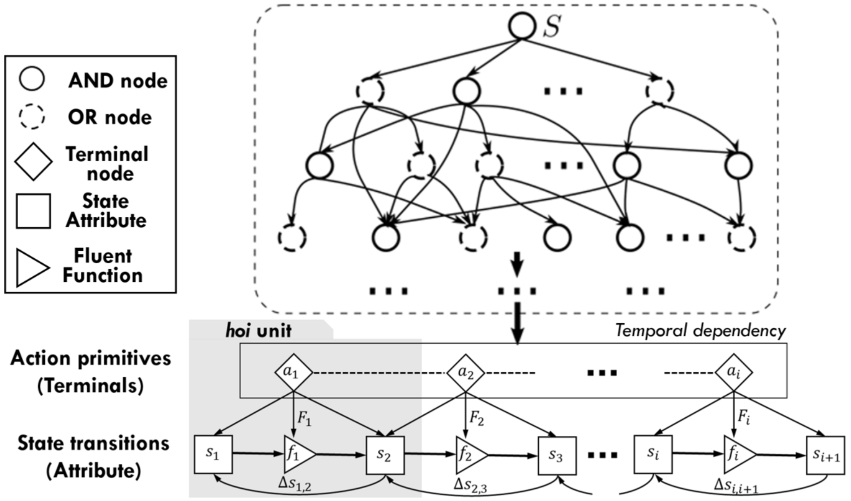


Figure 1: Illustration of a T-AOG. The T-AOG is a temporal grammar in which the terminal nodes are the *hoi* units. An hoi unit (shown in grey area) contains single motion primitive a_i that transits the state from the pre-condition s_i to post-condition s_{i+1}. The fluents function f_i represents the changes of physical state on object s_i caused by the action a_i with forces exerted F_i: $s_{i+1} = f_i(s_i, a_i; F_i)$.

- Σ is the language defined by the grammar, *i.e.* , the set of all valid sentences that can be generated by the grammar.

A **parse tree** is an instance of AOG, where one of the child nodes is selected for each Or-node. A temporal parse tree of an event is a sub-graph of the T-AOG that captures the temporal structure of the scenario. The terminal nodes of the parse tree form a valid sentence; in this case, terminal nodes are a set of *hoi* units consisting of the actions for an agent to execute in a fixed order and the fluents changes after executing such an action sequence.

4 Force-based Task-oriented Mirroring

4.1 Learning Force-state Associations as *hoi* Units

One needs to know the effect of a particular type of forces so that the desired action can be planned. This leads to the necessity of investigating the state changes caused by forces. We cast this problem into a reinforcement learning framework to learn a policy that associates them. The state space and the action (force) space from human demonstrations are discretized and quantized, and an iterative Q-Learning scheme is applied. We believe the proposed learning does not loss generality since one can scale up the process to continuous state space or action space by using DQN [27] and other advanced Q-Learning methods.

Categorize Force The pose and force data of human demonstrations were collected using a tactile glove. The forces exerted between a human hand and the object, together with the poses, are projected onto the mesh of the bottle as the force vector. The resulting force vector F_t^o is a 4-dimensional vector, where the first 3 dimensions represent the position of the object's surface vertices and 4th dimension as the force magnitude on each vertex.

A k-means clustering [28] is adopted to categorize the force vector F_i^o into N types. After assigning labels to each frame, we aggregate the segments with the same label and take the average (Figure 2(a)).

Quantize State The states of a rigid target object under manipulation actions can be described by the relative poses among object's parts, *e.g.* bottle and lid, multiple Lego blocks, *etc.*. Without loss of generality, we use the relative distance and relative rotation angle between the lid and the bottle, which are derived from their relative poses, as our state space. As shown in Figure 2(c), within

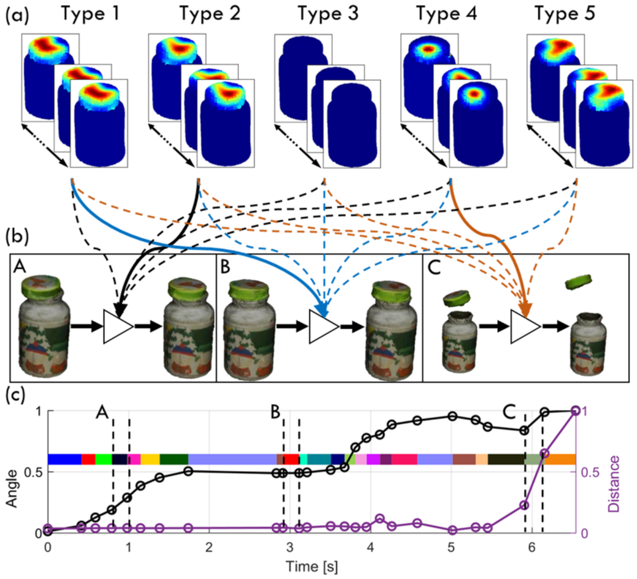


Figure 2: Force and state associations as *hoi* units. The manipulation force is clustered into 21 types. (a) 5 types of force vector, in which Type 3 has no force. (c) Given the categorized force and quantized states based on the forces, (b) the Q-learning algorithm associates a force vector to a specific state change (A: lid is twisted; B: initiate contact; C: pull off the lid) shown by the solid lines.

each segment of the force vector (shown in color bars), we take the average of the state (angle and distance) and normalize to unit length.

$$s_i = \langle d_i, \theta_i \rangle \subset [0,1]^2, \forall i \in \{1, \dots, M\} \tag{1}$$

where M is the total number of states, d_i and θ_i denote the relative distance and angle, respectively.

Associate Force and State as *hoi* Units by Q-Learning By replacing the actions in Q-learning with the labels of the force vector l_k, we adopt the tabular Q-Learning method to learn a policy that associates the current state s_i to a force type using the iterative Q-Learning update rule in a temporal difference fashion,

$$Q(s_i, l_k) = (1-\alpha) \cdot Q(s_i, l_k) + \alpha \cdot \left[r(s_i, l_k) + \gamma \cdot \max_k Q(s_{i+1}, l_k) \right], \tag{2}$$

where r denotes the reward, Q the Q-function, α the learning rate, and γ the discount factor. Here, we assume the system dynamics is deterministic.

Inference We pick the best action according to the Q-function

$$l_* = \operatorname*{argmax}_k Q(s_i, l_k). \tag{3}$$

The association among s_i, s_{i+1} and corresponding F_k naturally forms an *hoi* unit (Figure 1), which will be used for learning task-oriented grammar discussed in the next section.

4.2 Learning Task-Oriented Grammar

Grammar Induction Each successful demonstration contributes a sequence of *hoi* units that encode the types of forces and the state evolvement. We induce a T-AOG $\mathcal{G}$ from multiple demonstrations using a modified version of Automatic Distillation of Structure (ADIOS) algorithm presented in [29].

Parsing To generate a valid sentence from the learned T-AOG $\mathcal{G}$, we sample $\mathcal{G}$ by decomposing all the And-nodes and selecting one branch at each Or-node, resulting in a parse tree $pt = (hoi_0, \dots, hoi_K)$. This pt is *task-oriented* in the sense that its terminal nodes $hoi_k \in pt$ are invariant across embodiments for the given task.

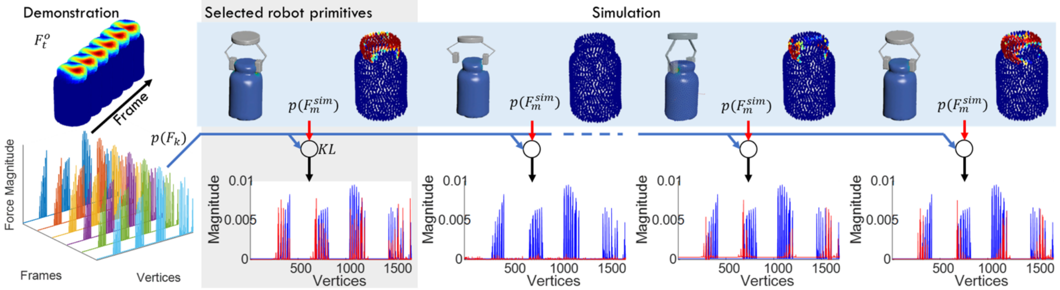


Figure 3: From the demonstration, each frame of the force in the same cluster l_k produces a force distribution F_t^o, and the average of them is the distribution of the force category F_k. Among the simulated force responses F_m^{sim} obtained from a physics-based simulator, the corresponding primitive of the most similar force, measured by KL divergence, is selected for robot execution. The first figure in the first row displays the forces in the same cluster, and the other four are the simulated robot primitives (downward, no contact, contact, and twist) and their force responses. The first figure in the second row shows the force distributions of the same cluster in each frame, whereas the other four are the distributions of F_k against each simulated force distribution F_m^{sim}, denoted by blue and red, respectively.

4.3 Mirroring to Robot

Simulation-based Motion Synthesis We defined a dictionary of discrete robot motion primitives $\Omega_{a^r} = \{a_1^r, \dots, a_M^r\}, M = 10$. These motion primitives are parameterized by the change of end-effector poses, including moves in all 6 canonical directions, rotations in both clockwise and counter-clockwise directions, and open/close the gripper. The task of opening a medicine bottle can be accomplished by a combinatorics of the primitives. Given a pt, we seek to generate a sequence of robot actions $\{a_i^r, i = 1, \dots, m\}$ that produce forces sufficient to cause the same changes of states as encoded in the sampled pt. In this sense, we say the robot action a_i^r is *functionally equivalent* to the human actions a_i^h in the demonstrations.

Without using an actual robot to run hundreds of manipulation actions on various bottles, a physics-based simulator (see Figure 3) is introduced to investigate the force exerted by the robot gripper on the bottle by executing the set of motion primitives. The detail of the simulation is described in Sec. 5.2. We denote the force vector (including the 3D pose and the magnitude of the force) obtained from the simulator as F_m^{sim}, where m is the index of robot primitives, and compare it to the corresponding F_k, the average force vector exerted by human demonstrations with label l_k.

The similarity of the two forces can be measured by Kullback–Leibler (KL) divergence, and the robot action is selected by

$$F^{sim*} = \operatorname*{argmin}_m \mathrm{KL}\left(P(F_k) \parallel P(F_m^{sim})\right) = \operatorname*{argmin}_m \sum_v \left[P_{F_k}(v) \log \frac{P_{F_k}(v)}{P_{F_m^{sim}}(v)} \right], \tag{4}$$

where v is the vertex index on the object mesh. Once F^{sim*} is selected, the robot would choose the corresponding primitive a^{r*} that produces F^{sim*}.

5 Experiment

5.1 Preliminary

Dataset The hand pose and force data is collected using a tactile glove that is equipped with i) a network of 15 IMUs to measure the rotations between individual phalanxes, and ii) 6 customized force sensors using Velostat, a piezoresistive material, to record the force in two regions (proximal and distal) on each phalange and a 4×4 regions on palm. The relative poses between the wrist of hand and object parts (*i.e.* bottle and lid) are obtained from Vicon motion capture system. The data is collected, processed, and visualized using the Robot Operating System (ROS).

Robot Platform We exercise the proposed framework in a robot platform with a dual-armed 7-DoF Baxter robot from Rethink Robotics mounted on a Data Speed mobility base. The robot is equipped with a ReFlex TakkTile gripper on the right wrist and a Robotiq S85 parallel gripper on the left. The entire system runs on ROS, and the arm motion planning is computed using *MoveIt!*.

5.2 Physics-based Simulation

The physics-based simulation needs to be able to capture intricate frictional contact between the robot gripper and the bottle. The total force applied at each point located at the surface of the bottle consists of several terms: the normal component of squeezing force from the gripper, the tangential component of static friction force from the gripper, the internal elastic force from the rest of the continuous bottle material, and gravity.

The key to achieving such force balance in the simulator is to model the deformation of the bottle. Various physical constitutive models and stress-strain relationships exist for polymers, and it is impractical for us to find the exact material parameters through mechanical tension or compression tests. Thus, we assume the deformation of the bottle is sufficiently far away from the plastic regime, and adopt a standard hyperelastic model: the Neo-Hookean model [30] for describing the mechanical stress under deformation

$$\mathbf{P} = \mu(\mathbf{F} - \mathbf{F}^{-T}) + \lambda \log(\det(\mathbf{F}))\mathbf{F}^{-T}, \tag{5}$$

where $\mathbf{F}$ is the deformation gradient tensor encoding the strain at each point, $\mathbf{P}$ is the first Piola-Kirchoff stress tensor describing its elastic mechanical stress, and μ, λ are material parameters describing the stiffness and incompressibility of the bottle. Based on this physical model, the governing equation describing the force balance of the bottle is given by

$$\nabla \cdot \mathbf{P} = \mathbf{f}^{ext}, \tag{6}$$

where $\mathbf{f}^{ext}$ denotes the total external force on the bottle.

We solve Equation (6) using the Finite Element Method [31]. The input bottle geometry is first converted from a triangulated surface to a tetrahedralized volume using TetGen [32]. The robot gripper mesh is converted into a watertight level set represented with the OpenVDB [33], which allows natural treatment of frictional contact under arbitrary kinematic rigid motion. The additional parameters including friction coefficient, μ, and λ are set empirically. Once the discretized equation system is solved to convergence, we evaluate the force magnitude at each discrete point of the container surface mesh and store them in F_m^{sim}.

5.3 Learning

Figure 4 illustrates the Q-learning setting. In particular, we set the success state to have a reward of $+1$, the failure state to have a reward of -1, and all other states to have 0 reward. The discount factor is set to be 0.99 and we use ϵ-greedy exploration with exponential decay. In Figure 4(a), the red curve shows the cumulative reward during each training episode and the blue curve shows the average cumulative reward during evaluation. The evaluation is performed every 10 episodes during training with a policy induced from the Q-table. As shown in the figure, during the training, the cumulative reward generally increases until after finding a path that leads to the maximum reward and begins fluctuating. The fluctuation happens because, even though an optimum path has been found in ϵ-greedy exploration policy, there is still a marginal probability of a non-optimal action being chosen at each step. During the evaluation, the reward monotonically increases slowly at first and jumps to the maximum. This is due to the optimal path found during training and the learning signals propagated into the Q-table. The policy induced from the Q-table converges to the optimum after approximate 900 episodes of training.

5.4 Robot Execution with Functional Equivalence

After a T-AOG is induced from the learned policy, we can sample a *pt* that encodes a list of force types the robot should imitate in order to cause the same changes of object states. Figure 5(a) shows the sequence of selected robot primitives based on the criteria of KL divergence. In this instance, there are 21 primitives to be accomplished in order to complete the task. The force responses of the 10 robot primitives are obtained and their similarities to the corresponding F_k are measured in each

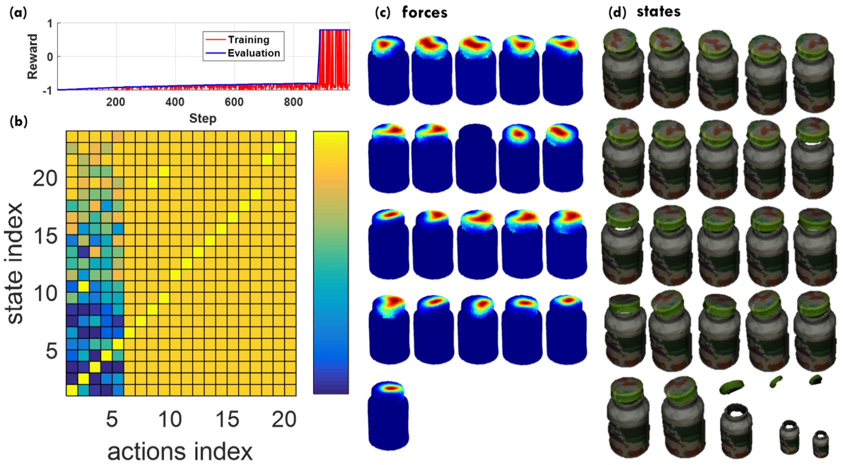
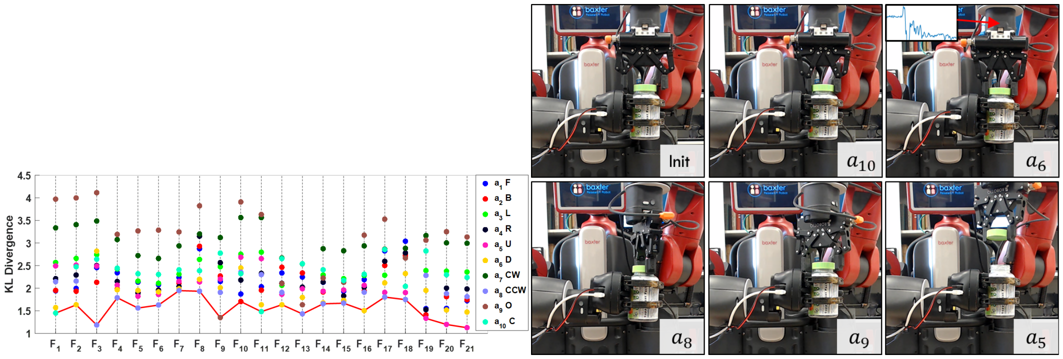


Figure 4: (a) Cumulative rewards during training and evaluation. The fluctuation of reward in training is due to the non-optimal action selection under the ϵ-greedy exploration policy. (b) The landscape of the learned Q table, where yellow indicates high values and blue indicates low values. (c) The 21 types of actions (forces) by clustering in one exemplary demonstration and (d) the 25 discretized states based on the forces (some force types appear more than once).



(a)

(b)

Figure 5: (a) The KL divergence for all actions primitives in one *pt*. The primitives are a_1 *move forward*, a_2 *move backward*, a_3 *move left*, a_4 *move right*, a_5 *move up*, a_6 *move down*, a_7 *rotate clockwise*, a_8 *rotate counter-clockwise*, a_9 *open gripper*, and a_{10} *close gripper*. The solid red line is the sequence of actions for a robot to execute. (b) Robot execution. Starting from the initial pose, the primitives (in grey area) are performed sequentially. For a_6 (*downward*), the force sensed in robot's wrist clearly indicates the bottle lid is being pushed. The robot opens the medicine bottle by a_5 (*upward*).

stage. The primitives with the lowest KL divergence (connected by the red line) are selected for robot execution. Specifically, the primitive *opening the gripper* has the largest divergence in most of the cases as it produces no forces to the object, except in F_9 when the demonstrator released the lid after one rotation. The pressing force that is critical to our task is also captured and mirrored to robot well (see F_2 and F_{16} where a *downward* primitive is planned). The gripper's counter-clockwise rotation is limited by the joint configuration from F_{13} to F_{18}. Our algorithm captures it and re-positions the gripper before F_{11}. Finally, *upward* primitives are selected to finish the task by pulling the lid.

The robot can execute the planned primitives and the execution process is shown in Figure 5(b). It starts from an initial position and performs the corresponding primitives sequentially. The grey area in the lower right corner shows the current primitive. The a_6 *downward* primitive indeed generates forces which are captured by the force sensor in the robot wrist. The result of successfully opening the medicine bottle validates the proposed framework.

6 Conclusion

This paper presents a *force-based, task-oriented mirroring* framework that enables a robot to learn a manipulation task by imitating the forces a demonstrator applies as well as the changes of the object physical states caused by the forces. This differs from prior work that either mimics the demonstrator's trajectory or matches the key-points, providing a deeper understanding of the physical world that a robot interacts with.

In the experiment, we use a pose and force sensing tactile glove to collect human demonstrations of opening a medicine bottle with a safety lock. The discretized force types and object state changes are associated by learning a policy using Q-Learning algorithm. Based on the learned associations, a T-AOG is induced to provide a robot the task planner on how to change the physical states by exerting a certain type of force. Our physics-based simulation engine is capable of emulating the forces produced by a set of robot motion primitives. Human demonstration actions are successfully "mirrored" to robot's actions with *functional equivalence* as they both produce similar forces and cause similar changes in the object states. The proposed framework is validated by an actual Baxter robot opening medicine bottles.

References

[1] Giacomo Rizzolatti and Laila Craighero. The mirror-neuron system. *Annual Reviews Neuroscience*, 27:169–192, 2004.

[2] Serge Thill, Daniele Caligiore, Anna M Borghi, Tom Ziemke, and Gianluca Baldassarre. Theories and computational models of affordance and mirror systems: an integrative review. *Neuroscience & Biobehavioral Reviews*, 37(3):491–521, 2013.

[3] Erhan Oztop and Michael A Arbib. Schema design and implementation of the grasp-related mirror neuron system. *Biological Cybernetics*, 87(2):116–140, 2002.

[4] James Bonaiuto, Edina Rosta, and Michael Arbib. Extending the mirror neuron system model, i. *Biological Cybernetics*, 96(1):9–38, 2007.

[5] James Bonaiuto and Michael A Arbib. Extending the mirror neuron system model, ii: what did i just do? a new role for mirror neurons. *Biological cybernetics*, 102(4):341–359, 2010.

[6] Masato Ito and Jun Tani. On-line imitative interaction with a humanoid robot using a dynamic neural network model of a mirror system. *Adaptive Behavior*, 12(2):93–115, 2004.

[7] Jonas Koenemann, Felix Burget, and Maren Bennewitz. Real-time imitation of human whole-body motions by humanoids. In *International Conference on Robotics and Automation (ICRA)*, 2014.

[8] Tianmin Shu, Xiaofeng Gao, Michael S Ryoo, and Song-Chun Zhu. Learning social affordance grammar from videos: Transferring human interactions to human-robot interactions. In *International Conference on Robotics and Automation (ICRA)*, 2017.

[9] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal. Learning and generalization of motor skills by learning from demonstration. In *International Conference on Robotics and Automation (ICRA)*, 2009.

[10] Yezhou Yang, Yi Li, Cornelia Fermüller, and Yiannis Aloimonos. Robot learning manipulation action plans by" watching" unconstrained videos from the world wide web. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[11] Guilherme Maeda, Marco Ewerton, Dorothea Koert, and Jan Peters. Acquiring and generalizing the embodiment mapping from human observations to robot skills. *IEEE Robotics and Automation Letters*, 1(2):784–791, 2016.

[12] Simon Manschitz, Michael Gienger, Jens Kober, and Jan Peters. Probabilistic decomposition of sequential force interaction tasks into movement primitives. In *International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[13] Florent Guenter, Micha Hersch, Sylvain Calinon, and Aude Billard. Reinforcement learning for imitating constrained reaching movements. *Advanced Robotics*, 21(13):1521–1544, 2007.

[14] Evangelos Theodorou, Jonas Buchli, and Stefan Schaal. Reinforcement learning of motor skills in high dimensions: A path integral approach. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2010.

[15] Urbain Prieur, Véronique Perdereau, and Alexandre Bernardino. Modeling and planning high-level in-hand manipulation actions from human knowledge and active learning from demonstration. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2012.

[16] Abhishek Gupta, Clemens Eppner, Sergey Levine, and Pieter Abbeel. Learning dexterous manipulation for a soft robotic hand from human demonstrations. In *International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016.

[17] Sergey Levine and Pieter Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[18] Sergey Levine, Nolan Wagener, and Pieter Abbeel. Learning contact-rich manipulation skills with guided policy search. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.

[19] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2000.

[20] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2004.

[21] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[22] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*. Chicago, IL, USA, 2008.

[23] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning (ICML)*, 1999.

[24] James MacGlashan and Michael L Littman. Between imitation and intention learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[25] Song-Chun Zhu and David Mumford. A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision*, 2(4):259–362, 2007.

[26] Scott H Johnson-Frey, Farah R Maloof, Roger Newman-Norlund, Chloe Farrer, Souheil Inati, and Scott T Grafton. Actions or hand-object interactions? human inferior frontal cortex and action observation. *Neuron*, 39(6):1053–1058, 2003.

[27] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[28] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892, 2002.

[29] Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. Predicting human activities using stochastic grammar. In *International Conference on Computer Vision (ICCV)*. IEEE, 2017.

[30] Christopher W Macosko. *Rheology: principles, measurements, and applications*. Wiley-vch, 1994.

[31] Javier Bonet and Richard D Wood. *Nonlinear continuum mechanics for finite element analysis*. Cambridge university press, 1997.

[32] Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 41(2):11, 2015.

[33] Ken Museth, Jeff Lait, John Johanson, Jeff Budsberg, Ron Henderson, Mihai Alden, Peter Cucka, David Hill, and Andrew Pearce. Openvdb: an open-source data structure and toolkit for high-resolution volumes. In *ACM SIGGRAPH 2013 courses*, page 19. ACM, 2013.