Learning Hybrid Object Kinematics for Efficient Hierarchical Planning Under Uncertainty

Ajinkya Jain
Department of Mechanical Engineering
University of Texas at Austin, USA
ajinkya@utexas.edu

Scott Niekum
Department of Computer Science
University of Texas at Austin, USA
sniekum@cs.utexas.edu

Abstract: Sudden changes in the dynamics of robotic tasks, such as contact with an object or the latching of a door, are often viewed as inconvenient discontinuities that make manipulation difficult. However, when these transitions are well-understood, they can be leveraged to reduce uncertainty or aid manipulation—for example, wiggling a screw to determine if it is fully inserted or not. Current model-free reinforcement learning approaches require large amounts of data to learn to leverage such dynamics, scale poorly as problem complexity grows, and do not transfer well to significantly different problems. By contrast, hierarchical planning-based methods scale well via plan decomposition and work well on a wide variety of problems, but often rely on precise hand-specified models and task decompositions. To combine the advantages of these opposing paradigms, we propose a new method, Act-CHAMP, which (1) learns hybrid kinematics models of objects from unsegmented data, (2) leverages actions, in addition to states, to outperform a state-of-the-art observation-only inference method, and (3) does so in a manner that is compatible with efficient, hierarchical POMDP planning. Beyond simply coping with challenging dynamics, we show that our end-to-end system leverages the learned kinematics to reduce uncertainty, plan efficiently, and use objects in novel ways not encountered during training.

1 Introduction

Robots working in human environments need to perform dexterous manipulation on a wide variety of objects. Such tasks typically involve making or breaking contacts with other objects, leading to sudden discontinuities in the task dynamics. Furthermore, many objects exhibit configuration-dependent dynamics, such as a refrigerator door that stays closed magnetically. While the presence of such nonlinearities in task dynamics can make it challenging to represent good manipulation policies and models, if well-understood, these nonlinearities can also be leveraged to improve task performance and reduce uncertainty. For example, when inserting a screw into the underside of a table, if direct visual feedback is not available, indirect feedback from wiggling the screw (a semi-rigid connection between the screw and the table) can be leveraged to ascertain whether the screw is inserted or not. In other words, the sensed change in dynamics (from free-body motion to rigid contact) serves as a landmark, partially informing the robot about the state of the system and reducing uncertainty. Such dynamics can be naturally represented as hybrid dynamics models, in which a discrete state represents which continuous dynamics model is active at any given time.

Current model-free reinforcement learning approaches [1–4] can learn to cope with hybrid dynamics implicitly, but require large amounts of data to do so, scale poorly as the problem complexity grows, face representational issues near discontinuities, and do not transfer well to significantly different problems. Conversely, hierarchical planning-based methods [5–8] can represent and reason about hybrid dynamics directly, scale well via plan decomposition, and work well on a wide variety of problems, but typically rely on precise hand-specified models and task decompositions. We propose a new algorithm, Action-conditional Changepoint Detection using Approximate Model Parameters (Act-CHAMP), to combine the advantages of these opposing paradigms. Act-CHAMP (1) learns hybrid motion models of complex objects from unsegmented human demonstrations, (2) reasons about actions, in addition to states, for model inference, (3) is more robust to noise in comparison to the observation-only model inference approaches, and (4) develops models that are rich enough
to be used for object manipulation tasks under uncertainty. Further, we combine Act-CHAMP with a recently proposed hierarchical POMDP planner, POMDP-HD [5], to develop an end-to-end system that takes unsegmented human demonstrations guiding robot’s interactions with an object as input and develops motion plans that can leverage learned hybrid model to perform planning under uncertainty for novel tasks involving that object.

Act-CHAMP is an action-conditional algorithm that offers numerous natural benefits over the state-of-the-art observation-only inference methods. First, Act-CHAMP can perform more robust model inference in the presence of noisy demonstrations. For example, if a demonstration predominantly contains actions that are applied orthogonally to the axis of motion of a drawer, due to the lack of observed motion, an observation-only model inference can easily misclassify the motion model to be rigid; on the other hand, an action-conditional inference can maintain an equal likelihood of observing either a rigid or a prismatic model since no movement would be expected under either model given the action. Second, due to an explicit representation of actions in the learned models, they can be directly used for motion planning. Third, an action-conditional model can be used by the robot to take informative exploratory actions for disambiguating between candidate object motion models.

To evaluate the proposed approach, we first show that Act-CHAMP can infer hybrid motion models of articulated objects with higher fidelity and less data than a state-of-the-art observation-only algorithm, CHAMP, [9]. We also consider two degenerate cases of noisy demonstrations consisting of predominantly actions applied orthogonal to the axis of motion to demonstrate its robustness to noise. Next, in a full end-to-end test, we learn hybrid kinematics models for a microwave and a drawer from human demonstrations and successfully perform manipulation tasks of opening and closing both the objects in new situations with high success rate, showing the planner’s ability to generalize. Finally, we show that the planner can leverage learned models creatively to complete a novel task efficiently—our method learns a kinematics model of a stapler and uses it to dexterously place the stapler at the target point, reachable only through a narrow corridor in the task configuration space.

2 Related Works

Learning Articulation Motion Models from Observations: A wide variety of approaches have been studied in the literature to learn kinematic models for articulated objects directly from visual data [9–12]. Sturm et al. [10, 13] proposed a probabilistic framework to learn motion models of articulation bodies from human demonstrations. However, the framework assumes that the objects are governed by a single articulation model, which may not hold true for all the objects. For example, a stapler intrinsically changes its articulation state (e.g. rigid vs. rotational) based on the relative angle between its arms. To address this, Niekum et al. [9] proposed an online changepoint detection algorithm, CHAMP, to detect both the governing articulation model and the temporal changepoints in the articulation relationship of the objects. In this work, we combine the CHAMP algorithm with action-conditional model inference to learn hybrid kinematics models of objects exhibiting configuration-dependent kinematics.

Interactive Perception: Interactive perception approaches aim at leveraging the robot’s actions to better perceive objects and build accurate kinematic models [14–16]. Katz et al. first used this approach to learn articulation motion models for planar objects [14], and later extended it to use RGB-D data to learn 3D kinematics of articulated objects [15]. Though these approaches of interactive perception use robot’s actions to generate perceptual signals for model estimation, they require the robot’s interaction behavior to be pre-scripted by an expert. Also, these approaches do not explicitly reason about actions while performing model inference as Act-CHAMP does.

Learning Dynamical Models from Pixels: Learning object kinematic/dynamics models directly from raw visual data is a promising direction for learning object motion models. The Embed to Control (E2C) method proposed by Watter et al. [17] uses a novel deep probabilistic generative model to convert raw image pixels into a low-dimensional latent space, in which stochastic optimal control can be applied. Byravan et al. developed SE3-nets [18] and SE-3Pose-Nets [19] to learn predictive dynamics models of object motion in a scene from input point-cloud data and applied action vectors which can be used to directly perform robot visuomotor control from input point cloud data. While deep neural network-based approaches have shown much potential, the biggest hurdle in using such approaches to a wide variety of real-world robotics tasks is the need for a vast amount of training data, which is often not readily available. Also, these approaches tend to transfer
poorly to new tasks. In this work, we combine model learning with generalizable planning under uncertainty to address these challenges, though deep learning methods may be useful in future work, in place of our simplified perception system.

Other Object Motion Modeling Frameworks: Articulation motion models can also be seen as geometric constraints imposed on two or more rigid bodies. Perez et al. [20] have proposed a method, C-LEARN, to learn geometric constraints encountered in a manipulation task from non-expert human demonstrations. Subramani et al. [21, 22] developed an approach to learn geometric constraints governing relative motion between objects from human demonstrations. Their proposed approach can successfully learn geometric constraints even from noisy demonstrations. However, the use of custom force-sensitive hand-held tools to record human demonstrations restricts the generalizability of the approach to a wider set of tasks.

3 Preliminaries

Changepoint Detection: Many kinematic relationships between objects are not well described by a single smooth model. For example, in most configurations, a microwave door is a revolute joint with respect to the microwave; however, due to the presence of a latch, this relationship changes to a rigid one when the door is closed. We can define a temporal changepoint in the time series of the observed motion of such an object to mark the transition point between the two governing motion models. The CHAMP algorithm [9, 23] computes the joint probability of observing a changepoint at time \( t \) and an event \( E_t \), denoting that given a changepoint at time \( j \), the MAP choice of changepoints have occurred prior to the time \( j \), in a given time series of observations \( y_{1:T} \) with the segment from \( j \) to \( t \) being governed by the model \( M \)

\[
\begin{align*}
P_t &= p(C_t = j, M, E_t, y_{1:t}) \\
P_t^{MAP} &= p(\text{Changepoint at } t, E_t, y_{1:t})
\end{align*}
\]

which results in

\[
\begin{align*}
P_t(j, M) &= (1 - G(t - j - 1)) \cdot L(j, t, M) \cdot p(M) \cdot P_j^{MAP} \\
P_t^{MAP} &= \max_{j, M} \left[ \frac{g(t - j)}{1 - G(t - j - 1)} \cdot P_t(j, M) \right]
\end{align*}
\]

where \( L(j, t, M) \) is the model evidence for the segment between time \( j \), and \( p(M) \) is the probability of the governing model being \( M \). Assuming that the data after a changepoint is independent of the data prior to that changepoint, the position of changepoints in the time series can be modeled as a Markov chain in which the transition probabilities are defined by the time since the last changepoint,

\[
p(\tau_{i+1} = t | \tau_i = j) = g(t - j)
\]

By searching for the values of \((j, q)\) that maximize \( P_t^{MAP} \), we can recover the Viterbi path for any point. We can repeat this process until the time \( t = 0 \) is reached to estimate all the changepoints that occurred in the given time series \( y_{1:T} \).

Hybrid Dynamics: In a hybrid dynamics model the states of the system evolve with time over both continuous space \( x \in \mathbb{R}^N \) and a finite set of discrete states \( q \in Q \subset \mathbb{W} \) [24]. Each discrete state of the system denotes a separate dynamics model that governs the evolution of the continuous states. The discrete state transitions can be represented as a directed graph with each possible discrete state \( q \) corresponding to a node and edges \((e \in E \subset Q \times Q)\) marking possible transitions between the nodes. These transitions are conditioned on the continuous states. A transition from the discrete state \( q \) to another state \( q' \) happens if the continuous states \( x \) are in the guard set \( G(q, q') \) of the edge \( e_q \) where \( e_q = \{q, q'\}, G(\cdot) : E \rightarrow P(X) \) and \( P(X) \) is the power set of \( X \). Thus, for each discrete state \( q, \) in a hybrid dynamics model we can define \( x_{t+1} = F^q(x_t, u_t), z_t = H^q(x_t) \) where \( x \in \mathbb{R}^n, u \in \mathbb{R}^m, z \in \mathbb{R}^l, F^q(x, u) \) and \( H^q(x) \) are the continuous state, control input, observation variables, state dynamics and observation functions respectively. A finite state Markov chain can be used to model the evolution of the discrete states of the system. Defining the state transition matrix as \( \Pi = \{\pi_{i,j}\} \), the discrete state evolution can be given as \( q_{t+1} = \Pi q_t \).

4 Approach

In this section, we present the details of our proposed algorithm, Act-CHAMP. We start with the details of the action-conditional model inference that is used to detect changepoints in the given
human demonstrations in section 4.1. Then in section 4.2, we describe how detected motion models and changepoints can be used to build hybrid kinematic models for the objects. Later, in section 4.3, we show how the learned hybrid kinematics models can be used to efficiently perform hierarchical POMDP planning in novel manipulation tasks.

### 4.1 Action-Conditional Model Inference

We infer the governing motion model between two rigid objects (or object parts) from time-series observations, \( y_{1:T} \) (relative motion between the two objects), as well as the corresponding applied actions, \( a_{i-1:T-1} \), observed from human demonstrations. Following the framework proposed by Sturm et al. [10], we define the relative transform between two objects with poses \( x_i \) and \( x_j \) \(\in SE(3)\) at time \( t \) as:

\[
\Delta_{ij,t} = x_{i,t} \odot x_{j,t}^{-1}.
\]

Additionally, we define an action \( a_i \) taken by the demonstrator at a time \( t \) as the intended displacement to be applied to the relative transform between two objects from time \( t \) to \( t + 1 \) as:

\[
a_i = \Delta_{ij,t+1} \oplus \Delta_{ij,t}.
\]

For a given time series of observations \( y_{1:T} \) and the corresponding applied actions \( a_{1:T} \), we define the model evidence for a model \( M \) having parameters \( \theta \) as:

\[
p(y_{1:T} \mid M, \theta, a_{1:T}) = \prod_{i=2}^{T} p(y_i \mid M, \theta, a_{i-1}, y_{i-1})
\]

where \( p(y_i \mid M, \theta, a_{i-1}, y_{i-1}) \) is the likelihood of observing the relative transform, \( y_t \), under articulation model \( M \) when the observed relative transform at time \( t - 1 \) was \( y_{t-1} \) and an action \( a_{i-1} \) was applied. Alternatively, we can define it as:

\[
\prod_{i=2}^{T} p(y_i \mid M, \theta, a_{i-1}, y_{i-1}) = \prod_{t=2}^{T} p(y_t \mid \hat{\Delta}_{ij,t})
\]

where \( \hat{\Delta}_{ij,t} \) is the predicted relative pose under the model \( M \) at time \( t \). For a given observation \( y_{t-1} \) and a corresponding applied action \( a_{t-1} \), a predicted configuration \( \hat{\Delta}_{ij,t} \) can be calculated as:

\[
\hat{\Delta}_{ij,t} = f( f_{M, \theta}^{-1}(y_{t-1}) + [J_{M, \theta}]_{i,t}^{-1} a_{t-1} )
\]

where \( f_{M, \theta}(\cdot) \) and \( f_{M, \theta}^{-1}(\cdot) \) represent the forward and inverse kinematics functions defined for model \( M \) connecting the the observed relative pose \( \Delta_{ij,t} \) at time \( t \) with a unique configuration \( c_i \), expressed in the generalized coordinates of the model \( M \) (e.g., a position along the prismatic axis, or an angle with respect to the axis of rotation) respectively. We define Jacobian \( J_{M, \theta} \) at a time \( t \) as:

\[
\delta \Delta_{ij,t} = [J_{M, \theta}]_{i,t}(\delta c_i) \quad \text{where} \quad \delta \Delta_{ij,t} \quad \text{and} \quad \delta c_i \quad \text{represent small perturbations applied to the relative pose and the configuration, respectively.}
\]

**Action-Conditional Changepoint Detection:** In the CHAMP algorithm, the evidence for a model \( M \) being the governing model for the segment between time \( j \) and \( t \) is calculated by conditioning only on observations. However, this can easily lead to false detection of changepoints resulting in an inaccurate system model. Consider an example case of deducing the motion model for a drawer from a noisy demonstration in which the majority of applied actions are orthogonal to the axis of motion of the drawer. Due to intermittent displacements, an observation-only model inference approach can falsely detect changepoints in the system motion model. On the other hand, an action-conditional inference can maintain an equal likelihood of observing either a rigid or a prismatic model under off-axis actions, leading to more accurate system model detection. Hence, we propose to use the action-conditional motion inference defined in the last section to calculate the model evidence for changepoint estimation. Specifically, we redefine the model evidence for changepoint estimation as:

\[
L(j, t, M) = p(y_{j+1:t} \mid M, a_{j:t-1}) = \int p(y_{j+1:t} \mid M, \theta, a_{j:t-1}) p(\theta) d\theta
\]

We perform the model inference on the given time series of observations and actions using an approximate BIC-penalized likelihood function, proposed by Niekum et al. [9]. Complete definitions of forward and inverse kinematics models for rigid, revolute and prismatic models and the data-likelihood observation model are beyond the scope of this work, but further details can be found in [9, 10].

1The operators \( \oplus \) and \( \odot \) represent motion composition operations. For example, if poses \( x_i \), \( x_j \) \( \in \mathbb{R}^{4 \times 4} \) are represented as homogeneous matrices, then these operators correspond to matrix multiplications \( x_i \odot x_j = x_i x_j \) and its inverse multiplication, \( x_i \oplus x_j = x_i^{-1} x_j \), respectively.
4.2 Learning Hybrid Motion Models

We propose to model the relative motion of the objects demonstrating configuration-dependent motion models as a hybrid model consisting of multiple local models, of which only one is active at a time. We derive the transition conditions between the local models using the estimated changepoints from the Act-CHAMP algorithm. Note that a changepoint at time $t = \tau$ denoting a transition from model $M_l$ to model $M_k$ will correspond to the observed relative transform $y_t$ between the two objects in the given time series of observation $y_t$. The inverse kinematics function $f_{M_l}^{-1}$ can be used to find the configurational changepoint $\tilde{c}_{lk}$, a fixed configuration in the state space defined for model $M_l$, that marks the transition from model $M_l$ to the next model $M_k$. Using this knowledge, we can represent the two models $M_l$ and $M_k$ as two discrete states of the hybrid model and define the guard set for the edge $e_{q_{M_l}^{M_k}}$ as $G(q_{M_l}, q_{M_k}) = \{c \in \mathbb{R} | c \geq \tilde{c}_{lk}\}$. Similarly, the guard set for the edge marking the reverse transition from model $M_k$ to $M_l$, $e_{q_{M_k}^{M_l}}$ will be $G(q_{M_k}, q_{M_l}) = \{c \in \mathbb{R} | c < \tilde{c}_{lk}\}$. If a third model $M_n$ is then detected for the system which is active when $\tilde{c}_{lk} < c \leq \tilde{c}_{kn}$, it can be easily added to the hybrid model by adding new edges $e_{q_{M_l}^{M_n}}$ and $e_{q_{M_k}^{M_n}}$ with the corresponding guard sets defined along the same lines. Accordingly, the guard set $G(q_{M_l}, q_{M_n})$ will be updated as $G(q_{M_l}, q_{M_n}) = \{c \in [\tilde{c}_{lk}, \tilde{c}_{kn}]\}$. Thus, by combining together all discovered local models and their corresponding transition conditions, a complete hybrid kinematics model of the system can be learned directly from unsegmented human demonstrations. Note that while we describe the hybrid model construction for one-DoF local models here, the same approach can be extended to multiple DoF models by defining the guard sets appropriately.

The motion model for each of the local models can be constructed using the learned forward and inverse kinematics functions. Let $x_t$, $\dot{x}_t$ and $u_t$ denote the 6-DoF position and velocity of the object being manipulated by the robot (e.g. the door in the case of a microwave), and the action applied on the object at time $t$, respectively. The state transition function and the observation function are defined as:

$$\begin{align*}
\dot{x}_t &= f_{M,\theta^*}(x_{t-1}, u_{t-1}) + v_t = f_{M,\theta^*}(f_{M,\theta^*}^{-1}(x_{t-1}) + [J_{M,\theta^*}W_{t-1}]^{-1} u_{t-1}) + v_t \\
&= x_t + w_t
\end{align*}$$

where $f_{M,\theta^*}$, $f_{M,\theta^*}^{-1}$, and $J_{M,\theta^*}$ denote the learned forward kinematics, inverse kinematics functions, and Jacobian for the model $M$ with optimal model parameters $\theta^*$, respectively. $v_t \sim \mathcal{N}(0, V_t)$ denotes the associated process noise vector due to the modeling errors, and $w_t \sim \mathcal{N}(0, W_t)$ denotes the observational noise vector. Process noise covariance matrix $V_t$ can be calculated from a distribution of predicted poses obtained by using estimated model parameters for different demonstrations for the same object. Since the same perception system is used for both model inference and manipulation, the observation noise covariance matrix $W_t$ is defined to be the same as the model inference observation error matrix $\Sigma_y$ [10].

4.3 POMDP Planning Using Learned Models

Estimating model parameters from noisy observations lead to inherent modeling errors in the learned models. Noisy feedback from imperfect sensors further increases uncertainty in the estimated state. If a motion plan does not account for this state uncertainty, it can easily lead to failures upon execution. Formulating the manipulation planning problem as a POMDP ensures that the state uncertainty is accounted for while developing the manipulation plans. Moreover, a POMDP formulation helps the system to actively take information-gathering actions, increasing the probability of success during execution.

Another challenge in general manipulation tasks is that they may involve making and breaking contacts with the environment. Hybrid dynamics models can be used to accurately model dynamics that change suddenly due to these types of contacts. A hierarchical POMDP planner for manipulation tasks involving contacts, the POMDP-HD planner, has been recently proposed by Jain and Niekum [5]. The POMDP-HD planner exploits the fact that the states marking transitions in the local models of a hybrid dynamics model can be treated as “landmarks” which provide information that can reduce state uncertainty. The POMDP-HD planner first plans at the high-level to develop candidate sequences of local models, “landmarks”, to visit along the path so as to reduce the state uncertainty. Next, it converts the candidate high-level plans to equivalent continuous space trajectories using the low-level planner. The cost-to-go for each of these trajectories is then estimated using...
a cost function penalizing the final state uncertainty while fulfilling the task objective. The continuous space trajectory with the minimum cost is returned as the best plan. A hierarchical POMDP planning approach also helps the POMDP-HD planner to decompose a long horizon POMDP into a sequence of smaller POMDPs for segments between these landmarks. This decomposition helps in making long-horizon POMDP problems tractable, which otherwise, can become intractable due to exponentially increasing computational cost with the time horizon [25]. Motivated by these benefits, we have combined Act-CHAMP with the POMDP-HD planner to develop an end-to-end system that can learn effective hybrid kinematics models for objects from human demonstrations and perform manipulation tasks using the learned models.

5 Experiments

In the first set of experiments, we compare the model inference performance of Act-CHAMP with the CHAMP algorithm [9], to estimate motion models for a microwave and a drawer. Next, we discuss the results of manipulation experiments to open and close the microwave door and the drawer using the learned models in a POMDP setting. Finally, we show how our end-to-end system can learn a hybrid kinematic model for a desk stapler from demonstrations and leverage this knowledge to perform a novel manipulation task. We provided kinesthetic demonstrations to a two-armed robot, in which the human (expert) physically moved the right arm of the robot, while the left arm shadowed the motion of the right arm to interact with objects while collecting unobstructed visual data. Relative poses of object parts were recorded as time-series observations with an RGB-D sensor using SimTrack object tracker [26]. For each time step $t$, the demonstrator’s action on the object was defined as the difference between the position of the right end-effector at times $t+1$ and $t$.

5.1 Learning Kinematics Models for Objects

We collected four sets of demonstrations to estimate motion models for the microwave and the drawer. Two sets provide low-noise data, by manipulating the door handle or drawer knob via a solid grasp. The other two sets provided noisier data, in which the actions were applied by pushing with the end-effector without a grasp.

With grasp: Both algorithms (CHAMP and Act-CHAMP) detected a single changepoint in the articulated motion of the microwave door and determined the trajectory to be composed of two motion models, namely rigid and revolute for a test set of 20 trials with 100% success (Table 1). For the drawer, both algorithms were able to successfully determine its motion to be composed of a single prismatic motion model for a test case of 20 trials with complete success (Table 1). This demonstrates that for clean, information-rich demonstrations, Act-CHAMP can perform on par with the baseline.

Microwave without grasp: When actions are applied directly on the microwave door, the majority of the applied actions are orthogonal to the axis of motion leading to low-information demonstrations. In this case, the observed significant relative displacement was found to be around 67.5% sparser in comparison to the microwave-with-grasp case (on average, only 27.71% of the observed displacements were found to be greater than the threshold of 0.005 rad in comparison to 85.43% for the microwave-with-grasp case). Extremely temporally sparse displacements combined with ob-

\[
\text{No. of time steps for which } y_t \geq 0.005 \text{ units } \times 100 \times \frac{\text{No. of time steps for which } y_t \geq 0.005 \text{ units}}{\text{Total number of time steps in the time series}} \times 100
\]
Table 1: Comparing model and changepoint detection performance of CHAMP and Act-CHAMP. Further analysis is presented in the appendices.

| Experiments                  | CHAMP                      | Act-CHAMP                  |
|------------------------------|----------------------------|----------------------------|
| Microwave with grasp         | 20/20 (100%)               | 20/20 (100%)               |
| Microwave without grasp      | 1/20 (5%)                  | 6/20 (30%)                 |
| Drawer with grasp            | 20/20 (100%)               | 20/20 (100%)               |
| Drawer without grasp         | 18/40 (45%)                | 29/40 (72.5%)              |

servational noise results in poor model inference using either of the algorithms. However, while CHAMP almost completely failed to detect correct motion models (success 5%), Act-CHAMP was able to correctly detect models in almost one-third of the trials with 30% success rate (see Table 1).

**Drawer without grasp:** When the actions were applied by pushing on the drawer, observed significant relative displacement was found to be around 58.92% sparser than the drawer-with-grasp case (19.05% in comparison to 46.65%). Temporal sparsity of relative displacements led CHAMP to falsely detect a changepoint in the provided demonstration and determine that the articulation motion model of the drawer to be composed of two separate prismatic articulation models with different model parameters (third panel from the left in Figure 1). However, due to action-conditional inference, Act-CHAMP correctly classified the motion to be composed of only one articulation model (rightmost panel in Figure 1). Act-CHAMP was found to outperform the CHAMP algorithm, correctly detecting the governing articulation model with 72.5% accuracy (see Table 1).

### 5.2 Object Manipulation Using Learned Models

To demonstrate that the learned models are sufficiently accurate for motion planning under uncertainty, we utilized them to perform the tasks of opening and closing a microwave door and a drawer using a robot manipulator. The POMDP-HD planner plans in the generalized coordinate space for each of the models and afterward converts the plan into Cartesian path waypoints to be followed by the robot end-effector. Figure 2 [left two panels] shows the belief space and actual trajectories for the microwave and drawer manipulation tasks for one task each. For both the objects, low final errors were reported, 0.0548 ± 0.0149 rad for the microwave and 0.56 ± 0.338 cm for the drawer, validating effectiveness of the proposed system.

### 5.3 Leveraging Learned Models for Novel Manipulations

Finally, we show that when combined, our learned models and planner are rich enough to complete novel tasks under uncertainty that require intelligent use of object kinematics. To do so, we use the proposed end-to-end system—Act-CHAMP combined with the POMDP-HD planner—to perform a manipulation task of placing a desk stapler at a target point on top of a tall stack of books. Due to the height of the stack, it is challenging to plan a collision-free path to deliver the stapler to the target location; if the robot attempts to place the stapler at the target point while its governing kinematic model is revolute, the lower arm of the stapler will swing freely and collide with the obstacle. However, a feasible collision-free motion plan can be obtained if the robot first closes and locks the stapler (i.e. rigid articulation), and then proceeds towards the goal. To change the state of the stapler from revolute to rigid, the robot can plan to make contact with the table surface, in order to press down and lock the stapler in a non-prehensile fashion.

As the task involves making and breaking contacts with the environment, we need to extend the learned hybrid motion model of the stapler to include local models due to contacts. We approximately define the contact state between the stapler and the table as to be either a line contact (an edge of the lower arm of the stapler in contact with the table), a surface contact (the lower arm lying flat on the table) or no contact. The set of possible local models for the hybrid task kinematics can be obtained by taking a Cartesian product of the set of possible kinematic models for the stapler and the set of possible contact states between the stapler and the table. However, if the stapler is in the rigid mode, its motion would be the same under all contact states. Hence, a compact task kinematics model would consist of four local models—the stapler in revolute mode with no contact with the table, the stapler in revolute mode with a line contact with the table, the stapler in revolute mode with a surface contact with the table, and the stapler in rigid mode.

Given a human demonstration of robot’s interaction with the stapler as input, the proposed system first learns a hybrid kinematics model for the stapler and then extends it to the hybrid task model using the provided task-specific parameters. Next, the POMDP-HD planner uses the learned
task model to develop motion plans in order to complete the task with minimum final state uncertainty. Motion plans generated by the planner are shown in Figure 2 [right panels]. As can be seen from the plots, the planner plans to make contacts with the table in order to reduce the relative angle between the stapler arms and change the articulation model of the stapler. The plan drags the stapler along the surface of the table, indicating that it waits until it is highly confident that the stapler has become rigid before breaking contact. Making contacts with the table along the path also helps in funneling down the uncertainty in stapler’s location relative to the table in a direction parallel to the table plane normal, thereby increasing the probability of reaching the target point successfully. Figure 3 shows the snapshots of the motion plan and actual execution of the robot performing the task.

6 Conclusion

Robots working in human environments require a fast and data-efficient way to learn motion models of objects around them in order to interact with them dexterously. We present a novel algorithm Act-CHAMP, that performs action-conditional model inference to learn hybrid kinematics models of objects from unsegmented human demonstrations. Action-conditional inference enables Act-CHAMP to deduce articulation motion models with higher accuracy than the current state-of-the-art in the presence of noise and leads to the development of models that can be used directly for manipulation planning. Furthermore, we combine Act-CHAMP with the POMDP-HD planner [5], a hierarchical POMDP planner that can plan under hybrid dynamics, to develop an end-to-end system that can learn hybrid kinematics models for objects from demonstrations and use them to perform manipulation tasks involving those objects. Finally, we demonstrate that the proposed end-to-end system can learn the motion model for a complex articulated object (stapler) and use it for a novel task in a manner that has not been previously observed.

One advantage of using an action-conditional model inference approach over observation-only approaches is that it can enable robots to take informative exploratory actions for learning object motion models autonomously. Hence, a natural extension for the current work is to develop an active learning framework that can be used by a robot to autonomously learn the motion models of objects in a small number of trials. Another promising direction is to infer more complex quasi-static articulation models between two objects in contact. For example, a cup placed on a table cannot move towards the table plane as long as contact exists between them. This restriction can be viewed as though the contact has caused the two objects to be in a quasi-static articulation relationship.

---

3 Experimental details are included in the appendices.
Taken together, these advances will move the state-of-the-art closer to a task-generic understanding of planning with complex objects in contact-rich settings.

References

[1] S. Levine, N. Wagener, and P. Abbeel. Learning contact-rich manipulation skills with guided policy search. In 2015 IEEE international conference on robotics and automation (ICRA), pages 156–163. IEEE, 2015.

[2] J. Fu, S. Levine, and P. Abbeel. One-shot learning of manipulation skills with online dynamics adaptation and neural network priors. In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, pages 4019–4026. IEEE, 2016.

[3] N. Mishra, P. Abbeel, and I. Mordatch. Prediction and control with temporal segment models. arXiv preprint arXiv:1703.04070, 2017.

[4] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In 2018 IEEE International Conference on Robotics and Automation (ICRA), pages 7559–7566. IEEE, 2018.

[5] A. Jain and S. Niekum. Efficient hierarchical robot motion planning under uncertainty and hybrid dynamics. In Conference on Robot Learning, pages 757–766, 2018.

[6] E. Brunskill, L. Kaelbling, T. Lozano-Perez, and N. Roy. Continuous-State POMDPs with Hybrid Dynamics. Symposium on Artificial Intelligence and Mathematics, pages 13–18, 2008.

[7] J. Pineau, G. Gordon, and S. Thrun. Policy-contingent abstraction for robust robot control. In Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence, pages 477–484. Morgan Kaufmann Publishers Inc., 2002.

[8] M. Toussaint, L. Charlin, and P. Poupart. Hierarchical pomdp controller optimization by likelihood maximization. In UAI, volume 24, pages 562–570, 2008.

[9] S. Niekum, S. Osentoski, C. G. Atkeson, and A. G. Barto. Online bayesian changepoint detection for articulated motion models. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pages 1468–1475. IEEE, 2015.

[10] J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects. Journal of Artificial Intelligence Research, 41:477–526, 2011.

[11] S. Pillai, M. R. Walter, and S. Teller. Learning articulated motions from visual demonstration. arXiv preprint arXiv:1502.01659, 2015.

[12] P. R. Barragán, L. P. Kaelbling, and T. Lozano-Pérez. Interactive bayesian identification of kinematic mechanisms. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 2013–2020. IEEE, 2014.

[13] J. Sturm, V. Pradeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard. Learning kinematic models for articulated objects. In IJCAI, pages 1851–1856, 2009.

[14] D. Katz and O. Brock. Manipulating articulated objects with interactive perception. In 2008 IEEE International Conference on Robotics and Automation, pages 272–277. IEEE, 2008.

[15] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In 2013 IEEE International Conference on Robotics and Automation, pages 5003–5010. IEEE, 2013.

[16] R. M. Martin and O. Brock. Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2494–2501. IEEE, 2014.

[17] M. Watter, J. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In Advances in neural information processing systems, pages 2746–2754, 2015.
[18] A. Byravan and D. Fox. Se3-nets: Learning rigid body motion using deep neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 173–180. IEEE, 2017.

[19] A. Byravan, F. Lee, F. Meier, and D. Fox. Se3-pose-nets: Structured deep dynamics models for visuomotor control. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[20] C. Pérez-D’Arpino and J. A. Shah. C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy. In *Robotics and Automation (ICRA)*, 2017 IEEE International Conference on, pages 4058–4065. IEEE, 2017.

[21] G. Subramani, M. Gleicher, and M. Zinn. Recognizing geometric constraints in human demonstrations using force and position signals. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[22] G. Subramani, M. Zinn, and M. Gleicher. Inferring geometric constraints in human demonstrations. *arXiv preprint arXiv:1810.00140*, 2018.

[23] P. Fearnhead and Z. Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.

[24] J. Lygeros, S. Sastry, and C. Tomlin. Hybrid systems: Foundations, advanced topics and applications. *under copyright to be published by Springer Verlag*, 2012.

[25] C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

[26] K. Pauwels and D. Kragic. Simtrack: A simulation-based framework for scalable real-time object pose detection and tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany, 2015.

[27] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez. Lqr-rrt*: Optimal sampling-based motion planning with automatically derived extension heuristics. In *2012 IEEE International Conference on Robotics and Automation*, pages 2537–2542. IEEE, 2012.
A Experimental Details

A.1 Demonstration Setup

Figure 4: Snapshots of human demonstrations provided to record data for the microwave and the drawer. (left to right) Microwave with grasp, Microwave without grasp, Drawer with grasp, and Drawer without a grasp.

A.2 Model Inference Performance Evaluation

| Experiments        | Algorithms | Changepoint Detection | Error in Estimated Parameters |
|--------------------|------------|-----------------------|--------------------------------|
|                    |            | Success | Position Error | Center: | Axis: | Radius: |
| Microwave with grasp | CHAMP     | 20/20 (100%) | 0.0010 ± 0.0007 m | 0.0461 ± 0.0238 m | 0.0165 ± 0.0128 rad | 0.0266 ± 0.0241 m |
|                    | ActCHAMP  | 20/20 (100%) | 0.0014 ± 0.0010 m | 0.0661 ± 0.0340 m | 0.0268 ± 0.0196 rad | 0.0129 ± 0.0586 m |
| Microwave with grasp | CHAMP     | 1/20 (5%)   | 0.0007 m           | 0.5103 m          | 0.4166 rad         | 0.0559 m               |
|                    | ActCHAMP  | 6/20 (30%)  | 0.0011 ± 0.0008 m | 0.3281 ± 0.1245 m | 0.5877 ± 0.204 rad  | 0.1637 ± 0.1029 m      |
| Drawer with grasp  | CHAMP     | 20/20 (100%) | —                  | 0.0008 ± 0.0006 m | 0.0011 ± 0.0005 m  |                        |
|                    | ActCHAMP  | 20/20 (100%) | —                  | 0.0015 ± 0.0018 m | 0.0106 ± 0.0051 m  |                        |
| Drawer without grasp | CHAMP    | 18/40 (45%) | —                  | 0.0548 ± 0.0149 rad | 0.115 ± 0.0311 |
|                    | ActCHAMP  | 29/40 (72.5%) | —                 | 0.56 ± 0.338 cm    | 0.0436 ± 0.0293 |

Table 2: Comparing model detection performance of CHAMP and Act-CHAMP.

B Microwave and Drawer Manipulation

We conducted manipulation experiments for the microwave and the drawer for different task conditions. Belief space and actual trajectories for one task for each of the object is shown in Figure 5. More statistical analysis is presented in Table 3.

| Experiments       | Final Error       | Relative Error |
|-------------------|-------------------|----------------|
| Microwave (3 trials) | 0.0548 ± 0.0149 rad | 0.115 ± 0.0311 |
| Drawer (5 trials)  | 0.56 ± 0.338 cm   | 0.0436 ± 0.0293 |

Table 3: Manipulation Performance using Learned Models
Figure 5: Trajectories comparing belief space plans [blue] and the actual trajectories [orange] for different microwave and drawer manipulation tasks using learned models. Error bars represent belief uncertainty.

C   Stapler Experimental Details

Using human demonstrations, the system begins by learning the hybrid kinematics model for the stapler. We define the stapler state in a global frame as \( x = \{x, y, z, \theta\}^T \), where \( \{x, y, z\} \) represents the Cartesian coordinates of the hinge of the stapler (estimated center of rotation for the revolute model) and \( \theta \) represents the angle between the two arms. The control input vector is defined as an applied displacement vector in the Cartesian space \( u = \{x, y, z\}^T \).

We use LQR-RRT* [27] as the low-level planner in the POMDP-HD planner. We define \( \mu_t \) to represent belief over the system states with an associated covariance matrix \( \Sigma_x \). The task objective
is to minimize a cost function $J$, 

$$
J(\mu_{1:T}, u_{1:T}, s_T) = s_T^T \Lambda s_T + \sum_{j=1}^{T} (\mu_i - x_{\text{goal}})^T Q (\mu_i - x_{\text{goal}}) + u_i^T R u_i
$$

(9)

where each edge corresponds to a trajectory having $T$ time steps, $s_T$ is a vector composed of columns of the covariance matrix $\Sigma_x$ at time $T$, $x_{\text{goal}}$ is the goal coordinates in Cartesian space, and $Q$, $R$ and $\lambda$ are cost matrices associated with the state, input, and covariance matrix. Observations consisted of the Cartesian position of the estimated center of rotation and the relative angle between the two stapler arms. The upper arm of the stapler was grasped using the robot’s end effector; however, no predefined location for grasping the stapler was specified.