

# A System for Imitation Learning of Contact-Rich Bimanual Manipulation Policies

Simon Stepputtis<sup>1</sup>, Maryam Bandari<sup>2</sup>, Stefan Schaal<sup>2</sup>, Heni Ben Amor<sup>3</sup>

**Abstract**—In this paper, we discuss a framework for teaching bimanual manipulation tasks by imitation. To this end, we present a system and algorithms for learning compliant and contact-rich robot behavior from human demonstrations. The presented system combines insights from admittance control and machine learning to extract control policies that can (a) recover from and adapt to a variety of disturbances in time and space, while also (b) effectively leveraging physical contact with the environment. We demonstrate the effectiveness of our approach using a real-world insertion task involving multiple simultaneous contacts between a manipulated object and insertion pegs. We also investigate efficient means of collecting training data for such bimanual settings. To this end, we conduct a human-subject study and analyze the effort and mental demand as reported by the users. Our experiments show that, while harder to provide, the additional force/torque information available in teleoperated demonstrations is crucial for phase estimation and task success. Ultimately, force/torque data substantially improves manipulation robustness, resulting in a 90% success rate in a multipoint insertion task. Code and videos can be found at <https://bimanualmanipulation.com/>

## I. INTRODUCTION

Manipulation still remains a critical challenge of robotics [1]. Over the past decades, there has been tremendous progress in endowing robots with motor skills for grasping and dexterity. However, the vast majority of work in this field focuses on scenarios involving a single robot arm and tightly controlled physical interactions with the environment. With decreasing prices, as well as the proliferation of collaborative and humanoid robotics, there is increased need for techniques that enable reliable, efficient and safe bimanual manipulation. Bimanual robots need to perform manipulation tasks that involve multiple points of contact and dynamic force exchange with objects (and humans) in their surroundings, e.g., lifting a box, inserting a tight-fitting part, unscrewing a bottle cap, or reacting to a human push. However, making early or premature contact with a target object may create forces that seriously jeopardize the manipulation process. In addition to physical interaction with their surroundings, the individual robot arms in a bimanual setup may themselves be exchanging forces and torques through manipulated objects. These forces may lead to oscillations, instabilities and damage to the underlying hardware. Consequently, compliant

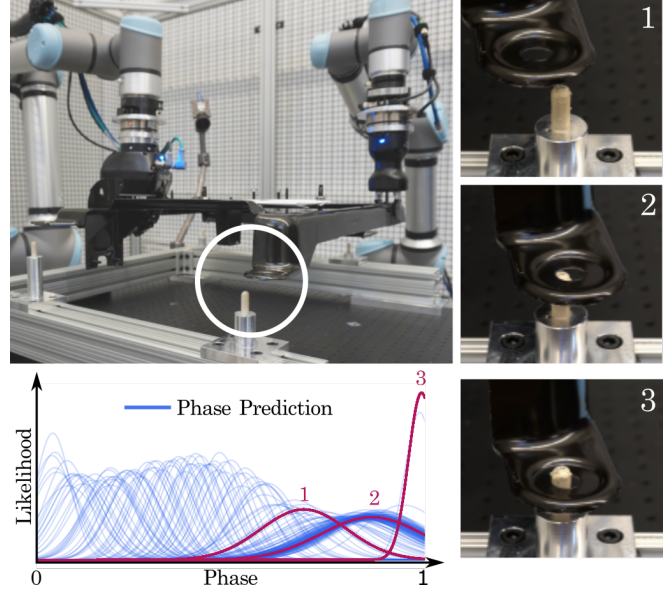


Fig. 1. Overview of the bimanual insertion task in which the two robots need to jointly insert the bracket onto four pins. The phase plots and detailed pictures show how our method is capable of adjusting the bracket's position.

control policies are required that allow bimanual robots to (a) deal and adapt to a wide variety of disturbances in space and time, and (b) effectively leverage physical contact to their advantage. Yet, designing control frameworks that can bridge these (potentially conflicting) requirements can be challenging and time-consuming. To date, only few publications have addressed such challenges underlying bimanual manipulation [2].

In this paper, we present a system for imitation learning of compliant bimanual manipulation policies. We describe a robotic setup in which human demonstrations of the task are recorded across a variety of sensing modalities. The setup leverages principles of admittance control to enable contact-rich and dynamic demonstrations without the risk of collisions, damage or wear-and-tear. In turn, the recorded data is used to learn Interaction Primitives which encode the demonstrated behavior in time and space. At runtime, interaction primitives are used to identify the temporal task progress as a function of external perturbations, as well as the optimal robot response to these perturbations and environmental conditions. In cases where physical perturbations affect task execution, the robot is able to account for it by performing corrective actions in either time or space. Since the presented approach is Bayesian in nature, it allows for

<sup>1</sup>Simon Stepputtis is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, United States [stepputtis@cmu.edu](mailto:stepputtis@cmu.edu)

<sup>2</sup>Maryam Bandari and Stefan Schaal are with [Google] Intrinsic, Mountain View, United States [sschaal, maryamb@google.com](mailto:sschaal, maryamb@google.com)

<sup>3</sup>Heni Ben Amor is with the School of Computing and Augmented Intelligence, Arizona State University, Tempe, United States [hbenamor@asu.com](mailto:hbenamor@asu.com)

powerful spatio-temporal inference from multimodal datastreams.

The main objective of this paper is to provide insights and solutions regarding a number of technical and theoretical considerations that have to be taken into account for contact-rich manipulation tasks in bimanual setups. We argue that four key components influence the performance of a system for imitation learning in bimanual manipulation. Specifically, we will discuss methodologies for a) *data collection*, b) *motor skill learning*, c) *task phase estimation*, and d) *compliance through sensing and control*. A critical conclusion in this regard is the **importance of task phase estimation and phase monitoring** during behavior execution.

## II. RELATED WORK

Bimanual manipulation requires the accurate coordination of multiple robot manipulators in the same real-time cycle. Early approaches largely focused on planning techniques in order to generate kinematic configurations that are feasible for both involved arms [3]. Due to the computational costs, more recent work [4] introduced precomputed reachability analysis and medial axis transforms to efficiently generate candidate robot configurations. In a similar vein, the work in [5] presents a certifiably-complete manipulation planner for assembly tasks involving two robots. However, such planning-based approaches assume quasi-static motions and do not provide the responsiveness needed for contact-rich, dynamic manipulation tasks. An alternative approach is to use machine learning techniques in order to extract reactive manipulation policies. The work in [6] used reinforcement learning (RL) in a low-dimensional space of synergies in order to find policies that optimally coordinate both arms. Amadio et al. [7] leverage symmetries in the kinematic structure to reduce the sample complexity of the trial-and-error process underlying reinforcement learning. Despite these optimizations, RL is a time-consuming process that does not leverage existing human knowledge about such bimanual tasks as found in industrial applications. In addition, RL on real robot platforms comes with a substantial burden of repeatedly resetting the experiment to its initial state, as well as wear-and-tear on robots and sensors. By contrast, imitation learning promises to use only a limited set of human expert demonstrations in order to extract an underlying policy. The work in [8] uses such an approach to learn primitives for bimanual manipulation. However, the work largely focuses on how to sequence individual primitives in order to realize a long-horizon task. Our work is closest related to the work in [9] and [10] in which Dynamic Motor Primitives (DMP) are used to learn bimanual manipulation policies from demonstrations. For an excellent overview article on robot learning for manipulation, we refer the reader to [2].

## III. METHODOLOGY

In this section, we will introduce our system for imitation learning in bimanual settings and discuss a variety of considerations regarding data collection, learning and compliance. The system learns compliant bimanual manipulation policies

from human demonstrations. Without loss of generality, we will focus on a specific task wherein two robot arms are required to first lift a bracket, which features four alignment grommets. Once lifted, the bracket is to be carefully inserted onto a set four pegs via the four grommets. Due to the multiple distributed positions, task execution typically involves substantial physical contact between the bracket and the pegs. Accordingly, after first contact, the bracket position and orientation may have to be repeatedly corrected for successful insertion. For an overview of this task see Fig. 1.

### A. Data Collection for Bimanual Manipulation

An important first consideration when learning such a delicate manipulation task is the collection of training data. Our system provides two alternative approaches to data collection, namely kinesthetic teaching [11] and tele-operation, as can be seen in Fig. 3. Kinesthetic teaching allows the human expert to provide demonstrations through physical guidance. In the bimanual setup, this can be achieved by either directly touching the involved robots or by moving the manipulated object, thereby applying forces on the attached hands of the robot. Alternatively, a *Space-Mouse* can be used for data collection. A Space-Mouse is a 6 degree-of-freedom (DoF) input device that was first developed for the control and teleoperation of robot arms in space, in particular for the Robot Technology Experiment on the Spacelab D2 mission [12]. Accordingly, its design and functionality is optimized for the demands of manipulation tasks. As can be seen in Fig. 2, a number of sensing modalities are continuously recorded during training, i.e., force-torque values, joint angle readings, tool center points, as well as the position and orientation of the manipulated object.

### B. Motor Skill Learning and Temporal Inference

Given the recorded set of demonstrations, a key next step is to extract a policy or motor skill that generalizes the observed behavior to new situations. A variety of methods can be used for this purpose. Behavioral cloning (BC) with neural networks [13] and Probabilistic Motor Primitives (ProMP) [14] are among the most prominent methods. However, the above methods purely focus on the spatial aspects of motor control, as do most other techniques for imitation learning of motor skills. As a result, robots are not empowered to reason about the temporal evolution of a task and how time progresses. However, for successful bimanual manipulation in contact-rich tasks, robots need to constantly monitor and reevaluate the temporal progress of task execution. As a result of force interactions between the robot and the environment, as well as between the individual manipulators, motor commands may not get accurately executed. Such a situation may be due to a variety of conditions such as physical obstructions, friction, an externally applied force, etc. To overcome such bottleneck situations, it is critical that the robot generates motor commands that are temporally-adequate until all obstacles are overcome.

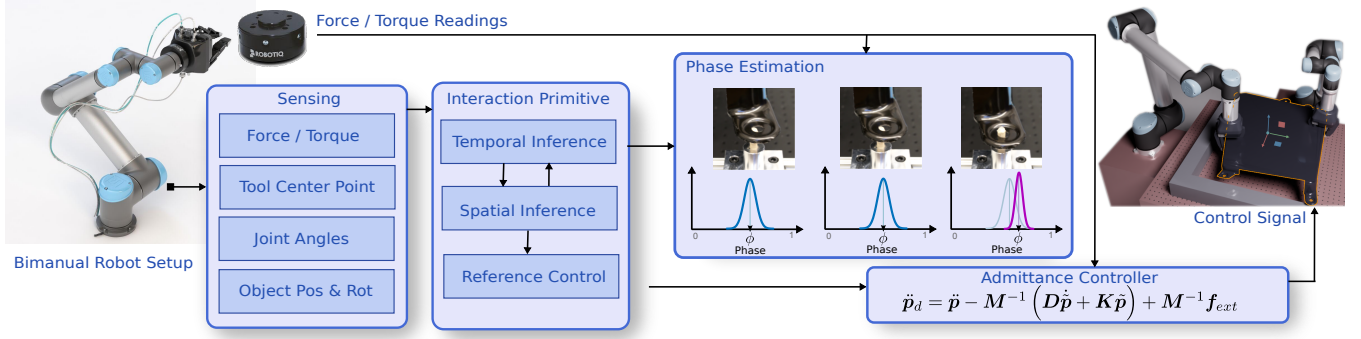


Fig. 2. Overview of the proposed system: sensing data acquired from a number of multimodal sensors is used to learn a Bayesian Interaction Primitive. In case of misalignment, temporal phase estimation and its progression is used with an admittance controller to overcome obstruction.

*Temporal Inference:* In our framework, we adopt a methodology for temporal reasoning inspired by work on human-robot interaction (HRI) [15]. In HRI scenarios, robots need to constantly re-estimate the current temporal phase rather than relying on a predefined internal clock for its progression. In this context, the term *phase* describes the relative temporal position within a task, i.e., the phase variable may be zero at the beginning of a task and one when the task is finished. In HRI, phase estimation is performed by observing the human partner’s movements and, in turn, inferring the most likely position along the time dimension. The work in [16] showed that this procedure is akin to performing robot localization in time rather than space. Bayesian Interaction Primitives (BIP) [17] is an imitation learning approach for HRI which leverages this insight to perform both spatial and temporal reasoning.

In our system, we use BIP in realtime to infer the spatial and temporal state of the execution of the manipulation task. However, rather than estimating the phase by observing an external, human partner, we use the multimodal sensing sources on the robot itself, e.g., force-torque and joint angles. An example of this process can be seen in Fig. 2. In this example, a bracket is carefully placed on four pegs. In the first two images, the object appears stuck on-top of the pegs and insertion cannot proceed successfully. In the plots below the image sequence, we see the estimated phase along with the corresponding variance (i.e. the uncertainty in estimation) visualized as a Gaussian distribution. We notice that the phase estimate roughly remains constant. In the final image, we see that the physical obstruction has been overcome. Accordingly, the phase estimate now moves forward in time. This ability to estimate the phase allows the robot to carefully monitor the progress of a task so as to determine whether to continue task execution or to perform a refinement action. Note that in a BIP the spatial and temporal inference go hand in hand – the robot determines both *what to do* and *when to do it*.

### C. Bayesian Interaction Primitives

Our overall goal in this paper is to learn a bimanual manipulation policy that can generate accurate robot controls from observed states. Each recorded demonstration  $\mathbf{Y} \in$

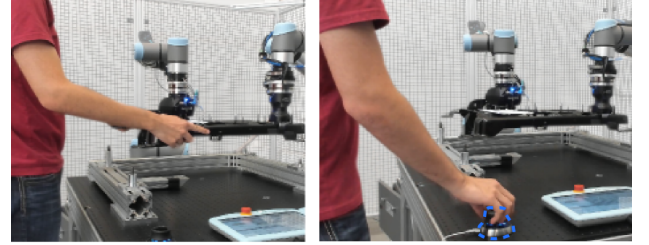


Fig. 3. Data Collection for imitation learning. Left: direct, physical contact allows kinesthetic teaching. Right, the robot arms are controlled via a space mouse. In the above example, the user controls the tool center point of the two robots.

$\mathbb{R}^{D \times T}$  contains  $T$  observations with a  $D$ -dimensional feature vector. Given the current state of the robot, the policy should generate optimal control signals that mimic the behavior of the human demonstrator. However, in the context of human-robot interaction and physical contact with the environment, it is also critical that the manipulation policy adapts to external perturbations. This may be, for example, a push from a human partner or forces generated from premature contact with the environment.

To learn such a policy, we use Ensemble Bayesian Interaction Primitives (EnBIP) [17] <sup>1</sup> and learn a generative probabilistic model over the training demonstrations. However, instead of immediately using time-discretized data, EnBIP transform the recorded demonstrations into a time-invariant representation by performing a basis function decomposition. Such decompositions have been popularized in [10], and have since been used in a number of motor primitive formulations [14], [15]. Applying the basis function decomposition, we now approximate each state dimension  $\mathbf{Y}_t^d = \Phi_{\phi(t)}^\top \mathbf{w}^d + \epsilon_y$  through a linear combination of  $B$  basis functions  $\Phi_{\phi(t)} \in \mathbb{R}^{B^d}$  with corresponding basis weights  $\mathbf{w}^d \in \mathbb{R}^{B^d}$ . Note the shift into a relative time measure known as *phase*  $\phi(t) \in \mathbb{R}$ , where  $0 \leq \phi(t) \leq 1$ , as well as the approximation error  $\epsilon_y$ . Intuitively, this allows the model to represent demonstrations with different lengths with the virtual length  $\phi(t)$ . The above decomposition generates  $d$  weight vectors  $\mathbf{w}$  of the same length  $B$  and is therefore a

<sup>1</sup>Code available at <https://github.com/ir-lab/intprim>

practical first step towards efficient encoding and modelling of the recorded data in a compact fashion. Concatenating all basis weight vectors for the individual dimensions then forms the compressed representation  $\mathbf{w} = [\mathbf{w}^{0\top}, \dots, \mathbf{w}^{D\top}] \in \mathbb{R}^B$  of a given trajectory where  $B = \sum_d B^d$ . In turn, we can now extract a probability distribution  $p(\mathbf{w})$  over all given demonstrations. Sampling from this distribution generates a sample trajectory containing all observed sensor states and robot controls in the bimanual task. Similarly, we can also condition on previous states of the robot to yield a posterior distribution  $p(\mathbf{w}_t | \mathbf{Y}_{1:t}, \mathbf{w}_0)$  over *future* states and controls.

However, a critical insight in BIP is the interplay between temporal and spatial reasoning. An estimation error can be the result of either errors in time or space. So far, inference assumes that the current time step or phase are known. This is typically only the case, when the robot's behavior is unencumbered by physical interactions with humans or the environment. To enable such adaptation, BIP reformulates the problem as a joint spatio-temporal inference – time and space are coupled and are jointly estimated. This insight is realized by forming a new state vector  $\mathbf{s} = [\phi, \dot{\phi}, \mathbf{w}]$ . The state vector holds both a spatial component, contained in the basis function weights  $\mathbf{w}$ , as well as temporal components in the form of the phase  $\phi$  and the phase velocity  $\dot{\phi}$ . The temporal variables  $\phi$  and  $\dot{\phi}$  describe where we are in time and how fast we are progressing with the task. Hence, generating the posterior:

$$p(\mathbf{s}_t | \mathbf{Y}_{1:t}, \mathbf{s}_0) \propto p(\mathbf{y}_t | \mathbf{s}_t) p(\mathbf{s}_t | \mathbf{Y}_{1:t-1}, \mathbf{s}_0). \quad (1)$$

now yields both the information about the spatial and temporal aspects of robot control, since these are encoded in  $\mathbf{s}_t$ . Performing the above inference step can efficiently be done via recursive filtering. More specifically, we use an Ensemble Kalman Filter (EnKF) as proposed in [18]. A major advantage of EnKF is its ability to model complex nonlinear distributions without having to specify any parametric family. The initial distribution is immediately formed by the provided demonstrations – no fitting to a parametric family of densities is needed. In addition, EnKF can be used with nonlinear transition functions and observation functions. As a result, linearization errors as found in other filters can be avoided. When compared to particle filters (PF) [19], EnKF avoid the problem of sample degeneracy and are typically more sample-efficient, i.e., need less ensemble members, than PF.

We start by defining an ensemble  $\mathbf{X}$  of  $E$  members shown by  $\mathbf{X} = [\mathbf{x}^1, \dots, \mathbf{x}^E]$ . Optimally we want to sample the initial ensemble  $\mathbf{X}_0$  directly from the prior  $\mathbf{x}_0 \sim p(\mathbf{w}_0)$  for all  $\mathbf{x}_0 \in \mathbf{X}_0$ ; however, since we do not have direct access to  $p(\mathbf{w}_0)$ , as a data-driven method, it is standard to instead sample from observed training demonstrations. Random selection on ensemble members is reasonable as the ensemble-based filtering approach provides robustness against possible non-Gaussian uncertainties, provided the number of ensemble members is not less than the number of example demonstrations  $E \leq N$ . As a two-step Bayesian estimation method, our first step approximates  $p(\mathbf{w}_t | \mathbf{Y}_{1:t-1}, \mathbf{w}_0)$  by propagating

each ensemble member forward one time step with:

$$\mathbf{x}_{t|t-1}^j = g(\mathbf{x}_{t-1|t-1}^j) + \epsilon_x, \quad 1 \leq j \leq E, \quad (2)$$

with constant-velocity state transition operator  $g(\cdot)$ , and noise error  $\epsilon_x$ . Next, the ensemble members are updated from the observation and the nonlinear observation operator  $h(\cdot)$ :

$$\mathbf{H}_t \mathbf{X}_{t|t-1} = \left[ h(\mathbf{x}_{t|t-1}^1), \dots, h(\mathbf{x}_{t|t-1}^E) \right]^\top, \quad (3)$$

$$\begin{aligned} \mathbf{H}_t \mathbf{A}_t &= \mathbf{H}_t \mathbf{X}_{t|t-1} \\ &- \left[ \frac{1}{E} \sum_{j=1}^E h(\mathbf{x}_{t|t-1}^j), \dots, \frac{1}{E} \sum_{j=1}^E h(\mathbf{x}_{t|t-1}^j) \right], \end{aligned} \quad (4)$$

The deviation of each ensemble member from the sample mean  $\mathbf{H}_t \mathbf{A}_t$  and the observation noise matrix  $\mathbf{R}$  can then be used to compute the innovation covariance with:

$$\mathbf{w}_t = \frac{1}{E-1} (\mathbf{H}_t \mathbf{A}_t) (\mathbf{H}_t \mathbf{A}_t)^\top + \mathbf{R}. \quad (5)$$

The Kalman gain is likewise calculated directly from the ensemble, with no need to specify an explicit covariance matrix, with

$$\mathbf{A}_t = \mathbf{X}_{t|t-1} - \frac{1}{E} \sum_{j=1}^E \mathbf{x}_{t|t-1}^j, \quad (6)$$

$$\mathbf{K}_t = \frac{1}{E-1} \mathbf{A}_t (\mathbf{H}_t \mathbf{A}_t)^\top \mathbf{w}_t^{-1}. \quad (7)$$

As is typical in recursive filtering, partial observations are sufficient to optimally estimate the full state, which we leverage to generate a posterior over unobservable latent variables, i.e. robot controls. Since the posterior is over weights  $\mathbf{w}$  it defines the controls for all future time steps.

By performing this inference scheme in each time step, we can generate posterior distributions that are conditioned on a multitude of sensors. In Fig. 9, for example, we can see the effect of conditioning an execution on high sensor readings from the force-torque sensor. In this specific case, the robot learned to change direction away from this force exchange.

#### D. Admittance Control

The last layer of our stacked control system is composed of a Cartesian admittance controller (suitable for a UR position controlled robot) with a variable target pose  $\mathbf{p}_d \in \mathbb{R}^6$  and target velocity  $\dot{\mathbf{p}}_d \in \mathbb{R}^6$ . The pose  $\mathbf{p}$  is the concatenation of the robot's Cartesian position and rotation. Internally, rotations are computed with quaternions; however, for notational simplicity, we use Cartesian rotations. While the interaction primitive is performing inference over both robots jointly, the admittance controllers are running separately on each of the two robots with different hyper parameters. The use of admittance control is critical in order to avoid force accumulations due to the closed kinematic-chain, contact with the environment, or force interactions with a human user.

For simplicity, the following description of the controller describes the setup for a single robot. Fundamentally, the



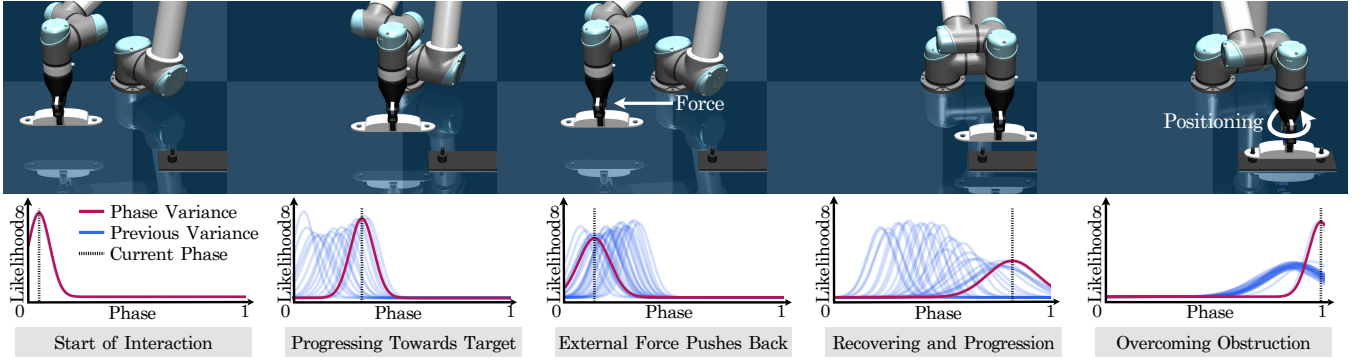


Fig. 4. Image sequence showing the realtime phase estimation  $\phi$  and its variance during the insertion of a bracket. The variance indicates the uncertainty underlying our estimates. In the third image, the robot is pushed backwards through a force that is applied to its gripper. In turn, the phase estimate is adjusted backwards in time during inference. In the final image, a physical obstruction causes the robot to get stuck (visible in the overlapping blue lines). Lastly, even the obstruction is overcome and the phase converges to value one, i.e., the goal is reached.

admittance controller implements the virtual dynamics calculating the external force and torque  $\mathbf{f}_{ext} \in \mathbb{R}^6$  in Cartesian space.

$$\mathbf{M}\ddot{\mathbf{p}} + \mathbf{D}\dot{\mathbf{p}} + \mathbf{K}\tilde{\mathbf{p}} = \mathbf{f}_{ext} \quad (8)$$

where the hyper-parameters are the virtual inertia  $\mathbf{M} \in \mathbb{R}^{6 \times 6}$ , virtual Cartesian stiffness  $\mathbf{K} \in \mathbb{R}^{6 \times 6}$ , virtual damping  $\mathbf{D} \in \mathbb{R}^{6 \times 6}$ , and pose error  $\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}_d$  with current pose  $\mathbf{p}$ . These parameters are set differently for each of the two robots to account for their individual dynamics. Given the desired control signal  $\mathbf{p}_d$  from the interaction primitive, target velocity  $\dot{\mathbf{p}}_d$  of zero, and external force and torque measure  $\mathbf{f}_{ext}$ , the robots' control signal  $\ddot{\mathbf{p}}_d$  is computed from equation 8 as follows

$$\ddot{\mathbf{p}}_d = \ddot{\mathbf{p}} - \mathbf{M}^{-1}(\mathbf{D}\dot{\mathbf{p}} + \mathbf{K}\tilde{\mathbf{p}}) + \mathbf{M}^{-1}\mathbf{f}_{ext} \quad (9)$$

The resulting target position  $\mathbf{p}$  and velocity  $\dot{\mathbf{p}}$  is calculated through integration from  $\ddot{\mathbf{p}}_d$ . This calculation is done separately for each robot and is sent to be executed with an inverse kinematics to get the desired joint angles for each robot.

#### IV. EVALUATION

We evaluate our approach in two separate settings. In the first setting, a single, simulated robot is placing a bracket with two grommets onto two pins (see Fig. 4). In this setup, a single 6 Degree of Freedom (Dof) UR5 robot is equipped with a force-torque sensor between the flange of the arm and a parallel jaw gripper, tasked with placing one of two different brackets with varying tolerances (1mm and 5mm) onto two pins. The simulated environment is implemented in MuJoCo [20]. In a second, bimanual setup on two real robots (as previously introduced in Fig. 1), a UR5 robot and a UR10 robot with 6Dof each are used. The two robots are equipped with force-torque sensors located between the end-effector and the gripper. However, while the UR5 robot uses a parallel jaw gripper, the UR10 includes an adaptive three finger gripper for increased stability during the grasp, thereby preventing undesired tilting of the object.

In both settings (single-arm in simulated and dual-arm in real experiments), we utilize a path planner with pre-determined way points to pick up the bracket from slightly randomized position to perform the initial lift while ensuring different grasping poses. The initial bracket position is varied within a margin of  $\pm 1$  cm. However, approaching the pins, as well as the final insertion task is performed via the described imitation learning model. Sufficiently precise manipulation is required to successfully insert the bracket given the 1 and 5 mm tolerance in simulation and 6 mm tolerance per pin in the real world bimanual setup. Note that this task does not cover Transition or Interference fits with potentially negative tolerances. In the simulated task, forces largely stem from contacts with surrounding objects while in the real-world task, forces may also be exchanged between the two robot arms during manipulation.

In the following sections, we evaluate the proposed system with respect to the different components, i.e., data collection, motor skill learning, as well as compliance and dynamic phase estimation. We also compare our method with a behavior cloning (BC) and ProMP [14] baseline. The latter is another common method derived from Dynamic Motor Primitives; however, it requires separate methods for spatio-tempora alignment of the motion.

##### A. Data Collection

To collect the required data for training the model, we use 30 demonstrations for the simulated and real-world setup each. For the simulated environment, 30 demonstrations are collected from pre-programmed and slightly randomized Bézier curves that alter the curvature of the generated motion, utilizing a 1mm tolerance bracket. Using the bracket with the smaller tolerance for training purposes is expected to also yield a successful model for the bracket with larger tolerances.

Similar to the simulated setup, 30 demonstrations are collected in the real-world setup. Fundamentally, demonstrations are collected from eight human subjects for a total of four datasets, utilizing either the Space-Mouse, or kinesthetic teaching. Additionally, one dataset each is collected with

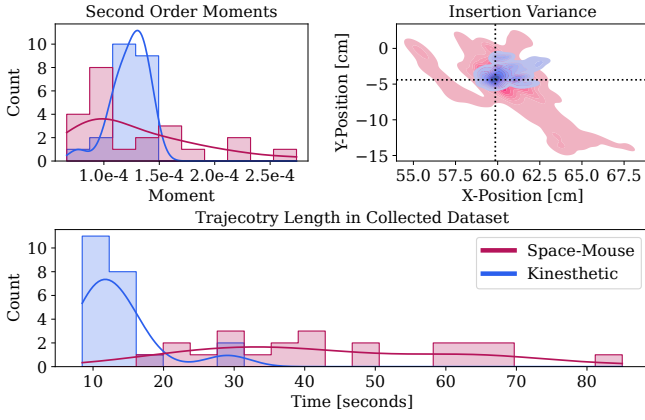


Fig. 5. Analysis of the recorded user data with regards to smoothness (top left), variance close to the final insertion (top right) and demonstration length (bottom).

varying starting positions of the bracket within a margin of three centimeters. With the Space-Mouse, the participants control the position of the bracket, inherently moving the two robots via the reference poses  $p_{ref}$  of each robot’s admittance controller through a fixed transformation from the bracket’s pose. When using kinesthetic teaching, the underlying admittance controllers allow the participants to freely move the bracket inside the overlapping workspace of the robots by continuously updating the controller’s reference pose with the bracket’s sensed position. To familiarize the participants with the intricacies of each training method, each participant had the opportunity to provide an initial, unrecorded demonstration.

1) *Kinesthetic vs. Space-Mouse Data*: A fundamental difference between the training methods is the availability of force-torque sensor data. When using kinesthetic teaching, external forces are induced by the human teacher during the demonstration, rendering sensed force-torque data unusable for subsequent skill learning. In contrast, collected force-torque data when using the Space-Mouse can be used for skill learning as only contact forces with the environment and forces resulting from the interaction of the two robots are measured.

Figure 5 shows a comparison of the demonstrations collected with kinesthetic teaching (blue) and the Space-Mouse (red) on the real robot. Generally demonstrations collected by using kinesthetic teaching result in smoother motion, shorter trajectories and reduced final adjustments prior to the final insertion. Prior to the final insertion of the pins, demonstrations collected with the Space-Mouse have a variance of 2.173 cm of the bracket location, while demonstrations collected with kinesthetic teaching show a significantly lower variance 0.886 cm. The need for final adjustments is also reflected in the average trajectory length of 44 and 14 seconds for the Space-Mouse and kinesthetic method, respectively.

2) *NASA TLX Workload*: In addition to evaluating the collected data itself, we also evaluated how the human participants perceive the workload of providing demonstrations

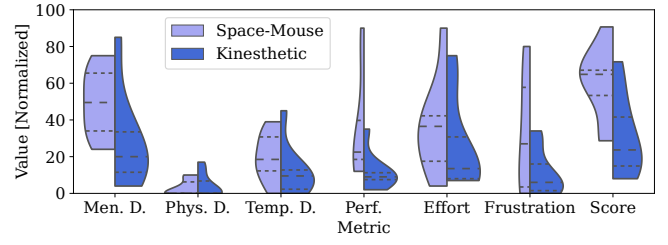


Fig. 6. Nasa TLX Workload evaluation. Metrics are weighted by how important the participants’ deemed them and normalized to [0, 100]

Collection Method		Dataset Variations			Success
		Extended	With F/T Data	Perturbed	
1	Space-Mouse				100.0%
2	Space-Mouse			✓	13.33%
3	Space-Mouse		✓		90.0%
4	Space-Mouse		✓	✓	20.0%
5	Space-Mouse	✓			100.0%
6	Space-Mouse	✓		✓	70.0%
7	Space-Mouse	✓	✓		93.3%
8	Space-Mouse	✓	✓	✓	90.0%
9	Kinesthetic				100.0%
10	Kinesthetic			✓	0.0%
11	Kinesthetic	✓			100.0%
12	Kinesthetic	✓		✓	73.33%

TABLE I

EVALUATION RESULTS ON THE REA-WORLD BIMANUAL SETUP

with each teaching method. While *workload* is a subjective measure and varies between different participants, the NASA Task Load Index [21], [22] provides a methodical assessment of the workload a user is experiencing when completing a task. We evaluate six categories: mental, physical, and temporal demand, as well as perceived performance, required effort and the users’ frustration using each of the training modalities. However, since every user has a different subjective assessment of how much each category influences the overall perceived workload, weights for each category are derived prior to the assessment to further increase the sensitivity of the metrics across multiple subjects.

Figure 6 shows the results of the workload assessment across eight users. Particularly the perceived performance as well as the frustration of the participants are significantly increased when using the Space-Mouse. This results is an interesting dichotomy: despite higher frustration, successfully completing the task with the Space-Mouse significantly increased the participants’ perception of performance. Overall, participants rated using the Space-Mouse as approximately twice as labour-intensive as kinesthetic teaching, with means of 59.33 and 30.88 respectively.

While the analysis of the data as well as the perceived workload from the users, positions kinesthetic teaching as favorable, its major drawback, however, is the inability to record meaningful force-torque sensor data.

### B. Motor Skill Learning

Table I shows the results of EnBIP on the real-world bimanual insertion task. We evaluated our approach with

Model	Tolerance	Disturbance	Success
1 ProMP	5mm		100.0%
2 ProMP	5mm	✓	0.0%
3 ProMP	1mm		33.33%
4 EnBIP	5mm		100.0%
5 EnBIP	5mm	✓	100.0%
6 EnBIP	1mm		90.0%

TABLE II  
RESULTS IN SIMULATED EXPERIMENTS

demonstrations accounting for varying starting positions (column *Extended Dataset*), usage of force-torque data (column *Force Sensor*), and variable starting positions of the bracket (column *Varied Position*). A test is counted as successful if all four grommets are placed on the correct pin and both robots released the bracket. The success rate is reported over 30 evaluations.

Using either dataset, a successful motor skill can be learned when the bracket is in a fixed starting position, resulting in a 100% success rate (lines 1, 5, 9, and 11 in Table I), even when the datasets are extended via demonstrations with varying starting positions of the bracket. However, when varying the starting position of the bracket during testing, we observe a 30% drop (lines 6 and 12) in success rate. This can be attributed to the robots not being able to accurately sense the obstacle when attempting final insertion. When adding the force-torque sensor data, which is exclusively available in the dataset collected with the Space-Mouse, the success rate increases to 90%, (line 8 of Table I). Adding force-torque sensor data is therefore critical for robustness when varying starting positions of the bracket and results in an overall performance improvement of 20%, compared to a setting in which no force-torque data is used (line 6 compared to line 8). As a result, while kinesthetic teaching produces cleaner data and is easier for participants to perform, the availability of force-torque data dramatically increases robustness in contact-rich tasks.

### C. Dynamic Phase Estimation

To compare the impact of online phase estimation in the presence of external disturbances, we compare EnBIP to ProMP on two different brackets in the simulated environment. Figure 4 shows the task in simulation on the top, as well as the current (red) and history of previous (blue) phase estimate. Just after introducing an external force in picture two, our model only slightly increased its uncertainty during the reverse motion (step 3); however, in step five where the robot has placed the bracket

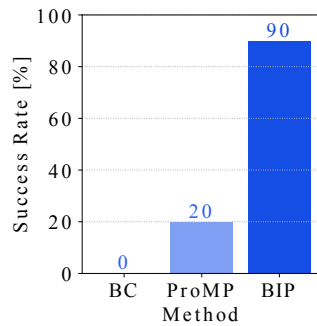


Fig. 7. Baseline comparison with varying starting pose

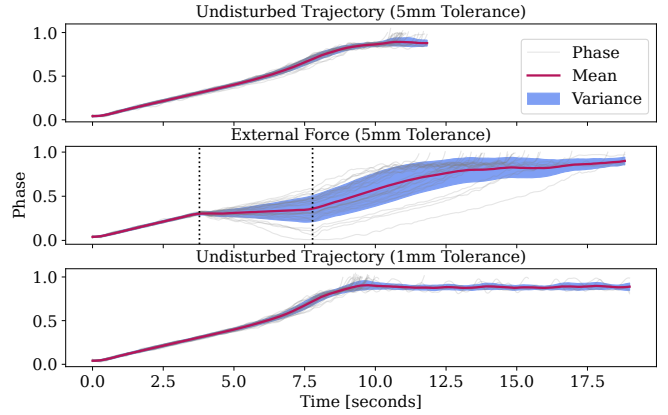


Fig. 8. Phase progression in three scenarios: Undisturbed (top), External force (middle), and small tolerances (bottom) in the simulated environment.

on the pins, the uncertainty has increased just before the final insertion was made. A similar behavior to the latter can also be seen in Figure 1 in the real-world experiment. This example shows the crucial influence of phase estimation and correction for successful task completion.

Figure 8 shows the estimated phase progression over 30 trials in the simulated environment of Figure 4 for three different scenarios: At the top, the bracket with 5mm tolerance is placed on the two pins without any external disturbance, expecting a diagonal line from phase 0 to 1. The middle figure introduces an external force applied between 4.5 and 7.5 seconds. The applied force is drawn from a normal distribution with a mean of 1.75 Newton and a variance of 0.5 Newton. Gray lines show the actual phase estimation of the 30 trials in addition to the mean (red) and variance (blue). Even though the motion is disturbed, the model is able to gracefully recover by adjusting the phase, ultimately succeeded in 100% if the tasks (Table II line 4 and 5). Finally, the lowest plot shows the phase progression with the 1mm tolerance bracket without any disturbances, besides the ones occurring directly upon insertion of the pins. This bracket requires a higher precision, thus while a straight diagonal line from start to end would be expected, the model adjusts the bracket's position on the pins for an extended period of time. Ultimately the model succeeds in 90% of the trials while failures are due to prematurely estimating task completion, or not finding the pins (Table II line 4 and 5).

Table II shows a comparison with ProMP in the simulated environment, utilizing a fixed phase progression that is sampled uniformly across the lengths of the training demonstrations. Line 5 shows a significantly improved success rates over ProMP in line 2 when external forces are applied to the system, underlining the importance of phase estimation.

While the model is able to recover from external perturbations, within reasonable limits, it is crucial that the recovery actions follow the demonstrated behavior. Figure 9 shows two scenarios in which either an external force, or an error in the joint sensors were introduced. In both cases, the model is able to adjust the motion accordingly (red lines) as compared to the demonstrated behaviors (blue lines), underlining our

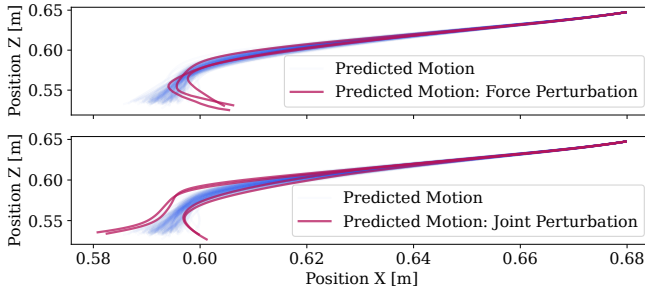


Fig. 9. Generalization capabilities of our IntPrim model. Conditioned on the initial object pose, the model predicts similar trajectories over consecutive runs (blue), however, introducing disturbances to the F/T sensors (top) or joint position sensors (bottom) causes the model to recondition the motion (red), following the general motion trend, but unable to complete the task.

model’s ability to adapt to these perturbations due to their connection learned in the EnBIP.

#### D. Baseline Comparison

We also compare our model against a BC [13] and a ProMP [14] baseline in the real-world bimanual insertion task, as shown in Figure 7. Similar to Bayesian Interaction Primitives, ProMPs are also derived from Dynamic Motor Primitives [10]. However, they do not include joint spatial and temporal reasoning. Instead, separate methods, such as Dynamic Time Warping (DTW) [23] may have to be used for temporal reasoning. For the purposes of our comparison, we used ProMPs in conjunction with DTW which resulted in a success rate of 20%. BC did not successfully complete any of the 30 attempts. By comparison, BIC achieves a success rate of 90%. These results highlight again the role of dynamic phase estimation in successful task completion.

#### V. CONCLUSION

In this paper, we introduced a framework for learning compliant bimanual policies from human demonstrations in contact-rich environments. Our approach combines the capabilities of admittance control and machine learning in order to enable efficient use of contacts with the environment while being able to adapt to a variety of external disturbances. We show that spatio-temporal inference plays a major role in creating safe, reliable and efficient control signals for physical interaction with objects and humans in a complex bimanual insertion task. Using the introduced methodology we achieve a success rate of 90%. Further, we have conducted a user study to identify optimal data collections interfaces. Our study shows that participants largely preferred kinesthetic teaching to teleoperation with a Space-Mouse; however, the multimodal data from the Space-Mouse provides crucial information for significantly improved task performance of the learned policy.

*Limitations:* While our approach is sample-efficient and can adapt to a range of perturbations, it is limited by the quality of the training data. The demonstrated behavior is assumed to be near-optimal in order to allow for precise phase estimation. Accordingly, the current system needs to

be extended to better adjust for demonstrations from naive teachers [24].

#### REFERENCES

- [1] A. Billard and D. Kragic, “Trends and challenges in robot manipulation,” *Science*, vol. 364, p. eaat8414, 06 2019.
- [2] O. Kroemer, S. Niekum, and G. D. Konidaris, “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *Journal of machine learning research*, vol. 22, no. 30, 2021.
- [3] Y. Koga and J.-C. Latombe, “Experiments in dual-arm manipulation planning,” in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2238–2245 vol.3.
- [4] N. Vahrenkamp, D. Berenson, T. Asfour, J. Kuffner, and R. Dillmann, “Humanoid motion planning for dual-arm manipulation and re-grasping tasks,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 2464–2470.
- [5] P. Lertkultanon and Q.-C. Pham, “A certified-complete bimanual manipulation planner,” *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1355–1368, 2018.
- [6] K. S. Luck and H. Ben Amor, “Extracting bimanual synergies with reinforcement learning,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4805–4812.
- [7] F. Amadio, A. Colomé, and C. Torras, “Exploiting symmetries in reinforcement learning of bimanual robotic tasks,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1838–1845, 2019.
- [8] R. Lioutikov, O. Kroemer, G. Maeda, and J. Peters, “Learning manipulation by sequencing motor primitives with a two-armed robot,” in *Intelligent Autonomous Systems 13*. Springer, 2016, pp. 1601–1611.
- [9] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, and S. Schaal, “Skill learning and task outcome prediction for manipulation,” in *2011 IEEE International Conference on Robotics and Automation*, 2011.
- [10] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: learning attractor models for motor behaviors,” *Neural computation*, vol. 25, no. 2, pp. 328–373, 2013.
- [11] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, “Handbook of robotics chapter 59: robot programming by demonstration,” *Handbook of Robotics*. Springer, 2008.
- [12] G. Hirzinger, N. Sporer, M. Schedl, J. Butterfaß, and M. Grebenstein, “Torque-controlled lightweight arms and articulated hands: Do we reach technological limits now?” *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 331–340, 2004.
- [13] D. A. Pomerleau, “Efficient training of artificial neural networks for autonomous navigation,” *Neural Computation*, vol. 3, no. 1, pp. 88–97, 1991.
- [14] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, “Probabilistic movement primitives,” *Advances in neural information processing systems*, vol. 26, 2013.
- [15] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, “Interaction primitives for human-robot cooperation tasks,” in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2831–2837.
- [16] J. Campbell and H. B. Amor, “Bayesian interaction primitives: A slam approach to human-robot interaction,” in *Conference on Robot Learning*. PMLR, 2017, pp. 379–387.
- [17] J. Campbell, S. Stepputtis, and H. Ben Amor, “Probabilistic multimodal modeling for human-robot interaction tasks,” in *Robotics: Science and Systems*, 2019.
- [18] M. Roth, G. Hendeby, C. Fritzsche, and F. Gustafsson, “The ensemble kalman filter: a signal processing perspective,” *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, Aug. 2017.
- [19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. Cambridge, Mass.: MIT Press, 2005.
- [20] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.
- [21] S. Hart, “Nasa-task load index (nasa-tlx); 20 years later,” 2006.
- [22] S. G. Hart and L. E. Staveland, “Development of nasa-tlx (task load index): Results of empirical and theoretical research,” in *Human Mental Workload*, ser. Advances in Psychology, 1988.
- [23] M. Ewerton, G. Neumann, R. Lioutikov, H. B. Amor, J. Peters, and G. Maeda, “Learning multiple collaborative tasks with a mixture of interaction primitives,” in *ICRA*. IEEE, 2015, pp. 1535–1542.
- [24] S. Chernova and A. L. Thomaz, *Robot Learning from Human Teachers*. Springer International Publishing, 2014.