Composable Interaction Primitives: A Structured Policy Class for Efficiently Learning Sustained-Contact Manipulation Skills

Ben Abbatematteo*[1,2], Eric Rosen*[2], Skye Thompson[2],
Tuluhan Akbulut[2], Sreehari Rammohan[2], George Konidaris[2]

***Abstract*— We propose a new policy class, Composable Interaction Primitives (CIPs), specialized for learning sustained-contact manipulation skills like opening a drawer, pulling a lever, turning a wheel, or shifting gears. CIPs have two primary design goals: to minimize what must be learned by exploiting structure present in the world and the robot, and to support sequential composition by construction, so that learned skills can be used by a task-level planner. Using an ablation experiment in four simulated manipulation tasks, we show that the structure included in CIPs substantially improves the efficiency of motor skill learning. We then show that CIPs can be used for plan execution in a zero-shot fashion by sequencing learned skills. We validate our approach on real robot hardware by learning and sequencing two manipulation skills.**

I. INTRODUCTION

The unique potential of robots lies in their ability to do physical work in the world — every process that currently requires a human to meaningfully interact with a physical object can only be automated by a robot. Despite this immense potential value, only a tiny fraction of the physical manipulation tasks that can be automated currently are [1]. There are multiple causes of this failure, but one of the most acute is that robots are currently not as flexible as humans in their ability to learn to interact with objects around them. A factory worker can be trained to basic proficiency in an unfamiliar task in a day; skillful and reliable execution of rote manual labor tasks rarely requires more than a few weeks. Achieving the same level of flexibility, reliability, and skill in robots requires major advances in their learning capabilities, so that a robot can be trained to solve a new task, and subsequently improve its own performance, in reasonable time and without the support of expert programmers.

There are broadly two families of approaches to autonomously learning manipulation skills. The first combines end-to-end deep neural networks with reinforcement learning (RL) [2], [3] to learn "pixel to torque" controllers [4] that directly map sensor input to motor output. Such approaches couple RL's promise of flexibility, generality, and autonomy with the opportunity to exploit the power of deep networks. However, they have dauntingly high sample complexity and face difficulty incorporating principled techniques from robotics such as forward and inverse kinematics, motion planning, wrench closure, and feedback control. The second family aims to develop carefully designed and highly structured policy classes [5]–[7] to achieve sample-efficient learning, thereby trading design effort, flexibility, and generality for sample efficiency. Such approaches have learned an impressive range of dynamic behaviors [3], [8] in a feasibly low number of interactions, but are best suited for targeting a restricted class of motor skills where there is structure to be exploited and sample efficiency is paramount.

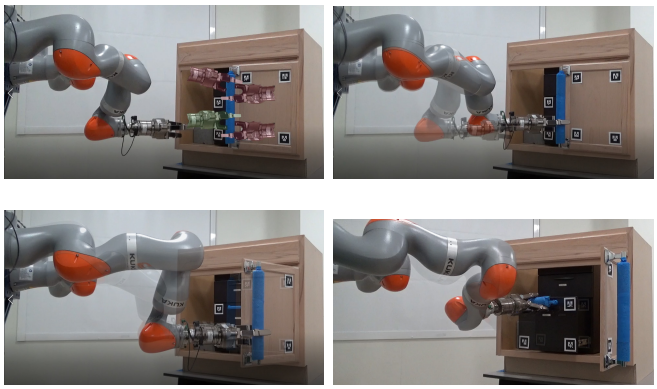


Fig. 1: **Composable Interaction Primitives (CIPs)** are a structured policy class that enables safer, sample-efficient learning of contact-rich manipulation skills by incorporating model-based priors. Our approach (a) identifies promising grasp poses for initiating manipulation (b) uses motion planning to move in free space rather than learning to reach (c) learns model-free policies for interaction where necessary (d) enables composition by construction to support task-level planning.

We focus on one such class, *sustained contact manipulation skills*, where a robot must establish stable contact (in the form of a grasp) with an object in order to change its state, and sustain that contact throughout execution. Examples of such tasks include opening a drawer, pulling a lever, turning a doorknob, opening a door, turning a wheel, or shifting gears. We introduce a new policy class, *Composable Interaction Primitives* (or CIPs), that draws from the best of both motor skill learning approaches: it exploits the structure present in sustained contact tasks, resulting in a policy class that is structured, safe, and highly parameter- (and therefore data-) efficient; and then applies deep networks to the components where learning from high-dimensional input is unavoidable. Additionally, CIPs are sequentially composable by construction, so that learned skills can be sequenced to solve new tasks in an order determined at runtime by a task-level planner. Using an ablation experiment in four simulated manipulation tasks, we experimentally explore the role of

[1]The University of Texas at Austin
[2]Brown University, Providence RI
* denotes equal contribution

structure in manipulation skill learning, and show each of the components of CIPs substantially improves learning efficiency and safety. We then demonstrate the use of CIPs to efficiently learn, and subsequently sequence on-demand, sustained-contact manipulation skills on real robot hardware.

II. BACKGROUND AND RELATED WORK

Motor skills are typically learned using RL [2], where tasks are formalized as a Markov Decision Process $M = (\mathcal{S}, \mathcal{A}, R, T, \gamma)$. The robot's goal is to learn a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ mapping a state to the action it should execute in that state, such that it maximizes the discounted sum of expected future rewards (or *return*).

In many cases, the target motor skill is not the entirety of the robot's task, but should instead be used as an executable subroutine as part of the solution. Such skills are often modeled using the *options framework* [9], where an option o is defined by a tuple (I_o, π_o, β_o), where $I_o \subseteq S$ is the *initiation set*, the set of states from which the robot may choose to execute the option; $\beta_o : S \rightarrow [0, 1]$ is the *termination condition*, giving the probability that option execution ceases in state s; and π_o is the *option policy*. The robot can choose to execute o if the current state is inside I_o, whereupon execution proceeds according to π_o and halts at each encountered state according to β_o. Modeling motor skills as options naturally supports reasoning about sequential compositionality — option o_2 can be executed after option o_1 if the state that o_1 leaves the robot in lies within o_2's initiation set [10]–[13]. Hierarchical RL studies the discovery and composition of such options [14]–[16].

A great deal of recent work has examined the setting where a robot learns to map its sensor input directly to motor torques via deep reinforcement learning [4], [17]. These methods offer flexibility, generality, and autonomy by exploiting recent advances in learning deep networks. However, that generality has a cost: such methods rely on access to massive amounts of compute and data and therefore typically require additional methods that implicitly encode design insight into the data set [18]–[20], collect experience from multiple robots in parallel [21], [22], or include human demonstration [23]–[25]. Additionally, these approaches make it difficult to incorporate the structural knowledge that robotics as a field has developed around techniques like forward and inverse kinematics, motion planning, wrench closure, safety, and feedback control.

An alternative approach is to carefully design and structure a policy class to guarantee desirable properties (e.g., stability, joint and torque limits, and safety constraints) while exploiting the properties of a broad class of target tasks to support sample-efficient learning. The most historically important such policy class is *Dynamic Movement Primitives*, or DMPs [5], [26], which have been used to learn an impressive range of dynamic behaviors [3], [8] in tens or low hundreds of interactions, though they must typically be bootstrapped by an expert demonstration trajectory [24], [27]. The key assumption underlying DMPs is that dynamic motions can be represented largely as a trajectory shape—represented separately for each joint, as a linear combination of learned weights with basis functions over time—coupled with a second-order dynamical system that stably controls the robot towards the shape trajectory. Although DMPs have seen wide usage [1], they have largely not been integrated with high-dimensional, multi-modal data for contact-rich tasks [28]. Other important policy classes overcome the standard shortcomings of DMPs, such as Probabilistic Movement Primitives [29], Conditional Movement Primitives [30]–[32], and Riemannian Motion Policies (RMPs) [6], [7], [33], [34]. These approaches help account for variability across demonstrations, high-dimensional task parameters, and different task spaces but still fail to robustly handle the contact dynamics present in dexterous manipulation [28]. Other approaches incorporate robotics primitives as actions directly in RL [35] or employ motion generation to execute policy actions [36], [37].

A burgeoning area of research is incorporating learned motor skills into task and motion planning (TAMP) solvers [38]–[41]. Cheng and Xu [42] propose a guided skill learning process, but assume existing TAMP infrastructure and heuristically resolve the grasping problem. Silver et al. [43] learn skills in a TAMP framework by segmenting demonstrations consisting of sequences of skills (ala [44]), but rely on sequential demonstrations and do not consider motion planning and other structure like grasping and safety constraints during learning. Our work shows that this structure is critical for efficient skill learning, and does not require demonstrations.

III. COMPOSABLE INTERACTION PRIMITIVES

The success of DMPs suggests that one approach to achieving successful motor skill learning is to match a restricted but important class of motor skills with a representation designed to exploit its structure [5], [26]. We address learning motor skills in the *sustained contact* regime: skills where a robot must establish and maintain contact with an object while exerting force on that object to successfully manipulate it, such as when a human opens a door, pulls a lever, wipes a surface, or shifts gears. Such motor skills are common, complex, and important: much of the work that a robot with a gripper will be tasked with performing in the world—all except pick-and-place and instantaneous contact skills like pushing a button—will require sustained interaction with an unmodeled (or partially modeled) object. They are also highly structured (e.g., including making and breaking contact via grasping), necessitate complex safety constraints such as joint position and torque limits, and demand precise control driven by policies that must be learned from noisy and high-dimensional tactile and force feedback. Finally, they should be designed to support composition: suitable for sequential execution to address new tasks in an order determined at runtime, a capability that is unlikely to occur by chance and can only be achieved through design. All these reasons make sustained-contact motor skills excellent candidates for a specialized motor policy class.

We identify four important properties present in sustained-contact motor skills. First, *skill execution can be decomposed into phases*: the robot first moves through free-space to reach a pre-grasp pose, then achieves a stable grasp, then manipulates the object, then releases its grasp, and finally controls its gripper back into free-space. Second, *most phases involve little or no per-task learning*: motion through free-space and to achieve or release a grasp can be computed using motion planning and feedback control, respectively; the choice of where to grasp the object is a supervised learning task that can be resolved (or at least bootstrapped) using a generic grasp detector [45]. Only the sustained-contact controller itself need be largely learned on a per-task basis, though it could be bootstrapped using learning from demonstration [24] or kinematic motion planning [46]. Third, *the sustained-contact controller itself naturally suggests structure*: the controller must be a function of force- and tactile-feedback, learned using reinforcement learning; the goal of learning should be to reach a task-specific goal (e.g., opening a door, or switching a light on) while avoiding task-general failure modes (like losing contact with the object or becoming stuck); and during learning the policy should be able to explore while being position- and torque-constrained so as to never damage the robot or the object. Here, task-specific structures are components that are either learned or hand-specified for a specific object manipulation skill, whereas task-general structures are components that may be specific to the robot but can be reused across different object manipulation skills. Finally, *a natural means of composition is through free-space motion planning*: motor skills can be sequenced by simply motion planning from one skill's release point to another skill's grasp point.

We therefore propose *Composable Interaction Primitives* (CIPs), a new policy class structured by these insights and aimed at learning composable sustained-contact manipulation skills in tens, rather thousands, of real-world interactions. CIPs are structured as a tuple, where components subscripted by c are specific to the task, and the remainder are specific to the robot but generic across tasks:

$$C = (\pi_c, \sigma, \beta_c, I_c, h, t, \Gamma, B)\text{, where:}$$

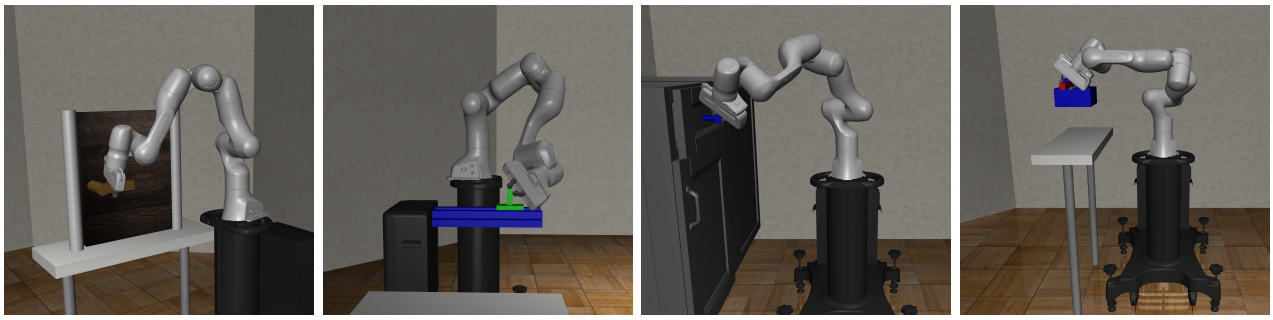
- $\pi_c : \phi \rightarrow \tau$ is a motor control policy that maps tactile sensor signals, proprioceptive data, and object state information to joint torques τ and gripper commands τ_g, with parameters ψ.
- Policy π_c is constrained by σ, a safety envelope specific to the robot but not to the task. Execution is constrained to obey σ so that the agent does not damage the object it is interacting with or itself.
- $\beta_c : \phi \rightarrow \{0, 1\}$ is a task-specific success indicator that maps the robot's observations ϕ to a boolean indicating whether the interaction primitive has achieved its goal.
- B is a task-general classifier indicating interaction failure (e.g., that contact has been lost, the interaction has timed out, or execution cannot continue without a safety constraint being violated). Once initiated, π_c continues execution until either β_c indicates success or B indicates failure. The resulting signal informs a policy search algorithm to optimize π_c.
- $I_c : v,\ g,\ \psi_g \rightarrow [0, 1]$ is the grasp initiation set, a probabilistic classifier conditioned on visual data v that maps end-effector poses g and grasp parameters ψ_g to the probability with which executing π_c from grasp g terminates in β_c (success) as opposed to B (failure). During learning, selecting promising grasps is formulated as a bandit problem.
- h and t are the head and the tail, motion planners that control the robot through free space to achieve a grasp generated by I_c, and extract it from contact back into free space—or into the head of another skill—after termination. These serve to establish and break contact, and to sequence skills: the tail of one skill simply becomes the head of another.
- $\Gamma : g,\ \psi_g \rightarrow \tau_g$ is a grasp controller parameterized by grasp pose g, sampled from the initiation set, and grasp parameters ψ_g (e.g. grasp type, force), and outputting motor commands for the gripper τ_g.

For most tasks, we envision that all the skill components are given or designed except π_c and I_c, which leads to a problem of jointly learning a policy and affordance model for functional grasping. The CIP model structures the motor skill learning problem so that: only motor control involving contact with the object is learned, and free-space motion is generated using a planner; interaction with an object is always safe; and motion planning is used for the remainder of motor control, especially to stitch motor skills together. At the same time, the components that must be learned offer natural opportunities for incorporating powerful deep network methods to learn rich sensorimotor policies. The result will be small, isolated pockets of motor skill learning connected by much longer trajectories generated by a motion planner. Note that the CIP model does not assume access to a simulator or dynamics model of the environment.

A. Instantiating CIPs

One benefit of the CIP framework is that its different components may be chosen to match the robot hardware it is being instantiated on. We now detail our specific choices of component instantiations used in the experiments (described in Section IV) as an illustrative example.

Motor control policy π_c: Sensor input from the touch sensors on the robot's grippers, the joint and Cartesian state of the robot, and object joint state are fed into a neural network policy. For the action space, we chose to have the robot command the end-effector in Cartesian space while remaining compliant to promote sample-efficiency and safety during sustained contact. We therefore selected the Variable Impedance Control in End-Effector Space [47] scheme as our action space. Motor policy π_c maps sensor readings ϕ to a desired delta end-effector position $\Delta^{pos} = (p_d - p)$ and rotation $\Delta^{ori} = R_d \ominus R$, as well as commanded stiffness terms $k_p^p \in \mathbb{R}^{3\times3}$ and $k_p^R \in \mathbb{R}^{3\times3}$ for position and rotation respectively. These terms are then used to directly map to



(a) Open door task (b) Slide knob task (c) Open drawer task (d) Flip lever task

Fig. 2: Simulation Task Environments

joint torques τ via:

$$\tau = J_p[\Lambda_p[k_p^p(p_d - p) - k_d^p v]] \quad (1)$$
$$+ J_R[\Lambda_R[k_p^R(R_d \ominus R) - k_d^R \omega]], \quad (2)$$

where Λ_p and Λ_R are the position and orientation components of the inertia matrix $\Lambda \in \mathbb{R}^{6\times6}$ in the end-effector frame, J_p and J_R are the position and orientation components of the end-effector Jacobian J, and $R_d \ominus R$ corresponds to subtraction in $SO(3)$. Linear and angular velocity are denoted by v and ω, respectively; $k_d^p \in \mathbb{R}^{3\times3}$ and $k_d^R \in \mathbb{R}^{3\times3}$ are the damping matrices for position and rotation, set such that the system is critically damped. The resulting action space therefore consists of 13 dimensions: six for end-effector pose, six parameterizing the diagonal of the stiffness matrices, and one for the state of the parallel-jaw gripper.

Safety envelope σ: We limit the maximum value of stiffness parameters k_p^p and k_p^R, so that the robot remains compliant and does not generate high torque values when it contacts the object. In addition, the torques are clipped if they exceed the allowed range. We use a two-fold strategy to prevent joint limit violations, with two threshold parameters, σ_1 and σ_2 ($\sigma_1 > \sigma_2$), that check how close the robot joints are to its limits. If a joint position θ_i exceeds its threshold σ_1, we switch to a null-space controller [48] that attempts to move θ_i away from its limit without changing the end-effector pose. If the robot nonetheless exceeds σ_2 at joint index i (e.g. due to a high enough initial velocity to overcome the null-space controller), the controller generates a torque in the opposite direction for θ_i until it returns to a safe configuration.

Task-specific success indicator β_c: These were designed by hand for each task, and return true when the object's joint states are above a threshold position. In principle, they could be learned from data.

Task-general failure classifier B: In our case, B simply served as a joint limit safety check: if the robot is within 5 degrees of its joint limits, the classifier returns true and ends the learning episode. Episodes are also terminated if the agent loses contact with the object for several timesteps.

Grasp initiation set I_c: In each case, the visual data v is represented as a point cloud of the scene, which is segmented to only include the part of the object that the robot should manipulate. An existing task-general grasp generator `GPG` [49] is used to sample a set of grasp poses G based on the normals calculated from the point cloud. Each grasp $g \in G$ is then checked for reachability and collision. Grasps g which pass these checks are added to a list of acceptable grasp poses that define the domain of I_c.

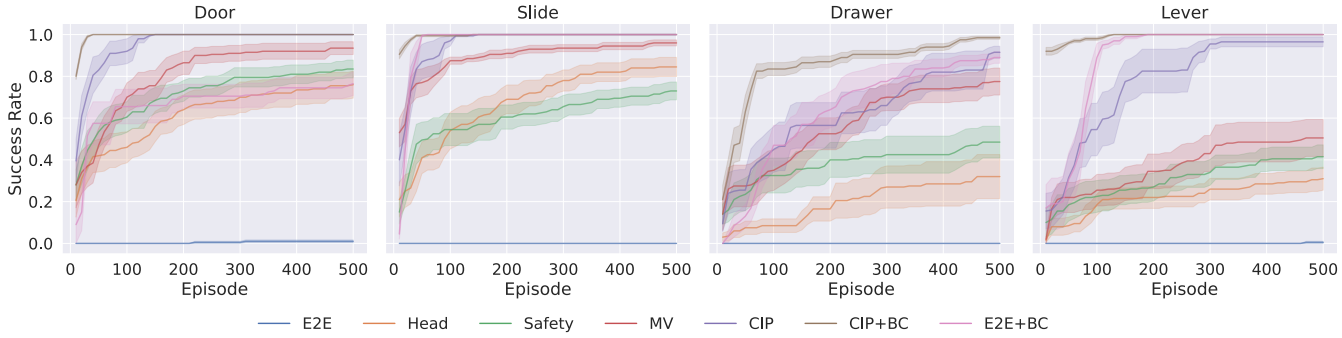
To sample a grasp pose $g \in I_c$ for the head h during learning, we cast grasp sampling as a bandit problem that is solved with Upper Confidence Bounds (UCB) [50] where Q-values are task success rates. We therefore treat learning I_c and sampling grasp poses as an active learning problem, and use UCB since it appropriately balances exploration and exploitation.

Once a grasp pose g is sampled, we obtain a suitable joint configuration θ by optimizing manipulability. A manipulability score is computed for a joint configuration θ as the product of two values: 1) the manipulability index introduced by [51] that analyses the volume of the manipulability ellipsoid: $w = \sqrt{det(JJ^T)}$ where J is the Jacobian for a particular joint configuration θ, and 2) a penalization term introduced by [52] based on the distance to the upper and lower joint limits for a particular joint configuration θ. These two metrics capture for a joint configuration θ how close the robot's end-effector is to a singularity and how close the robot's joints are to joint limits, respectively.

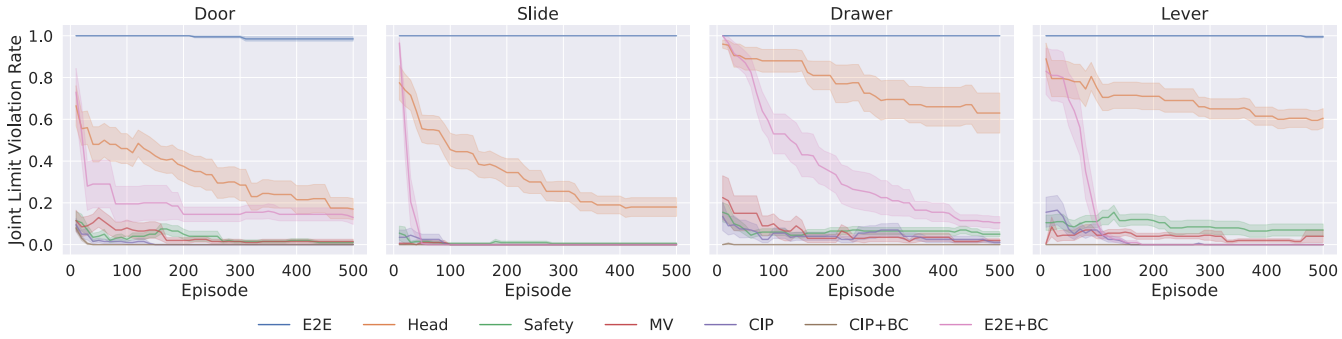
During learning, at the start of each episode, the manipulator is reset to the sampled joint configuration. The policy is executed and updated normally. The success or failure of the rollout is recorded and used to update the initiation set classifier using UCB. During task planning (sequencing skills after the skills and classifiers are trained) the highest probability grasp is selected for execution.

Motion planners h and t: These were instantiated for each domain using the IKFlow inverse kinematics solver [53] and MoveIt! [54] to move the robot to the grasp pose sampled from the grasp initiation set I_c.

Grasp Controller Γ: The implementation of the grasping controller depends heavily on the morphology of the gripper. For simple parallel jaw grippers the parameterization is simply the opening state of the gripper. For more complex



(a) Success rates for simulated tasks (Door, Slide, Drawer, Lever).



(b) Joint limit violation rates for simulated tasks (Door, Slide, Drawer, Lever).

Fig. 3: Task success rates and joint limit violation rates vs. the number of training episodes. The shaded region around the average shows the standard error over 10 seeds.

hands, e.g. with three or more fingers, the grasping controller may select among power and pinch grasps [55].

IV. EXPERIMENTS

We evaluate the CIP framework in simulation using ROBOSUITE [56]. We conducted experiments on four different articulated object tasks: opening a door, opening a drawer, sliding a knob, and lifting a lever. The position and orientation of the object in each episode is randomized over a small range as in the original benchmark.

The observations consist of the state of the object, position and velocities of the robot's joints, end-effector pose, and tactile readings from the force sensors at the robot's gripper. We trained policies using TD3 [57]. The reward functions are dense as a function of progress toward the object goal joint state, which leverages potential-based reward shaping [58] to ensure the optimal policy is not changed compared to the sparse reward setting based on success.

We consider two evaluation metrics: 1) **Task Success Rate**, and 2) **Joint Limit Violation Rate**, a proxy measure for how safe the policy is. To analyze how each of the structures of CIP impact these metrics, we run ablations that incrementally include structure described in Section III-A:

1) **E2E**: This setting is an end-to-end baseline that incorporates none of the CIP structure. The robot begins in a home pose with no contact to the object, and must learn a complete policy for moving to the object and manipulating it.

2) **Head**: This baseline incorporates the head h structure of the CIP, but no initiation set learning—the robot samples grasp poses randomly and uses naive IK to reach them.

3) **Safety**: This baseline extends **Head** to additionally incorporate the safety envelope.

4) **Manipulability Value (MV)**: This baseline extends the **Safety** setting, and additionally incorporates the manipulability value into the sampling approach for I_c. After sampling a random grasp, we sample a set of inverse kinematics solutions and select the one with the highest manipulability value.

5) **CIP**: Incorporates all the structure of the CIP; extends the **MV** approach to additionally perform active learning with UCB over grasp poses.

6) **CIP+BC**: Ten demonstrations from an expert policy are provided to the **CIP** learner and incorporated into policy search using the behavior cloning loss and Q-Filtering [25].

7) **E2E+BC**: Ten demonstrations from an expert policy are provided to the **E2E** learner.

Results: The results for all our experiments can be found in Figure 3, where we show the best-to-date performance for both metrics across all the tasks. Across all the tasks, the **E2E** baseline is unable to learn a meaningful policy, and has many joint violation rates throughout learning. This is expected as exploration is extremely challenging in the absence of a strong reward signal for reaching the object and making contact. We also see that once the head of the CIP is incorporated (**Head**), the agent is more performant,

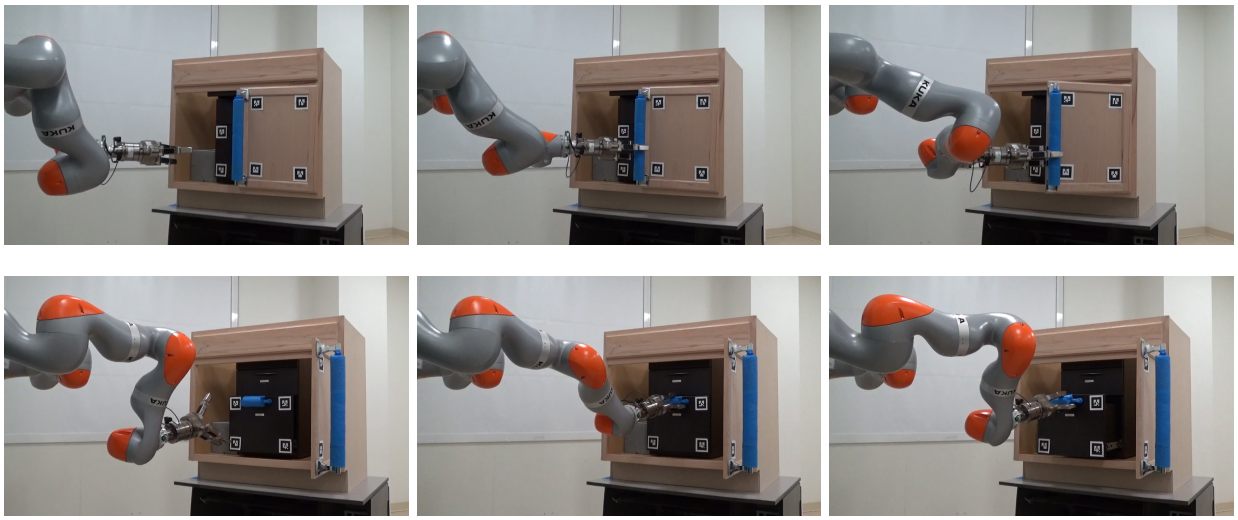


Fig. 4: Two learned CIPs executed in succession on robot hardware.

but still encounters many joint state violations throughout the learning process. The **Safety** baseline is able to achieve a task success rate on par with **Head**, but significantly reduces the joint state violation rates during learning. The **MV** baseline has improved task success over the **Head** baseline, which demonstrates the usefulness of incorporating the manipulability value when selecting joint configurations for sustained-contact manipulation tasks, but still has trouble learning an effective policy for the Lever and Drawer task in a small number of training episodes. Once the full structure of the CIP is incorporated (**CIP**), the agent is able to rapidly learn a policy with a high success rate (at least an average of 80%) within hundreds of training episodes. When demonstrations are available (**CIP+BC**) we see rapid, safe learning within tens of episodes. Note that **CIP+BC** outperforms the end-to-end agent with demonstrations (**E2E+BC**). We hypothesize that the Drawer task is challenging due to the relatively low manipulability of grasps on the object. These results demonstrate that each structural component of the CIP is useful for promoting safe and efficient learning across a diverse set of sustained-contact manipulation tasks.

V. SKILL COMPOSITION DEMONSTRATION ON HARDWARE

One of the advantages of the CIP structure is it enables zero-shot composition by construction. The motion planning performed via the head h and tail t, together with learned initiation sets I_c, enable a robot to learn sustained-contact manipulation skills in isolation, and then sequentially execute the skills with no additional learning. We validate our approach by learning and sequencing two manipulation skills—opening a cabinet door and pulling a drawer open—on a KUKA LBR iiwa7 with a Schunk Dexterous Hand 3-fingered gripper as shown in Figure 4. The sequence is determined a priori by an expert but could be computed using off-the-shelf task planning methods. For each skill, we produce a set of possible pinch grasps using a grasp pose generator [49] and filtering for collision, IK feasibility, and successful contact. Given a grasp proposed by the UCB sampler, we select an IK solution with high manipulability index as described in Section III-A. The observation space consists of the Cartesian position and orientation of the robot end effector, tactile readings from each of 6 touch sensors on the gripper's fingers, and the state of the door or drawer. The actions consist of displacements in position and are executed using the iiwa's Cartesian Impedance control mode. A shaped reward is provided as in the simulation experiments as a function of object state tracked using `ar_track_alvar`. We provide 3-5 demonstrations for each skill. Please refer to the video supplementary material for further detail. As shown in Figure 4, the robot is able to successfully learn each skill, and to compose the two skills in sequence.

VI. CONCLUSION

We propose a new policy class for sustained-contact manipulation skills: Composable Interaction Primitives (CIPs). CIPs are designed to exploit readily-accessible structure in the world and structure in the robot to enable sample-efficient and safe policy learning, and be easily leveraged by high-level planners due to their sequential composability via motion planning. Future work will investigate efficient methods to learn effect models to autonomously construct a symbolic vocabulary to support integrating CIPs with a high-level task planner, and learning CIPs from high-dimensional visual observations.

VII. ACKNOWLEDGEMENTS

This research was supported by NSF CAREER Award 1844960 to Konidaris, an Amazon Faculty Research Award to Konidaris, NSF Fellowship Award to Thompson, AFOSR DURIP FA9550-21-1-0308, and ONR contracts N00014-22-1-2592, N00014-21-1-2584. Partial funding for this work provided by The Boston Dynamics AI Institute ("The AI Institute"). Disclosure: George Konidaris is the Chief Scientific Advisor of Realtime Robotics, a robotics company that develops motion planning software.

REFERENCES

[1] O. Kroemer, S. Niekum, and G. Konidaris, "A review of robot learning for manipulation: Challenges, representations, and algorithms," *Journal of Machine Learning Research*, vol. 22, no. 30, pp. 1–82, 2021.

[2] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[3] J. Kober and J. Peters, "Policy search for motor primitives in robotics," *Machine Learning*, vol. 84, no. 1-2, pp. 171–203, 2010.

[4] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[5] A. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds., 2002, pp. 1547–1554.

[6] C.-A. Cheng, M. Mukadam, J. Issac, S. Birchfield, D. Fox, B. Boots, and N. Ratliff, "Rmpflow: A geometric framework for generation of multitask motion policies," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 968–987, 2021.

[7] N. Ratliff, J. Issac, D. Kappler, S. Birchfield, and D. Fox, "Riemannian motion policies," *arXiv preprint arXiv:1801.02854*, 2018.

[8] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to select and generalize striking movements in robot table tennis," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 263–279, 2013.

[9] R. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[10] T. Lozano-Perez, M. Mason, and R. Taylor, "Automatic synthesis of fine-motion strategies for robots," *International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.

[11] R. Burridge, A. Rizzi, and D. Koditschek, "Sequential composition of dynamically dextrous robot behaviors," *International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.

[12] R. Tedrake, "LQR-Trees: Feedback motion planning on sparse randomized trees." in *Robotics: Science and Systems V*, 2009, pp. 18–24.

[13] G. Konidaris and A. Barto, "Skill discovery in continuous reinforcement learning domains using skill chaining," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 1015–1023.

[14] O. Nachum, S. S. Gu, H. Lee, and S. Levine, "Data-efficient hierarchical reinforcement learning," *Advances in neural information processing systems*, vol. 31, 2018.

[15] A. Gupta, R. Mendonca, Y. Liu, P. Abbeel, and S. Levine, "Meta-reinforcement learning of structured exploration strategies," *Advances in neural information processing systems*, vol. 31, 2018.

[16] L. Huo, Z. Wang, M. Xu, and Y. Song, "Learning diverse sub-policies via a task-agnostic regularization on action distributions," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3932–3936.

[17] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.

[18] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1126–1135.

[19] F. Sadeghi and S. Levine, "CAD2RL: Real single-image flight without a single real image," in *Robotics: Science and Systems XIII*, 2016.

[20] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[21] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[22] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation," *arXiv preprint arXiv:1806.10293*, 2018.

[23] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[24] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, pp. 469–483, 2009.

[25] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 6292–6299.

[26] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive motion of animals and machines*. Springer, 2006, pp. 261–280.

[27] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual review of control, robotics, and autonomous systems*, vol. 3, pp. 297–330, 2020.

[28] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *ArXiv*, vol. abs/2102.03861, 2021.

[29] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in Neural Information Processing Systems*, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[30] M. Y. Seker, M. Imre, J. Piater, and E. Ugur, "Conditional neural movement primitives," in *Proceedings of Robotics: Science and Systems*, FreiburgimBreisgau, Germany, June 2019.

[31] M. Akbulut, E. Oztop, M. Y. Seker, H. X, A. Tekden, and E. Ugur, "Acnmp: Skill transfer and task extrapolation through learning from demonstration and reinforcement learning via representation sharing," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 1896–1907.

[32] M. T. Akbulut, U. Bozdogan, A. Tekden, and E. Ugur, "Reward conditioned neural movement primitives for population-based variational policy optimization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 10 808–10 814.

[33] S. Shaw, B. Abbatematteo, and G. Konidaris, "Rmps for safe impedance control in contact-rich manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2707–2713.

[34] M. Xie, K. Van Wyk, A. Handa, S. Tyree, D. Fox, H. Ravichandar, and N. D. Ratliff, "Neural geometric fabrics: Efficiently learning high-dimensional policies from demonstration," in *6th Annual Conference on Robot Learning*.

[35] S. Nasiriany, H. Liu, and Y. Zhu, "Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7477–7484.

[36] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Leveraging motion generation in reinforcement learning for mobile manipulation," *arXiv preprint arXiv:2008.07792*, 2020.

[37] J. Yamada, Y. Lee, G. Salhotra, K. Pertsch, M. Pflueger, G. Sukhatme, J. Lim, and P. Englert, "Motion planner augmented reinforcement learning for robot manipulation in obstructed environments," in *Conference on Robot Learning*. PMLR, 2021, pp. 589–603.

[38] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.

[39] Z. Wang, C. R. Garrett, L. P. Kaelbling, and T. Lozano-Pérez, "Learning compositional models of robot skills for task and motion planning," *The International Journal of Robotics Research*, vol. 40, no. 6-7, pp. 866–894, 2021.

[40] C. Agia, T. Migimatsu, J. Wu, and J. Bohg, "Stap: Sequencing task-agnostic policies," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7951–7958.

[41] D. Xu, A. Mandlekar, R. Martín-Martín, Y. Zhu, S. Savarese, and L. Fei-Fei, "Deep affordance foresight: Planning through what can be done in the future," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6206–6213.

[42] S. Cheng and D. Xu, "League: Guided skill learning and abstraction for long-horizon manipulation," *IEEE Robotics and Automation Letters*, 2023.

[43] T. Silver, A. Athalye, J. B. Tenenbaum, T. Lozano-Perez, and L. P. Kaelbling, "Learning neuro-symbolic skills for bilevel planning," *arXiv preprint arXiv:2206.10680*, 2022.

[44] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto, "Learning and generalization of complex tasks from unstructured demonstrations," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5239–5246.

[45] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on robotics*, vol. 30, no. 2, pp. 289–309, 2013.

[46] B. Abbatematteo, E. Rosen, S. Tellex, and G. Konidaris, "Bootstrapping motor skill learning with motion planning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4926–4933.

[47] R. Martín-Martín, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, "Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1010–1017.

[48] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robotics Autom.*, vol. 3, pp. 43–53, 1987.

[49] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[50] L. Kocsis and C. Szepesvári, "Bandit based Monte-Carlo planning," in *Proceedings of the 17th European Conference on Machine Learning*, 2006, pp. 282–293.

[51] T. Yoshikawa, "Manipulability of robotic mechanisms," *The international journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[52] M.-J. Tsai, *WORKSPACE GEOMETRIC CHARACTERIZATION AND MANIPULABILITY OF INDUSTRIAL ROBOTS (KINEMATICS)*. The Ohio State University, 1986.

[53] B. Ames, J. Morgan, and G. Konidaris, "Ikflow: Generating diverse inverse kinematics solutions," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7177–7184, 2022.

[54] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *Journal of Software Engineering for Robotics*, vol. 5, no. 1, p. 3–16, May 2014.

[55] M. Corsaro, S. Tellex, and G. Konidaris, "Learning to detect multimodal grasps for dexterous grasping in dense clutter," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4647–4653.

[56] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv preprint arXiv:2009.12293*, 2020.

[57] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*. PMLR, 2018, pp. 1587–1596.

[58] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Icml*, vol. 99, 1999, pp. 278–287.