Composable Causality in Semantic Robot Programming

Emily Sheetz, Xiaotong Chen, Zhen Zeng, Kaizhi Zheng, Qiuyu Shi,
Odest Chadwicke Jenkins

Statement of Contributions

This work is co-authored by several of my labmates. Co-author Zeng helped refine the problem formulation and methodology, helped write early drafts of the introduction, and proofread the paper. Co-author Chen helped write early drafts of the experiments section. Co-authors Chen, Zheng, and Shi defined grasp poses for each object part, helped implement the constrained random placement of objects in the scene, developed the automated grasping pipeline, implemented the failure recovery mechanisms in our pipeline, and tested the pipeline to identify issues. I defined the problem, implemented the control basis, determined our approach, implemented the *causal control basis* predictions, ran prediction trials, tested and modified the constrained random placement of objects, tested and modified the grasping and failure recovery mechanisms, ran experiments, and wrote the paper. I was responsible for the development, implementation, and testing of the core contribution of this work (the *causal control basis*) and the writing of the paper.

Composable Causality in Semantic Robot Programming

Emily Sheetz, Xiaotong Chen, Zhen Zeng, Kaizhi Zheng, Qiuyu Shi, and Odest Chadwicke Jenkins

***Abstract*— Assembly tasks are challenging for robot manipulation because the robot must reason over the composed effects of actions and execute multi-objective behaviors. Robots typically use pre-defined priorities provided by users to determine how to compose controller behaviors, but we want the robot to autonomously select these compositions based on their composed effects within the task. We present Composable Causality in Semantic Robot Programming to allow robots to reason over the composed effects of controllers when executing multi-objective actions and autonomously compose controllers without pre-defined priorities. Our proposed *causal control basis* combines controller behaviors with causal graphs indicating pre-conditions and effects of multi-objective actions and temporal graphs connecting high-level symbolic actions to controller behaviors. The robot uses the *causal control basis* to predict the transition probability of achieving the composed effects of a multi-objective action. The composed causality estimates are used to select which action to execute within the context of a furniture assembly task. We evaluate the robot's transition probability estimates in different furniture assembly trials in simulation on the Baxter robot. The robot's ability to assemble furniture using different multi-objective connection actions demonstrates the usefulness of the composed causality estimates from our *causal control basis*.**

I. INTRODUCTION

Within the robotics community, assembly tasks have become a domain of interest because of the unique challenges in reasoning over objects and executing complex behaviors in long-horizon tasks. Robots have difficulty assembling objects because they have to compose the effects of multiple behaviors and maintain these composed effects as they move on to the next step. We need a flexible way to program our robots to perform assembly tasks, and Semantic Robot Programming (SRP) [21] has emerged as an intuitive way to express task goals to robots. Within SRP, a user can declaratively program a robot to achieve a task by demonstrating desired goal scenes of the task. The robot can then infer goal conditions and reason over available objects and actions to reach the goal. Researchers have studied object affordances, executing single actions, and compositions of behaviors within single actions. We need to extend this work so robots can reason more about the objects themselves, plan how to compose effects on these objects during task execution, and overcome the challenges of assembly tasks.

Assembly tasks are challenging because robots need to determine when to enact controller behaviors, predict the composed effects of actions, and maintain the effects of these behaviors throughout task execution. Expressing actions as *object-centric controllers* allows robots to compose

This work was supported by a NASA Space Technology Graduate Research Opportunity.

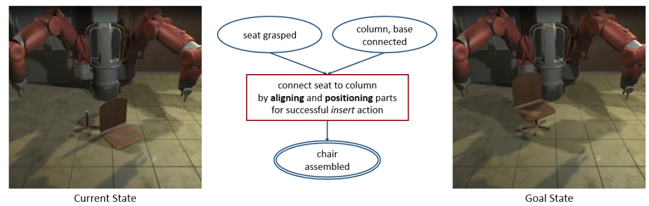


Fig. 1: A robot assembling furniture through goal-directed manipulation. Assembly tasks are challenging because the robot must reason over composed effects and maintain effects as it executes the task. In this case, the robot needs to connect the chair seat to the chair column, but needs to determine how to compose the *alignment* and *positioning* behaviors such that a successful connection is achieved. The robot will reason over the composable causality of controllers to assemble furniture pieces and achieve the task goal.

behaviors together to perform more complex multi-objective actions. However, the compositions of behaviors are generally determined by a pre-defined fixed priority of objectives provided by the user. To compose behaviors autonomously, robots need to reason over pre- and post-conditions of these behaviors by grounding them in the perceived scene, rather than in a symbolic manner that is disconnected from the realities of physical execution of the actions. Figure 1 shows an example of the challenges a robot faces during furniture assembly. To finish assembling the chair, the robot needs to properly *position* and *align* the seat relative to the chair column in order to connect the two parts together. The robot needs to determine how to compose these behaviors together so that the action is most likely to result in a successful connection and achieve the task goal of assembling the chair. Expressing user insights on controller compositions to robots remains an open question in the field. This is especially challenging because of the complexity of the causal relations over these controllers the robot would need to understand, specifically when controllers can be enacted, what controllers to compose, and what the effects of these behaviors will be.

We propose that notions of *causality* provide the insight needed to address the challenges of determining when controllers can be enacted and predicting the effects of controller compositions. Causality—the relationship between cause and effect—is used to determine the effects of interventions on observed distributions. In the case of goal-directed manipulation tasks, causality can allow robots to reason about the effects of composing controllers on the perceived objects and affordances. If the robot can predict the transition probability (or composed causality) of composed controller behaviors, then the robot can determine the compositions of controllers to execute within challenging long-horizon assembly tasks without relying on pre-defined priorities.

In this paper, we propose a *causal control basis* to build on SRP and provide robots with the power to autonomously compose controllers to achieve assembly tasks. The *causal control basis* will tell the robot the pre-conditions of enacting controller compositions, which controller behaviors are involved in a particular multi-objective connect action, and what sequences of behaviors correspond to high-level symbolic actions. Using the given *causal control basis*, the robot can estimate the transition probability that a particular controller composition will achieve the desired composed effects. This reasoning will allow the robot to autonomously compose controller behaviors without relying on pre-defined priorities of controller objectives. During task execution, the robot will select controller compositions based on their predicted composed effects and execute the afforded multi-objective actions to assemble a piece of furniture, as seen in Figure 1. We test our *causal control basis* and composed effect estimation in simulation on the Baxter robot, and find that the robot can autonomously compose and execute sequences of controllers to achieve a variety of furniture assembly tasks. Our work on **Composable Causality in Semantic Robot Programming** demonstrates that reasoning over the causality of a composable control basis makes the robot more capable of achieving challenging goal-directed manipulation tasks within the SRP paradigm.

II. RELATED WORK

A. Causality

Work on causality provides formal methods for analyzing cause and effect relationships between variables. Causal relationships can be expressed as Causal Bayesian networks (CBNs) [4] and analyzed through observational, interventional, and counterfactual queries [11], [12]. The do-calculus [13] focuses on answering queries about interventions on the variables in the environment. These intervention queries, in which an agent acts on and changes the variables in the environment, are of particular interest in the context of robot planning and manipulation.

Research involving causality has evolved to incorporate several different causal models, rather than a single CBN. For robot manipulation tasks in particular, the complexities of perception, planning action sequences, and action execution are best captured when reasoned over separately. Xiong et al. [20] found that hierarchical spatial, temporal, and causal concepts and the relationships between these concepts can be learned from demonstration, used to plan and execute cloth-folding tasks, and generalized to novel tasks. We take inspiration from the hierarchical temporal and causal models in Xiong et al. [20] and build on this work by proposing a *causal control basis* to allow robots to autonomously compose controllers in long-horizon tasks.

B. Object-Centric Controllers and Control Basis

Due to the significance of object affordances [5], robotics research has gravitated towards interacting with objects through *object-centric* motions. *Task frames* specify a coordinate frame attached to the manipulated object. By considering interactions with respect to the object and the motions that are performed on the object for the duration of the action, task frames serve as a bridge between the high-level symbolic description of actions and the low-level servomechanism execution of actions [1].

While motion planners successfully allow robots to plan how to move from one point to another, they do not allow the robot to move from one point to another in a specified manner, move while achieving concurrent motion goals, or react quickly to changes in the environment. Object-centric controllers, however, offer advantages such as being used to comprise a *control basis* that forms the building blocks of all behaviors the robot might need to execute and the ability to be *composed* through nullspace composition to yield multi-objective behaviors [14], [15]. Nullspace composition is particularly helpful when grasping objects to ensure that grasp closure is maintained relative to different end-effectors or gravity while the object is being moved through the workspace [16]. The control basis is also useful for conditioning behaviors such as avoiding joint limits or reactive behaviors such as avoiding obstacles [6]. Rohanimanesh et al. [17] explore running controllers in sequence or concurrently to achieve goals of different priorities and demonstrate the power of running controllers concurrently to achieve multiple goals. All of these works demonstrate the versatility of object-centric controllers. We build on these works by extending the composition of object-centric controllers to long-horizon tasks, where it is necessary to reason about sequencing the composed controllers.

Of particular interest is defining a control basis that can be used to perform tool-use tasks, since using tools typically involves performing multiple behaviors simultaneously. Sharma et al. [19] present a reinforcement learning approach to determining how controllers should be composed to perform different tasks. Their work emphasizes the importance of object-centric motions for tasks that involve using tools. We build on the idea of the robot autonomously determining compositions of object-centric controllers within tool-use tasks and extend it to long-horizon construction tasks using causal relations between controller behaviors.

C. Construction Tasks

We consider tool-use and construction tasks as an interesting domain for complex goal-directed manipulation. Tool-use and construction are challenging problems, as they require the ability to plan over long-horizons, understand properties of objects, and manipulate objects in particular ways. Nair et al. [9] investigate robots' abilities to construct tools to achieve task goals. They emphasize important insights into the use of tools—for example, that tools are typically comprised of a grasp part and an action part—and demonstrate that their tool construction pipeline effectively allows robots to construct tools with equivalent actions and effects as a canonical reference tool. We build on this work by testing the power of *causal* reasoning in assembly tasks, rather than the *geometric* reasoning they use to assemble tools. Lee et al. [7] developed the *IKEA Furniture Assembly Environment*, which serves as a testbed for the perception, planning, and control required to perform construction tasks. Though their assembly environment is designed for reinforcement learning, we use the IKEA Furniture Assembly Environment to allow the robot to predict the effects of a given control policy. Together, both of these works emphasize the immense interest and challenge of tool-use and construction tasks, which motivates our choice of assembly tasks as our problem domain.

III. METHODS

A. Problem Formulation

To perform assembly tasks that require multi-objective behaviors, the robot needs to predict the *transition function* of the controllers—the probability that a given composition of controllers will achieve their composed effects. Given a task goal, the robot can construct a high-level task plan of symbolic actions, each of which is decomposed into a sequence of executable motions. For actions that require multi-objective behaviors, we want the robot to determine how to compose the given controllers and execute the planned symbolic action. The robot will determine the correct controller composition by estimating the transition probability that the composition achieves the desired composed effects.

We assume we have a control basis Φ of controllers that can be composed using nullspace projection such that the composed controllers can achieve multiple objectives. The robot needs to reason over the *causality* of the controllers based on its predictions of the transition function. We formulate this probabilistic planning problem as a Markov Decision Process (MDP). An MDP is a tuple describing the state space, action space, transition function, and reward or cost function for a task. Policies for achieving goals in MDPs maximize cumulative expected reward or minimize cumulative expected cost. These policies can be found using iterative methods (value iteration and policy iteration) or learning methods (reinforcement learning) [8], [18]. In our work, the task planning is achieved using an off-the-shelf planner, but we determine the sequence of controllers used to execute this task plan using estimates of the transition function. The robot's estimation of the transition function will allow it to reason over the causality of the controllers, specifically by maintaining the pre-conditions and predicting the effects of the controllers.

Our multi-objective assembly task is expressed as the MDP (S, A, P, C). The state space S is the combined configuration space of the robot and configuration space of the objects in the scene. For robot configuration space Q and object configuration space $O = (SE(3))^N$ for N object parts in the scene, the state space is denoted by the Cartesian product $S = Q \times O$. The action space A is the set of all possible controllers and compositions in the given control basis Φ. The controllers that can be running at any given time are elements of the power set of the control basis $\mathcal{P}(\Phi) \setminus \varnothing$. Let M_t be the set of controllers that is running at time t. Controllers can be composed using nullspace projection. For example, suppose we have controllers ϕ_i and ϕ_j that achieve objectives i and j, respectively. One possible composition of these controllers is $\phi_j \lhd \phi_i$, which is read "controller j subject to controller i". This composition means that the command induced by controller ϕ_j is projected into the nullspace of controller ϕ_i to ensure that the progress made towards achieving objective j does not disrupt the progress made towards achieving objective i. All possible compositions of the running controllers M_t are elements of the symmetric group S_{M_t}, which is the set of permutations over M_t. In the example where $M_t = \{\phi_i, \phi_j\}$, the possible compositions are $S_{M_t} = \{\phi_j \lhd \phi_i, \phi_i \lhd \phi_j\}$. The action space for control basis Φ is $A = \{S_{M_t} \mid M_t \in \mathcal{P}(\Phi) \setminus \varnothing\}$. The transition probability $P(s' \mid s, a)$ indicates the probability of achieving the composed effects s' of a (composed) controller $a \in A$ when enacted in the current state s. The cost function $C_a(s)$ is the cost of enacting controller a in state s. As the controllers run, they meet their objective(s) by minimizing the value of their potential function ϕ. The robot wants to execute the controller a that will achieve its composed effects and minimize the cost of its objective function, meaning $C_a(s) = \phi_a(s)$.

The action space is determined by the causal control basis, but the robot is not given any information about the transition probabilities associated with the (composed) controllers. The robot needs to estimate the transition probability for each possible controller composition. The robot performs the transition probability estimation offline, before task execution. During task execution, the robot queries the *causal control basis* and selects the action with the maximum predicted transition probability, meaning it will execute the controller composition that is most likely to achieve its composed effects.

B. Causal Control Basis

We propose a *causal control basis* $\mathbf{\Phi}$ that the robot will use to predict the transition probabilities of actions and determine which composition of controllers to execute to achieve assembly tasks. The *causal control basis* is given to the robot and is comprised of:

- the implemented controllers in the control basis Φ;
- the set of composed causal graphs G_C, which represent the induced effects once the controller reaches its goal; and
- the set of temporal graphs G_T that represent the sequence of controllers that correspond to high-level symbolic actions.

Our *causal control basis* is denoted $\mathbf{\Phi} = (\Phi, G_C, G_T)$. The combination of causal and temporal information is inspired by the hierarchies of spatial, temporal, and causal concepts used for robot manipulation in previous works [20]. For the tasks considered in the experiments, we define the control basis—which will be discussed in greater detail in Section IV-A.1—to include pose, position, rotation, and screw controllers. However, our formulation will work with an arbitrary control basis.

The effects of the controllers, composed causality of the controllers, and temporal graphs for the high-level actions are all provided to the robot within the *causal control basis*. The components of the *causal control basis* tell the robot the sequence of controller behaviors associated with each high-level symbolic action, pre-conditions under which controller compositions can be enacted, and which controllers are involved in each multi-objective action. However, the robot does not know how to compose controllers to achieve the desired composed effects. For each possible composition a, the robot must use the control basis to estimate the transition probability $P(s' \mid s, a)$, which will allow the robot to predict if the composed controllers will result in successful assembly of the furniture piece.

We explored several possible approaches for predicting the transition probabilities of the controller compositions, which will be described in the next sections.

1) Possible Approach: Online Walkouts: The first possible approach is the robot performing online walkouts during task execution. When it comes time to execute a multi-objective connection action in the current state s, the robot will simulate the execution of every possible composition of controllers a to see which composition will converge and achieve the desired composed effects s'. The robot selects the composition that resulted in a successful simulated connection $P(s' \mid s, a) = 1$ and executes it.

The online walkout approach allows the robot to test the controller compositions in the context of the current task. However, performing online walkouts while the robot is executing the task slows down execution time, and users will likely not want to wait a significant amount of time while the robot decides what action to take. Online walkouts also restrict the transition probability predictions to the current task, and does not allow the robot to generalize these predictions across action instances.

2) Possible Approach: Offline Point-Wise Predictions: A second possible approach is to perform offline point-wise predictions before task execution. In order to reduce the time the robot takes to choose actions during task execution, the robot can use offline point-wise predictions to predict whether the controller composition will achieve its composed effects from a single controller update.

For example, suppose we have controllers ϕ_i and ϕ_j that achieve objectives i and j, respectively, and we want to estimate the transition probability of one possible composition of these controllers, $\phi_j \lhd \phi_i$. For a random initial state $s \in S$ and controller goals $s' \in S$, the robot can perform a single controller update and compute a point-wise prediction of whether composition $\phi_j \lhd \phi_i$ will achieve effects i and j:

$$\hat{P}(s' \mid s, a) = \frac{\|\mathcal{N}(\boldsymbol{J}_i)\Delta \boldsymbol{q}_j\|}{\|\Delta \boldsymbol{q}_j\|}$$

This estimation of the transition probability is based on how the projection of the command $\Delta \boldsymbol{q}_j$ from ϕ_j into the nullspace of ϕ_i—denoted $\mathcal{N}(\boldsymbol{J}_i)$—compares to the unprojected command from ϕ_j. For a large number of uniformly random samples $s \in S$, the average predicted transition probability estimates tell the robot which compositions achieve their composed effects across action instances.

Empirically, a point-wise prediction does provide reasonable estimates of the transition probability such that the action with the maximal predicted transition probability results in successful task execution. However, the point-wise prediction does not capture the full trajectory of the controller execution. A point-wise estimate does not capture if the controllers will reach a local minimum, which will prevent the robot from achieving the composed effects of the controllers.

3) Possible Approach: Offline Walkouts: Another possible approach is to perform offline walkouts. Before task execution, the robot uniformly samples initial states $s \in S$ and controller goals $s' \in S$ and simulates execution of the full trajectory of a controller composition to determine if the composition will achieve its composed effects s'. Based on the simulated trajectory, the robot computes the probability the composition achieved its composed effects in that action instance. Suppose the robot is predicting the transition probability for arbitrary controller composition $\phi_k \lhd \phi_j \lhd \phi_i$. The robot executes the composed controllers until they converge or until some large time threshold T. The predicted transition probability in this instance of executing $a = \phi_k \lhd \phi_j \lhd \phi_i$ is:

$$\hat{P}(s' \mid s, a) = \begin{cases} 1 & \text{objectives met} \\ 0 & \text{bad progress} \\ \frac{\phi_a(s) - \phi_a(s_T)}{\phi_a(s)} & \text{otherwise} \end{cases} \tag{1}$$

where s_T is the state at time threshold T, ϕ_a indicates the composed cost or objective function values at the given state, and bad progress means the controllers reached a local minimum. For a large number of random samples, the average predicted transition probability indicates how the controller composition performs across action instances.

Though performing the walkouts offline for a large number of random samples is time consuming, the online decision making during execution is quick. Offline walkouts allow the robot to consider the full trajectory of the controller execution when predicting the transition probability. This approach also allows the robot to generalize the transition probability predictions across action instances. For these reasons, our *causal control basis* uses offline walkouts to estimate the transition probabilities of controllers as in Equation 1. During task execution, the *causal control basis* indicates which controller composition to execute by selecting the composition with the greatest predicted transition probability. The predicted transition probabilities allow the robot to estimate the causality of the controllers and the likelihood that the controllers will achieve their composed effects.

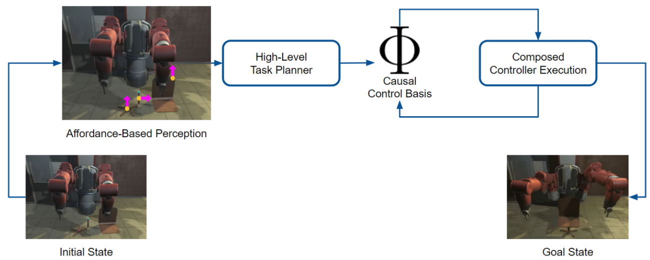


Fig. 2: Our pipeline for Composable Causality in Semantic Robot Programming. The robot perceives the objects and affordances in the initial scene using affordance-based perception, constructs a high-level task plan using an off-the-shelf task planner, converts the task plan into a sequence of controller compositions based on their predicted composed effects by the *causal control basis*, and executes the sequence of controller compositions to achieve the furniture assembly task.

IV. EXPERIMENTS AND RESULTS

Figure 2 describes the pipeline for assembling furniture using Composable Causality in Semantic Robot Programming and the use of our proposed *causal control basis*. We assume that the robot has parsed the goal conditions of the task as in SRP [21] and that we have affordance-based perception such as Affordance Coordinate Frames (ACFs) [2] to perceive the objects and affordances in the scene. These perceived objects and affordances seed the initial state of an off-the-shelf high-level task planner, in this case the Pyperplan STRIPS planning library[1]. Given the PDDL description of the high-level actions, the inferred task goals, and the perceived objects and affordances, the Pyperplan A* planner constructs the high-level task plan. The *causal control basis* converts each action in the high-level task plan into a sequence of (possibly composed) controller commands and instantiates the action based on the current poses of the objects and their connection sites. The robot executes this sequence of controllers to achieve the task goal of assembling furniture.

We evaluate the effectiveness of the proposed *causal control basis* in various furniture assembly tasks in simulation using the Baxter robot in the IKEA Furniture Assembly Environment[2] [7]. In the experiments, we assume known object poses during manipulation and grasp poses for every object part are provided. The connection of two parts is implemented as welding in the Mujoco simulation, which checks the position and axis alignment of connecting points. The connection points are shown in the simulator as colored dots, with sites of same color indicating where two parts are connected together.

A. Causal Control Basis for Furniture Assembly

1) Control Basis Implementation: In this work, we define the control basis Φ for furniture assembly by 6D pose, 3D position, rotation, and screw controllers. All of these controllers are object-centric controllers, meaning objectives are expressed in the object frame. They are implemented as potential field controllers, which minimize their potential function ϕ using gradient descent, and are based on attractive potential fields that attract the robot and objects to their goal. We define the following notation for each controller:

- 6D pose controller ϕ_{6Dpose} puts the object in a target pose $\boldsymbol{x}_{\text{goal}} \in SE(3)$. The state of the potential field is the current pose $\boldsymbol{x}$ of the object. The potential function encodes the Euclidean distance between the current and target poses, and the controller commands the robot based on the gradient:

$$\phi_{\text{6Dpose}}(\boldsymbol{x}) = \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}_{\text{goal}}\|^2$$
$$\nabla\phi_{\text{6Dpose}}(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{x}_{\text{goal}}\|$$

- 3D position controller ϕ_{pos} puts the object at a target position $\boldsymbol{p}_{\text{goal}} \in \mathbb{R}^3$. The state of the potential field is the current position $\boldsymbol{p}$ of the object. The potential function encodes the Euclidean distance between the current and target positions, and the controller commands the robot based on the gradient:

$$\phi_{\text{pos}}(\boldsymbol{p}) = \tfrac{1}{2}\|\boldsymbol{p} - \boldsymbol{p}_{\text{goal}}\|^2$$
$$\nabla\phi_{\text{pos}}(\boldsymbol{p}) = \|\boldsymbol{p} - \boldsymbol{p}_{\text{goal}}\|$$

- Rotation controller ϕ_{rot} puts the object at a target rotation expressed as a quaternion $q_{\text{goal}} \in \mathbb{H}$ where $\mathbb{H}$ is the Hamilton algebra over quaternions. This controller is used to modify the relative orientation between the object being acted on by the gripper and a target object. The state of the potential field is the current quaternion rotation q of the object. The potential function encodes the angle difference between the current and target quaternions, and the controller commands the robot based on the difference rotation quaternion:

$$\phi_{\text{rot}}(q) = 2\arccos\left(q \cdot q_{\text{goal}}\right)$$
$$\nabla\phi_{\text{rot}}(q) = q^{-1}q_{\text{goal}}$$

- Screw controller ϕ_{screw} rotates the wrist of the robot by a relative rotation θ to perform a screwing motion. From the initial wrist joint configuration $\boldsymbol{q}_{\text{init\_wrist}}$, the target wrist configuration is $\boldsymbol{q}_{\text{goal\_wrist}} = \boldsymbol{q}_{\text{init\_wrist}} + \theta$. The state of the potential field is the current configuration $\boldsymbol{q}_{\text{wrist}}$ of the wrist joint. The potential function encodes the squared difference between the current and target wrist configurations, and the controller commands the robot based on the gradient:

$$\phi_{\text{screw}}(\boldsymbol{q}_{\text{wrist}}) = \tfrac{1}{2}(\boldsymbol{q}_{\text{wrist}} - \boldsymbol{q}_{\text{goal\_wrist}})^2$$
$$\nabla\phi_{\text{screw}}(\boldsymbol{q}_{\text{wrist}}) = \boldsymbol{q}_{\text{wrist}} - \boldsymbol{q}_{\text{goal\_wrist}}$$

Our control basis Φ is the set of these controllers:

$$\Phi = \{\phi_{\text{6Dpose}}, \phi_{\text{pos}}, \phi_{\text{rot}}, \phi_{\text{screw}}\} \tag{2}$$

The robot will use this control basis to assemble different furniture pieces.

[1] https://github.com/aibasel/pyperplan
[2] https://clvrai.github.io/furniture/

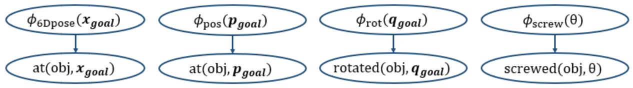


Fig. 3: The causal graphs for the single-objective controllers in the control basis for furniture assembly tasks. Once the controller reaches its goal and stops running, each controller induces an effect on the acted on object `obj` based on the given controller goal.

2) *Causal Graphs:* The set of causal graphs G_C indicate the effects of the controllers in the control basis. Figure 3 shows the causal graphs for the single-objective controllers in the control basis Φ. We consider two possible multi-objective connect actions, `insert` and `screw`. Figure 4 shows the causal graphs for these multi-objective actions, which indicate the controllers that are involved in these connect actions, the pre-conditions of enacting these compositions, and the intended effects of these compositions. The robot will use the transition probability predictions from the *causal control basis* to determine how to compose these controllers together, since the causal graphs only indicate which controllers are involved in the high-level connect actions.

3) *Temporal Graphs:* The set of temporal graphs G_T indicate the sequence of controllers that correspond to the high-level actions. For assembling furniture, we provide PDDL descriptions of high-level `pick-up`, `insert`, and `screw` actions. The temporal graphs in Figure 5 indicate the decomposition of these high-level actions into sequences of controller behaviors. Our decomposition of high-level actions into controller behaviors is not novel; for example, it is common practice to decompose a `pick-up` action into a pre-grasp, grasp, and post-grasp pose. However, for a multi-objective connection action `insert` or `screw`, the robot will have to determine what composition of the controllers—indicated in the corresponding causal graph in Figure 4—to execute within the sequence—indicated by the blue boxes in Figure 5. The decision for which composition is executed in the sequence is determined by selecting the composed controller with maximum predicted transition probability.

B. Composed Causality Predictions

For the `insert` and `screw` connection actions, the robot simulated 500 executions of each possible composition, for a total of 4000 offline walkouts to estimate the transition probabilities of both actions. Each simulated execution uniformly sampled a random start state $s \in S$ and a goal state $s' \in S$ and executed the given controller composition $a \in A$ until convergence or a large time threshold. We used time threshold $T = 300$ controller updates as the cutoff for the simulated execution. The predicted transition probability $\hat{P}$ was computed for each walkout as in Equation 1 and averaged across walkouts for the same composition to determine the estimated composed causality of the composition a.

The transition probability predictions for the `insert` and `screw` actions are shown in Table I and Table II, respectively. The composition with the highest probability

Composition a	Predicted Transition Probability $\hat{P}(s' \mid s, a)$
$\phi_{pos} \lhd \phi_{rot}$	**0.723**
$\phi_{rot} \lhd \phi_{pos}$	0.711

TABLE I: Transition probability predictions for `insert` action, based on 500 offline walkouts for each composition.

Composition a	Predicted Transition Probability $\hat{P}(s' \mid s, a)$
$\phi_{rot} \lhd \phi_{screw} \lhd \phi_{pos}$	**0.937**
$\phi_{pos} \lhd \phi_{screw} \lhd \phi_{rot}$	0.936
$\phi_{screw} \lhd \phi_{pos} \lhd \phi_{rot}$	0.929
$\phi_{pos} \lhd \phi_{rot} \lhd \phi_{screw}$	0.925
$\phi_{screw} \lhd \phi_{rot} \lhd \phi_{pos}$	0.923
$\phi_{rot} \lhd \phi_{pos} \lhd \phi_{screw}$	0.904

TABLE II: Transition probability predictions for `screw` action, based on 500 offline walkouts for each composition.

for the `insert` action indicates that positioning the object should be performed subject to aligning the object with the target, $\phi_{pos} \lhd \phi_{rot}$. This agrees with what a pre-defined priority might tell the robot, since two parts need to be properly aligned before we can position them together and perform the `insert` connection. The composition with the highest probability for the `screw` action indicates that aligning the object should be performed subject to screwing and positioning the object, $\phi_{rot} \lhd \phi_{screw} \lhd \phi_{pos}$. The position and screw controllers $\phi_{screw} \lhd \phi_{pos}$ create the spiraling motion for the `screw` action. The similar values for the transition probability estimates for the `screw` composition are due to the fact that the screw controller ϕ_{screw} only affects the wrist roll joint of the robot, and therefore is not likely to conflict with the other controller objectives.

C. Furniture Assembly Task Results

The composition with the maximum transition probability prediction is used to execute connect actions in a variety of furniture assembly tasks. This allows the robot to test the accuracy of the estimated composed causality of the controllers. We tested the `insert` action within 10 random trials of swivel chair assembly and tested the `screw` action within 6 random trials of table assembly. Across all trials, we compute the success rate of the multi-objective `insert` and `screw` actions as well as the success rate of the entire assembly task. We also record the average time used for high-level task planning, querying the *causal control basis* to select sequences of controller compositions and appropriate controller goals, and task execution. Summary information for the swivel chair trials is in Table III and select images from one of the random trials are in Figure 6. Summary information for the table trials is in Table IV and select images from one of the random trials are in Figure 7.

In Table III and Table IV, we see that the connect action success rates are very similar to their predicted transition probabilities in Table I and Table II, respectively. This indicates that the transition probability predictions accurately capture the performance of the compositions during task execution. When connect actions failed, it was largely due

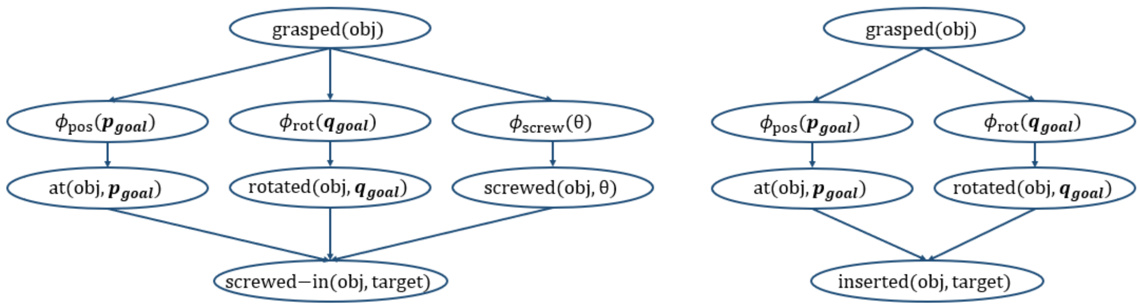


Fig. 4: The causal graphs for the multi-objective connect actions in the control basis for furniture assembly tasks. When assembling furniture, the robot needs to attach the acted on object part `obj` to a target object part `target`. These connections are either achieved by `insert` or `screw` actions. These causal graphs indicate what behaviors are necessary to connect objects together, the pre-conditions of enacting these compositions, and the composed effects of these compositions.

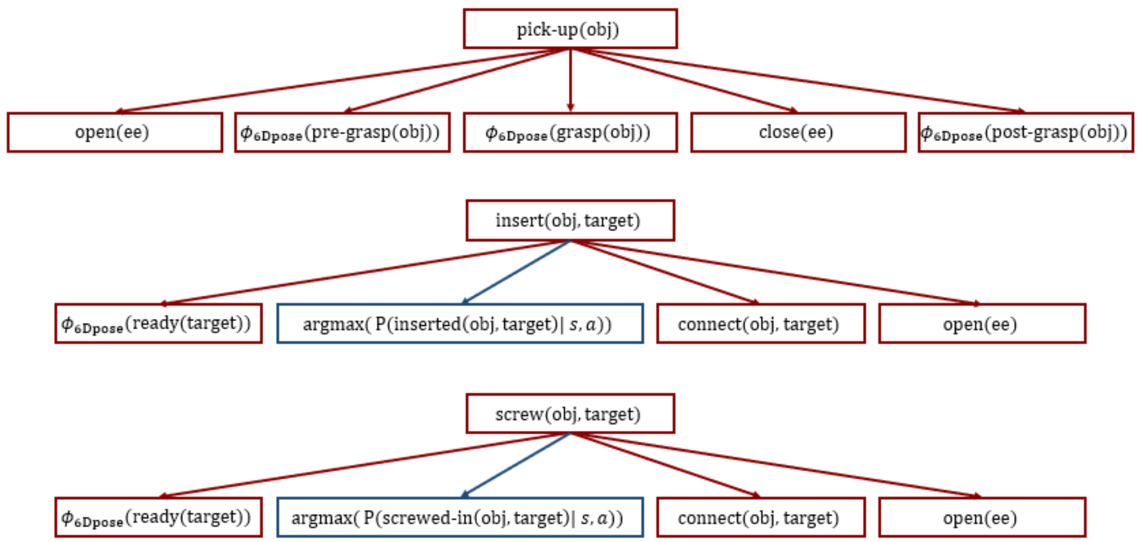


Fig. 5: The temporal graphs for the high-level symbolic actions. To pick up an object `obj`, the high-level symbolic action `pick-up(obj)` is executed by the sequence of low-level actions: open the gripper, move the end-effector to the pre-grasp pose, move the end-effector to the grasp pose, close the gripper around the object, and move the end-effector to the post-grasp pose. To insert or screw an object `obj` into a target object `target`, the high-level symbolic action is executed by the sequence of low-level actions: move the end-effector to the ready pose determined by target object `target`, perform the multi-objective controller composition that is most likely to achieve the composed effects, connect the two object parts, and open the gripper. Since the composed causality diagram in Figure 4 only indicates what behaviors should be used to achieve each type of connection, the *causal control basis* will need to determine how to compose these behaviors and substitute the properly composed and instantiated controller in the blue box in the temporal graph.

`Insert` Action Success Rate	0.714
Swivel Chair Assembly Task Success Rate	1
Average High-Level Task Planning Time (s)	0.028
Average Controller Selection/Instantiation Time (s)	0.205
Average Execution Time (s)	266.241

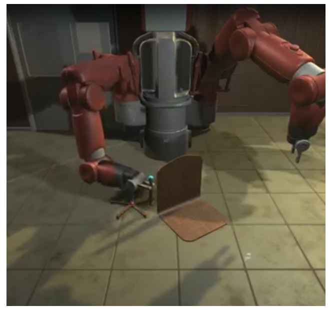
TABLE III: Results from 10 swivel chair assembly tasks, which require `insert` actions.

`Screw` Action Success Rate	0.923
Table Assembly Task Success Rate	1
Average High-Level Task Planning Time (s)	0.048
Average Controller Selection/Instantiation Time (s)	0.074
Average Execution Time (s)	492.072

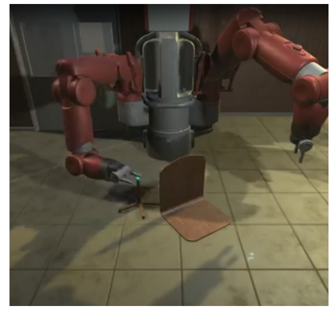
TABLE IV: Results from 6 table assembly tasks, which require `screw` actions.

to joint limits being reached or collisions between objects, both of which impeded the controllers from achieving their composed effects. However, the task success rate for both the swivel chair and table tasks indicate that when the compositions for the connect actions did not result in successful connections, the robot could recover by retrying the action

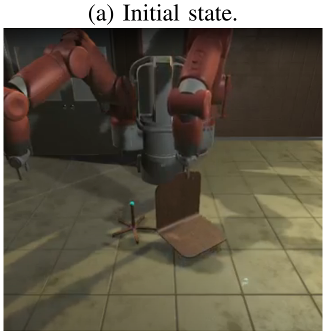
(a) Initial state.



(b) Pick up chair column.

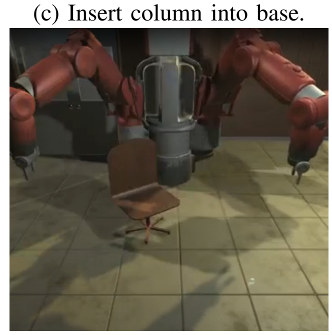


(c) Insert column into base.



(d) Pick up chair seat.

(e) Insert seat into column.



(f) Goal state.

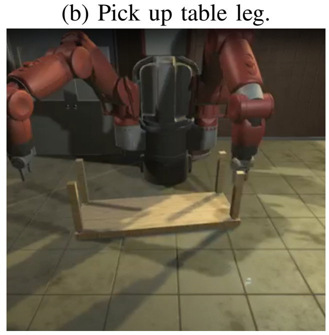
Fig. 6: Execution of swivel chair assembly task using `insert` actions to connect parts together.

(a) Initial state.

(b) Pick up table leg.

(c) Screw leg into table top.

(d) Pick up table leg.



(e) Screw leg into table top.

(f) Goal state.

Fig. 7: Execution of table assembly task using `screw` actions to connect parts together.

and ultimately achieve a successful connection.

The time spent on task planning, selecting and instantiating controllers, and executing the full task demonstrate the advantage of our offline walkout approach to predicting the transition probabilities. Though performing the walkouts was time consuming, considering the full trajectory of the controller execution allowed the robot to accurately estimate the composed causality of the controllers. This also meant that during execution, the time spent selecting and instantiating controller compositions for the given connect action—which amounts to querying the *causal control basis* and setting controller goals based on the poses of objects and their connection sites—was very small compared to the execution time. Performing walkouts offline before task execution made it more straightforward for the robot to make decisions about which controllers to execute during execution.

V. Discussion and Conclusion

The robot's ability to successfully assemble different furniture pieces demonstrates the accuracy of our proposed *causal control basis* in predicting the composed effects of controller behaviors. The results of our transition probability predictions and furniture assembly experiments indicate that the *causal control basis* effectively estimates the composed causality of controller behaviors such that these compositions can be successfully executed to achieve assembly tasks.

There are several areas of future work that would improve our proposed *causal control basis*. Instances of action failures were due to the controllers reaching local minima, the robot reaching joint limits, or collisions between objects. Future work would add additional controller behaviors for avoiding collisions and add mechanisms for avoiding or getting out of local minima and joint limits. Another area for future work is extending our *causal control basis* to coordinate the arms for bimanual manipulation tasks. Our experiments in this work focused on estimating the composed causality of single multi-objective actions within a long-horizon task, but this could be extended to considering the composed causality of multiple actions. Composing controllers across multiple actions would require bimanual manipulation. For example, the robot may need to hold one object part in place while it connects another object part. If at any point the stationary part drifts out of place, the robot would need to correct this before continuing to connect the other part. In cases where composed causality estimates extend across multiple actions, the arms would need to be coordinated to achieve multiple connections simultaneously. Exploring the capabilities of the *causal control basis* to include bimanual manipulation is a compelling future direction of research.

In this work, we proposed a *causal control basis* for achieving Composable Causality in Semantic Robot Programming. Our *causal control basis* allows the robot to predict the transition probabilities of controller compositions, thereby estimating the composed causality of multi-objective actions. The robot accurately estimated the composed causality of controllers in a variety of furniture assembly trials, as indicated by the successful execution of multi-objective connect actions and successful assembly of different furniture pieces across a number of random trials. Our work in Composable Causality in Semantic Robot Programming demonstrates that reasoning over a *causal control basis* allows the robot to autonomously compose controller behaviors without pre-defined priorities to achieve furniture assembly tasks.

References

[1] D. H. Ballard, "Task Frames in Robot Manipulation," pp. 16-22, 1983.

[2] X. Chen, K. Zheng, Z. Zeng, S. Basu, J. Cooney, J. Pavlasek, and O. C. Jenkins, "Manipulation-Oriented Object Perception in Clutter through Affordance Coordinate Frames," *arXiv preprint arXiv:2101.08202*, 2020.

[3] F.-J. Chu, R. Xu, C. Tang, and P. A. Vela, "Recognizing Object Affordances to Support Scene Reasoning for Manipulation Tasks," *arXiv preprint arXiv:1909.05770*, 2019.

[4] I. Dasgupta, J. Wang, S. Chiappa, J. Mitrovic, P. Ortega, D. Raposo, E. Hughes, P. Battaglia, M. Botvinick, and Z. Kurth-Nelson, "Causal Reasoning from Meta-Reinforcement Learning," *arXiv preprint arXiv:1901.08162*, 2019.

[5] J. J. Gibson, "The Theory of Affordances," pp. 67-82, 1977.

[6] S. Hart and R. Grupen, "Natural Task Decomposition with Intrinsic Potential Fields," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 2507–2512.

[7] Y. Lee, E. S. Hu, Z. Yang, A. Yin, and J. J. Lim, "IKEA Furniture Assembly Environment for Long-Horizon Complex Manipulation Tasks," *arXiv preprint arXiv:1911.07246*, 2019.

[8] M. L. Litman, T. L. Dean, and L. P. Kaelbling, "On the Complexity of Solving Markov Decision Problems," *arXiv preprint arXiv:1302.4971*, 2013.

[9] L. Nair, J. Balloch, and S. Chernova, "Tool Macgyvering: Tool Construction Using Geometric Reasoning," *International Conference on Robotics and Automation (ICRA)*, pp. 5837–5843, 2019.

[10] J. Pavlasek, S. Lewis, K. Desingh, and O. C. Jenkins, "Parts-Based Articulated Object Localization in Clutter Using Belief Propagation," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[11] J. Pearl, "Causal Diagrams for Empirical Research," *Biometrika*, vol. 82, no. 4, pp. 669–688, 1995.

[12] J. Pearl, *Causality*. Cambridge University Press, 2009.

[13] J. Pearl, "The Do-Calculus Revisited," *arXiv preprint arXiv:1210.4852*, 2012.

[14] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Manipulation Gaits: Sequences of Grasp Control Tasks," *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, pp. 801–806, 2004.

[15] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Nullspace Composition of Control Laws for Grasping," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[16] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Null-Space Grasp Control: Theory and Experiments," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 282–295, 2010.

[17] K. Rohanimanesh, R. Platt, S. Mahadevan, and R. Grupen, "Coarticulation in Markov Decision Processes," *Advances in Neural Information Processing Systems*, 2005.

[18] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, Third Edition*. Pearson Education, 2010.

[19] M. Sharma, J. Liang, J. Zhao, A. LaGrassa, and O. Kroemer, "Learning to Compose Hierarchical Object-Centric Controllers for Robotic Manipulation," *arXiv preprint arXiv:2011.04627*, 2020.

[20] C. Xiong, N. Shukla, W. Xiong, and S.-C. Zhu, "Robot Learning with a Spatial, Temporal, and Causal And-Or Graph," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2144–2151, 2016.

[21] Z. Zeng, Z. Zhou, Z. Sui, and O. C. Jenkins, "Semantic Robot Programming for Goal-Directed Manipulation in Cluttered Scenes," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7462–7469, 2018.

[22] Z. Zeng, "Semantic Robot Programming for Taskable Goal-Directed Manipulation," Ph.D. dissertation, University of Michigan, 2020.