SCALE: Causal Learning and Discovery of Robot Manipulation Skills using Simulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: We propose SCALE, an approach for discovering and learning distinct robot skills for a given task based on the context variables that are relevant to each skill. Our approach frames simulation as a causal reasoning engine, where interventions elicit the dependence between solving a manipulation task and the context variables. To incorporate this causal relationship, we model each robot skill as a Regional Compressed Option, which extends the options framework by associating a causal process and its relevant variables with the option. There exist multiple distinct causal processes for a task, each of which is valid only in a region of the context space. We call this region the Data Generating Region (DGR) of an option and learn it via interventions in simulation. The DGR is used to guide policy learning by training the policy only on states within its DGR. Each skill can be viewed as a distinct behavior that emerges from its underlying causal process. We demonstrate our approach for two representative manipulation tasks: block stacking and peg-in-hole insertion under uncertainty. Our experiments show that our approach yields diverse skills that are compact, robust to domain shifts and suitable for sim-to-real transfer.

Keywords: skill discovery, causal learning, manipulation

1 Introduction

We want robots to help and work alongside humans in their homes, kitchens and restaurants. However, outside of structured environments, robots currently struggle at reliably performing even some of the basic manipulation tasks that humans can do with ease. Why are humans so much better despite the vast diversity of objects and their complex interactions that they potentially need to reason about? First, humans usually know multiple ways to solve a task to be robust to failures and variations in the environment. For example, if a tight jar doesn't open with our bare hands, we may use a piece of cloth to improve our grip. Second, humans excel at selectively attending [1] to only a small part of the environment that is relevant to the task. Selective attention significantly reduces the computational complexity of reasoning and allows them to handle complicated situations.

Prior works in manipulation skill learning have leveraged these two observations separately. Most methods [2, 3, 4, 5, 6] learn skills by associating each skill with a sub-goal, where, the sub-goals are hand-designed or learned from demonstrations. Once the sub-goals have been assigned, feature selection [7, 8] and abstraction selection [9, 10] can be used to reduce the complexity of skill learning. However, such approaches are quite sensitive to the sub-goals and struggle to distinguish between different strategies to achieve the same goal. Our main insight is to *associate a skill with not just a sub-goal, but also with the variables that are causally relevant to it*. For example, opening a jar with our bare hands is a skill distinct from opening it with the help of a piece of cloth. Only *hand* and *jar* are relevant to the former, while the latter also relies on the properties of *cloth*. Hence, these two strategies should be represented as two distinct skills even though they achieve the same goal.

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

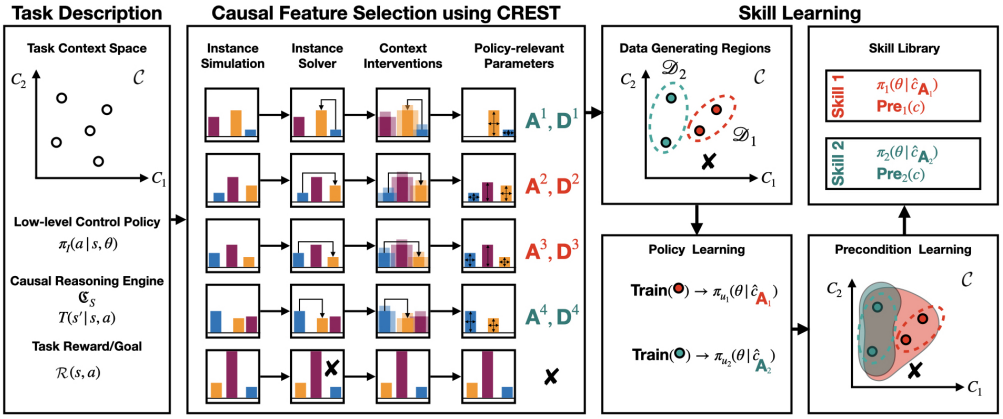


Figure 1: The figure shows an overview of the proposed framework. The robot is given a context space, control policy, task simulator, and task reward. The robot then samples a set of contexts to create task instances, which it subsequently solves for that instance. The robot then applies interventions on the context to identify skill-relevant parameters. Contexts with the same set of policy-relevant parameters are combined to form data generation regions. Each region is then used to learn a skill policy with the corresponding set of policy-relevant parameters. For each skill we subsequently learn a set of preconditions within the space of the precondition-relevant context parameters to determine where the skill can ultimately be applied. The pairs of policies and preconditions are then combined to create a skill library for completing the given task.

Based on this principle, a manipulation task involving n variables can have up to 2^n skills, each corresponding to a subset of the variables being causally relevant. This is a very large space to search for skills and not all subsets may correspond to a useful skill. Hence, we propose SCALE (**S**kills from **CA**usal **LE**arning), an efficient approach for robot skill learning through causal feature selection in simulation (Fig. 1).[1] Instead of naïvely generating data, in our approach, the robot *interacts* with the simulator by conducting causal interventions. This elicits the causal features for completing a task under different settings, yielding a diverse and compact library of skills. Our approach learns skills that are described by physically meaningful properties, e.g., in the door opening task, factors such as door position would be included, but spurious variables relating to irrelevant objects are excluded. Our simulator can be viewed as a *causal reasoning engine*: an interactive resource that the robot leverages for causal reasoning, leading to richer learning outcomes. Our contributions of this work are as follows:

- We introduce SCALE, an algorithm for learning a robot skill library from causal interventions in simulation.
- We conduct experiments that demonstrate that SCALE outperforms baseline approaches for two manipulation tasks of block stacking and sensorless peg insertion, in addition to a downstream task of creating a block tower. Moreover, we also demonstrate sim-to-real transfer of the skills learned by SCALE for block stacking.

2 Related Work

Robot skill learning. Building robots that can solve a wide variety of complex tasks is one of the fundamental problems in robotics. A popular approach is to learn skills parameterized by the task parameters as these can generalize over related tasks. Prior works [11, 12] show such parameterized skills lie on a low dimensional piecewise-smooth manifold in the context space and identify this structure using ISOMAP [13]. For higher-dimensional problems, it becomes infeasible to learn directly in the full context space. One approach is to learn a library of simple parameterized skills

[1] Website: https://sites.google.com/view/scale-causal-learn-robot-skill

which can be composed to solve more complex tasks [14, 15, 16, 17]. Recent works [18, 19] propose a differentiable attention mechanism to learn context-specific attention but these have been evaluated only in relatively small domains. Popular methods for unsupervised skill discovery include graph-based methods [20, 21] that seek to build a graph of skills to cover the task space and information-theoretic methods [22, 23, 6] that seek to maximize the diversity of skills.

Causality in robotics and reinforcement learning. Causality is the science of cause and effect [24, 25, 26]. Although the advantages of causal inference and discovery within the biomedical sciences, economics, and genomics have been well-established [27, 28], the integration of causality within machine learning is nascent [29, 30]. In robotics, causality-based approaches are particularly under-explored despite the potential advantages of greater reasoning and learning capabilities [31], particularly through structure and transfer learning [32]. Most similar to our work is that of CREST [33], an algorithm for identifying features for a robot policy through causal interventions. Our algorithm SCALE leverages the work of Lee et al. on CREST for determining the causally relevant variables for each robot skill. Causality has also empowered learning the structure of physical systems from videos [34] and explanations for robot failures [35]. Within reinforcement learning more broadly, causality plays a central role for improving performance through greater structure [36], learning latent factors in dynamics via causal curiosity [37], learning invariant policies [38], and learning a dynamics model that can yield state abstractions [39].

Intuitive physics. Intuitive physics is the ability to approximately predict and model the physical world without explicit understanding of the underlying dynamics [40]. Literature in cognitive psychology has suggested that humans develop mental intuitive physics models to support fast prediction and understanding of complex physical scenes which enables physical reasoning [41]. Computational learning of intuitive physics have been successful, enabling reinforcement learning and planning applications owing to the models ability for forward prediction [42, 43, 44]. In our work, our causal reasoning engine can be viewed as an internal model that uses interventions to elicit the physical mechanisms by which the data arise.

3 Preliminaries

3.1 Task Formulation

The robot learns a set of skills $\boldsymbol{\mathcal{K}} = \{\mathcal{K}_1, \ldots, \mathcal{K}_K\}$, where each skill solves a distribution of manipulation tasks. Each task is modeled as a *manipulation MDP* [45], $M := (\mathcal{S}, \mathcal{A}, \mathcal{R}, T, \gamma, \tau)$, where $s \in \mathcal{S}$ is the state space, $a \in \mathcal{A}$ is the action space, $\mathcal{R}$ is the reward function, T is the transition function, γ is the discount factor, and τ is additional task information. Tasks have a reward function of the form $\mathcal{R}(s, a) = \alpha\mathcal{R}_G(s) - \mathcal{R}_C(s, a)$, where $\mathcal{R}_G = \mathbb{1}\{s_f \in \mathcal{S}_G\}$ is a goal bonus provided when the final state s_f lies within the goal space $\mathcal{S}_G \subseteq \mathcal{S}$. Task costs, such as path traversal, are specified by $\mathcal{R}_C$. Tasks are solved if the final reward $R_f > R_S$, where R_S is a solved threshold.

Options. Each skill $\mathcal{K}$ is a parameterized option [46, 10]. An option $\mathcal{O} := (\pi, \mathcal{I}, \beta)$ is defined using three components: (1) the option control policy $\pi(a|s)$; (2) an initiation set $\mathcal{I} = \{R_f > R_S|s\}$ that specifies where the final reward R_f solves the task when taking option $\mathcal{O}$ in state s using option policy π; and (3) termination condition $\beta(s)$ that specifies when the option concludes. For the termination condition of this work, our skills execute open-loop with fixed duration.

Context. We define the context $c \in \mathcal{C}$ as a set of variables $\mathbf{C}$ that fully specify the manipulation task. The context space $\mathcal{C} := \mathcal{S} \times \tau$ generalizes the state space to include geometric and other time-invariant task properties defined by τ. Each skill requires only a subset of the full context, determined via causal feature selection (c.f., Sec. 5.1).

Contextual policies. We use a hierarchical approach [47] to decompose the option control policy into an *upper-level* policy $\pi_u(\theta|c)$ and a *lower-level* policy $\pi_l(a|s, \theta)$. Given a context $c \in \mathcal{C}$, $\pi_u : c \mapsto \theta$ specifies the parameters θ for the lower-level policy π_l. For example, π_l could be a Cartesian-space impedance controller, where θ specifies the sequence of waypoints to be followed by the robot end-effector.

For our work, we assume the lower-level controller π_l is given, and we learn the upper-level policy π_{uk}. The option policy therefore has the form $\pi_k(a|c) = \int \pi_l(a|s,\theta)\pi_{uk}(\theta|c)\mathrm{d}\theta$. The lower-level controller is shared across the different skills.

3.2 Compressed Context and Feature Selection

SCALE learns *compressed* skills that only use causally relevant context variables, as many will be unimportant. For our work, we disregard dimensions of the context space that are not chosen by causal feature selection (c.f., Sec. 5.1). To bridge compressed and full context space, we use a dimensionality reduction matrix $D \in \mathbb{1}^{d\times|\mathcal{C}|}$, which is a rectangular selection matrix whose elements are either 1 indicating the dimension is kept or 0 otherwise. A compressed context is expressed by $\hat{c} = Dc$, where the matrix reduces to a smaller space of dimension $d < |\mathcal{C}|$.

3.3 Causal Reasoning in Simulation

SCALE leverages a simulator with the key capability of interacting with scenes through context interventions, which enables the causal learning in SCALE. For this reason, we formalize the simulator as a *causal reasoning engine* $\mathcal{W} := (\mathfrak{C}_S, T)$, where $\mathfrak{C}_S$ is the scene structural causal model (SCM) and T is the transition model. App. B provides greater discussion of this formalization.

4 Skill Formulation

4.1 Regional Compressed Option

In our work, we formalize each robot skill $\mathcal{K}$ as a *Regional Compressed Option* (RCO), where $\mathcal{K} := (\pi_k, \text{Pre}, \beta, \mathscr{D})$ and π_k is the option control *policy*, Pre is the *precondition*, β is the *termination condition*, and $\mathscr{D}$ is the *data generating region* (DGR).[2]

In this model, the policy $\pi_k(a|\hat{c}_{\mathbf{A}_k})$ uses compressed context $\hat{c}_{\mathbf{A}_k}$. The learned, upper-level policy is $\pi_{uk}(\theta|\hat{c}_{\mathbf{A}_k})$, where $\hat{c}_{\mathbf{A}_k} = D_{\mathbf{A}_k}c$ and $D_{\mathbf{A}_k}$ is the compression matrix. When referring to the context dimensions used for the policy π_{uk}, we refer to them as the set $\mathbf{A}_k$. The precondition $\text{Pre}(c) = P(R_f > R_S|c)$ is a probabilistic initiation set [48].

4.2 Data Generating Region

The DGR $\mathscr{D} = \{c : \hat{c}_{\mathbf{D}_k} \in \hat{\mathcal{I}}\}$ is a space that captures the compressed initiation set $\hat{\mathcal{I}}$, which is the initiation set $\mathcal{I}$ with unimportant dimensions removed. The DGR compressed context is determined through $\hat{c}_{\mathbf{D}_k} = D_{\mathbf{D}_k}c$. When referring to the context dimensions of the DGR, we refer to them as the set $\mathbf{D}_k$. If a context c lies within the DGR, then $\mathscr{D}(c) = 1$ if $D_{\mathbf{D}_k}c \in \hat{\mathcal{I}}$ and 0 otherwise.

Data generating region: two views. The DGR, a space that represents where a skill is valid, has two equivalent views. In the options view, the DGR approximates the initiation set of the skill. In the causality view, the DGR encodes where the causal process exists of the robot executing the skill. In both views, the DGR provides a region of training data that is used to learn the skill policy. In our work, the DGR is learned through sample-based causal interventions. For greater intuition, App. E presents a toy experiment that provides DGR visualizations.

5 Self-Supervised Skill Discovery in Simulation

The SCALE algorithm comprises two steps: 1) skill dataset generation and 2) skill training. These steps are described in Sec. 5.1 and 5.2, respectively. Algorithm descriptions can be found in App. D.

[2]Named because it is the region that contains the data generating process (i.e., causal process) of the robot executing the skill to optimally solve the task.

5.1 Batch Data Generation

First, the robot interacts with the simulator $\mathcal{W}$ to collect skill training data. This is done by collecting a batch dataset $\mathcal{D}_B$. The robot samples n random scenes represented by c_i and attempts to determine the lower-level controller parameters θ_i that solve the specific task. In practice, we use Relative Entropy Policy Search (REPS) [49], but any suitable planner, trajectory optimizer, or reinforcement learning algorithm would suffice. Unsolved tasks are disregarded and not collected in $\mathcal{D}_B$.

Causal feature selection. For successfully solved scenes, the relevant variables for the policy $\mathbf{A}_i$ and DGR $\mathbf{D}_i$ are selected using the CREST algorithm [33]. CREST conducts feature selection through causal interventions. Intuitively, a variable is causal if, for all other variables held equal, interventions upon this variable induce a change in the final obtained reward R_f. A spurious variable has no effect on the reward and thus can safely be ignored. CREST performs interventions $\mathbf{I}$ over a local (e.g., 10%) fraction of context space $\mathcal{C}$ to yield $\mathbf{A}_i$. Similarly, $\mathbf{D}_i$ is obtained through interventions over the entire space $\mathcal{C}$. The compression matrices $D_{\mathbf{A}_i}$ and $D_{\mathbf{D}_i}$ are then recovered from $\mathbf{A}_i$ and $\mathbf{D}_i$, respectively. Finally, the batch dataset is appended by the dataset point $(c_i, \theta_i, D_{\mathbf{A}_i}, D_{\mathbf{D}_i})$.

Splitting batch data into skill data. After dataset collection, batch dataset $\mathcal{D}_B$ is split into different skill datasets according to the relevent variable sets. In this work, we assign highly occurring batch data that contain the same matrices $D_{\mathbf{A}}$ into the same skill dataset $\mathcal{D}_k$, while also taking the union over all associated $D_{\mathbf{D}}$. This assumption may not always hold, but is sufficient for the tasks we examine in this work. More sophisticated ways of splitting the batch dataset is left for future work.

5.2 Skill Training

The second phase of SCALE trains each skill $\mathcal{K}_k$ using $\mathcal{D}_k$. Each skill has compression matrices $D_{\mathbf{A}_k}$ and $D_{\mathbf{D}_k}$ with task solution datapoints (c, θ).

DGR. The DGR $\mathscr{D}$ is first trained on $\hat{c}_{\mathbf{D}_k}$ using $D_{\mathbf{D}_k}$. For this work, we use a one-class SVM to model $\hat{\mathcal{I}}$, but in principle, any one-class classification algorithm would suffice.

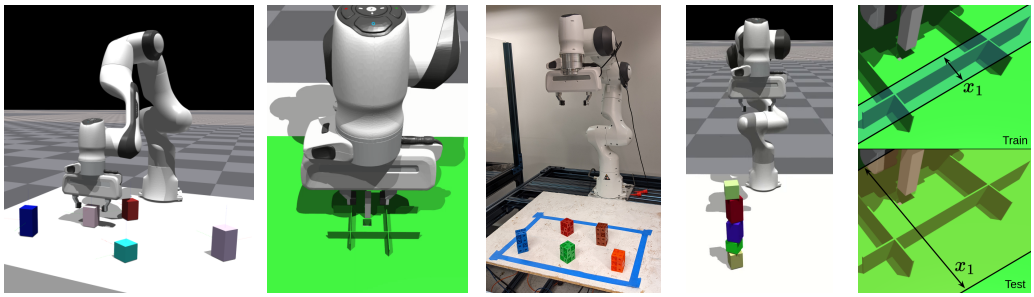
Policy. The policy is trained next. The skill dataset contexts are filtered through the DGR $\mathscr{D}(c)$ to obtain inliers c^+ for policy training data. This ensures policy training data is consistent with the initiation set. Then, the policy $\pi_{uk}(\theta|\hat{c}_{\mathbf{A}_k})$ is trained using $\hat{c}^+_{\mathbf{A}_k}$ (using $D_{\mathbf{A}_k}$) and the corresponding parameters θ^+. For our work, policies are learned using a regression model, but in principle reinforcement learning could be used as in Lee et al [33]. With π_{uk} learned, the final skill policy $\pi_k(a|\hat{c}_{\mathbf{A}_k})$ is determined.

Preconditions. The preconditions Pre are learned last through policy evaluation. Using the simulator $\mathcal{W}$, contexts c are re-sampled and evaluated with policy π_k to obtain rewards R_f. This evaluation data (c, R_f) is used to train a precondition classifier to obtain Pre. For our work, we use a nonlinear SVM classifier that has probability estimates.

6 Experimental Results

We conduct skill learning experiments with SCALE for block stacking and peg-in-hole insertion tasks with the Franka Emika Panda robot (Fig. 2): High-precision control is desirable in many industrial applications [50], motivating the ability to include only relevant variables for a policy by excluding spurious features that could degrade performance. We conduct our experiments in NVIDIA IsaacGym [51, 52], a high-fidelity physics simulator that also serves as our causal reasoning engine $\mathcal{W}$. We use a custom library that implements the scene SCM $\mathfrak{C}_S$ to facilitate scene creation and interventions. The forward simulation of physics provides the transition model T.

Baselines. We compare SCALE to baseline approaches with monolithic policies (without any skills) for either the full-dimensional context space ("monopolicy") or a reduced context space obtained by using the most commonly occurring CREST result ("crest-monopolicy"). The CREST monopolicy



(a) **(b)** **(c)** **(d)** **(e)**

Figure 2: SCALE discovers skills for the Franka Emika Panda robot using causal learning in simulation for two manipulation tasks: (a) block stacking and (b) peg-in-hole insertion. In addition to skill learning experiments, we also show how SCALE can yield skills (c) for sim-to-real transfer (App. G); (d) for generalization in downstream tasks, such as stacking a block tower (App. H); and (e) are robust to task domain shifts (App. J).

represents naïvely using CREST, ignoring that CREST provides locally different results within the underlying data (a property that SCALE leverages).

6.1 Block Stacking

Task representation. In the block stacking task, the robot starts with a source block (B_1) grasped, and it learns to place it on top of a target block (B_2). To do this, the robot uses a controller π_l that defines the trajectory for the robot end-effector to traverse via impedance control. This trajectory is parameterized by $\theta_b = [\,\theta_{\Delta x},\, \theta_{\Delta y},\, \theta_{\Delta z_u}\, \theta_{\Delta z_d},\,]^{\mathrm{T}} \in \mathbb{R}^4$, which specify waypoints the robot follows sequentially. Specifically, these parameters characterize a trajectory where the robot lifts the source block vertically, moves horizontally, descends vertically, and releases the block.

For this task, the context variables $\mathbf{C}_B$ are $\{\mathbf{C}_{B_1}, \ldots, \mathbf{C}_{B_{N_B}}, h_\pi\}$, which is the union of context variables for each of $N_B = 5$ blocks plus the table height h_π upon which the blocks are placed. The context variables for each block b are $\{x_b^w, y_b^w, \psi_b, h_b, R_b, G_b, B_b\}$, yielding a 36-dimensional context space for this problem. Here, x_b^w and y_b^w are the world x- and y-positions of the block, and the block's orientation is represented by a rotation angle ψ_b around the block's vertical axis (z). The z-dimension (height) of the block is h_i. Additional experimental details are available in App. F.

Skill learning results: variable selection. From a batch dataset of 585 samples, SCALE found the skill library $\boldsymbol{\mathcal{K}}_{blocks} = \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$ that is shown in Tab. 1 that were learned using 340 samples of the dataset. These 340 samples were selected for being the most commonly occurring within the dataset, based on a heuristic threshold. Even though there are five blocks and 36 possible variables, the skills generally consisted of a much smaller subset of variables, relating to the geometry of the source and target blocks. Note that $\mathcal{K}_2$'s relevant variables for the policy, $\mathbf{A}_{\mathcal{K}_2}$, are consistent with earlier work by Lee et al. [33] for this domain. This is generally considered to be the "ground truth" variable result for unobstructed block motion in this case. Skill $\mathcal{K}_1$ could be seen as a version of $\mathcal{K}_2$ when h_2 is not needed. Rarely, the source block's rotation ψ_1 become important (e.g., the source block's final pose was not fully stable when stacked on the target block), and thus a skill emerges with this variable ($\mathcal{K}_3$). Variables for neither block color nor table height are observed, which is an expected result as they are irrelevant to this control problem.

Skill learning results: task evaluation. We evaluate the skill library $\boldsymbol{\mathcal{K}}_{blocks}$ over the entire task distribution and show the results in Tab. 2. That is, for each context sample, the robot evaluates each skill's precondition and selects the skill with the highest probability of success. Approaches ending "-all" use the entire batch dataset, whereas "-sk" and "-subs" are ablations that use either the SCALE dataset or a randomly downsampled subset of equivalent size to the SCALE dataset, respectively. For both function classes, SCALE yields an approach that outperforms full-dimensional policies and is generally comparable to CREST-reduced policies. The overall best performing approach was scale-lin (90.49%) with generally similar performance to the CREST baselines. Performance across

Table 1: Skills $\boldsymbol{\mathcal{K}}_{blocks}$ that were discovered for the block stacking task. $\mathbf{A}$ and $\mathbf{D}$ are the variables used for the skill's policy and DGR, respectively. Data is the quantity of data used for each skill (from a batch dataset of 585 samples, 340 samples were used to train skills). Tsk. Sv. %, shown for both scale-lin and scale-nonlin, is the rate of task solves over the entire context space using only that skill.

Skill		Data	Tsk. Sv. %, Lin	Tsk. Sv. %, Nonlin.
$\mathcal{K}_1$	$\mathbf{A}$: $\{x_1^w, y_1^w, x_2^w, y_2^w\}$ $\mathbf{D}$: $\{x_1^w, y_1^w, x_2^w, y_2^w, h_2\}$	53 (9.06%)	65.36% (200)	18.36% (56)
$\mathcal{K}_2$	$\mathbf{A}$: $\{x_1^w, y_1^w, x_2^w, y_2^w, h_2\}$ $\mathbf{D}$: $\{x_1^w, y_1^w, h_1, x_2^w, y_2^w, h_2\}$	272 (46.50%)	78.76% (241)	55.88% (171)
$\mathcal{K}_3$	$\mathbf{A}$: $\{x_1^w, y_1^w, \psi_1, x_2^w, y_2^w, h_2\}$ $\mathbf{D}$: $\{x_1^w, y_1^w, \psi_1, x_2^w, y_2^w, h_2\}$	15 (2.56%)	34.31% (105)	1.31% (4)

Table 2: Task evaluation results for using the skill library $\boldsymbol{\mathcal{K}}_{blocks}$ for the block stacking task. Ctrl. is the approach control (skills or one monolithic policy). Fn. Cl. is the approach's function class. Linear approaches use Bayesian ridge regression, whereas nonlinear methods consist of a multilayer perceptron with a 16x16x16 architecture using ReLU activations. Task Solve % is the rate of task solves over the entire context space using the approach. Methods within $\pm 2\%$ (the stochasticity of the simulator) of the best approach are bold. $|\mathbf{A}|$ is the quantity of input variables used for the approach's policy. Data is the amount of training data used for the approach. A ground truth policy is also shown, using all context variables and additional domain knowledge.

Approach	Ctrl.	Fn. Cl.	Task Solve %	$\lvert\mathbf{A}\rvert$	Data
scale-lin (ours)	3 skills	Linear	**90.49%** (276)	4/5/6	340
monopolicy-lin-sk	1 policy	Linear	80.72% (247)	36	340
monopolicy-lin-all	1 policy	Linear	85.95% (263)	36	585
crest-monopolicy-lin-subs	1 policy	Linear	**89.87%** (275)	5	340
crest-monopolicy-lin-all	1 policy	Linear	**89.87%** (275)	5	585
scale-nonlin (ours)	3 skills	Nonlin.	**63.40%** (194)	4/5/6	340
monopolicy-nonlin-sk	1 policy	Nonlin.	1.31% (4)	36	340
monopolicy-nonlin-all	1 policy	Nonlin.	10.13% (31)	36	585
crest-monopolicy-nonlin-subs	1 policy	Nonlin.	58.17% (178)	5	340
crest-monopolicy-nonlin-all	1 policy	Nonlin.	**60.78%** (186)	5	585
ground-truth-policy	1 policy	Nonlin.	95.75% (293)	*	–

all nonlinear approaches was generally lower. The similarity in performance of scale-nonlin to the full-data CREST baseline is strictly due to sample size; on a sample-adjusted basis, scale-nonlin is slightly more performant. App. F provides details of the SCALE skill selection.

Sim-to-real experiment. We transfer the skills learned by SCALE and our baselines to a real Franka Panda robot without any fine-tuning. SCALE outperforms the baselines for both linear and nonlinear function classes. Please see App. G for details.

6.2 Sensorless Peg-in-Hole Insertion

Our second domain is peg-in-hole insertion under sensing uncertainty. It requires the robot to insert a cuboidal peg of cross-section 1 cm $\times$ 1 cm into a cuboidal hole of cross-section 1.3 cm $\times$ 1.3 cm. The robot gets a noisy initial position of the hole with the noise sampled from a Gaussian distribution $\mathcal{N}(0, 0.3^2 \text{ cm}^2)$. No further sensory observations are available. Due to this uncertainty, a naïve strategy of directly trying to push the peg down at the observed location of the hole achieves a success rate of only 34%. To address this, Brost [53] propose an appealing approach in which the robot takes *uncertainty reducing* actions by getting in contact with the environment (e.g., a fixture next to the hole). Our goal in this experiment is to learn such skills autonomously.

Task representation. Each assembly task has 4 axis-aligned cuboidal fixtures (i.e., walls) of fixed dimensions around the hole. The 8-dimensional context variables $\mathbf{C}_P$ are $\{x_1, y_1, \ldots, x_4, y_4\}$, containing the (x, y) coordinates of these fixtures with respect to the hole. The positions are different in every task, but it is always possible for the robot to localize against any of the walls to complete the task. We use a 6-parameter policy space: three $(\Delta x, \Delta y, \Delta z)$ actions executed in sequence in the robot's end-effector frame. In every policy, Δz's are designed to move the peg down while Δx and Δy are parameters that are learnt using RL. Additional experimental details are available in App. I.

Table 3: Skills $\boldsymbol{\mathcal{K}}_{peg}$ that were discovered for the peg-in-hole insertion task. Columns are the same as in Tab. 1, except Data represents which 168 samples were used to train skills (from a batch dataset of 210 samples).

Skill	$\mathbf{A}$	$\mathbf{D}$	Data	Task Solve %, Lin	Task Solve %, Nonlin.
$\mathcal{K}_1$	$\{\}$	$\{x_1, y_2, y_3, x_4\}$	56 (26.67%)	64.84% (166)	61.72% (158)
$\mathcal{K}_2$	$\{x_4\}$	$\{x_4\}$	25 (11.90%)	97.66% (250)	84.38% (216)
$\mathcal{K}_3$	$\{x_1\}$	$\{x_1\}$	27 (12.86%)	44.53% (114)	84.77% (217)
$\mathcal{K}_4$	$\{y_3\}$	$\{y_3\}$	28 (13.33%)	94.14% (241)	82.81% (212)
$\mathcal{K}_5$	$\{y_2\}$	$\{y_2\}$	32 (15.24%)	98.44% (252)	79.69% (204)

Table 4: Task evaluation results for using the skill library $\boldsymbol{\mathcal{K}}_{peg}$ for peg insertion (columns in Tab. 2).

Approach	Ctrl.	Fn. Cl.	Task Solve %	$\lvert\mathbf{A}\rvert$	Data
scale-lin (ours)	5 skills	Linear	**96.48%** (247)	0/1/1/1/1	168
monopolicy-lin-all	1 policy	Linear	62.50% (160)	8	210
crest-monopolicy-lin-all	1 policy	Linear	62.89% (161)	1	210
scale-nonlin (ours)	5 skills	Nonlinear	**88.67%** (227)	0/1/1/1/1	168
monopolicy-nonlin-all	1 policy	Nonlinear	12.89% (33)	8	210
crest-monopolicy-nonlin-all	1 policy	Nonlinear	55.47% (142)	1	210

Skill learning results: variable selection. Table 3 enumerates the skills $\boldsymbol{\mathcal{K}}_{peg} = \{\mathcal{K}_1, \ldots, \mathcal{K}_5\}$ discovered by SCALE. Skills $\mathcal{K}_{2-5}$ localize against one of the 4 walls. For each such skill, only the wall being used for localization is relevant to the skill and the other walls can be ignored. Consequently, the set of relevant variables for these skills contains only the distance to the wall that it localizes against. For the linear case, all of these skills except for $\mathcal{K}_3$ have high success rate, whereas success rates are slightly lower for the nonlinear approach. Interestingly, SCALE also discovers a skill $\mathcal{K}_1$ that has an empty set of relevant variables. For more discussion of this skill, see App. I.

Skill learning results: task evaluation. Table 4 presents the task evaluation of the skill library $\boldsymbol{\mathcal{K}}_{peg}$ for 256 tasks distributed across the context space. For both linear and nonlinear cases, SCALE outperforms both baselines. The low success of monopolicy-nonlin-all is likely due to insufficient data owing to a larger network. The most common CREST result was variable x_4 (21.90%), so this was used for the CREST baselines. However, it only localizes against one wall. The improvement of SCALE over the CREST baselines implies SCALE skills benefit from the DGRs through greater quality training data, whereas the CREST approaches use the entire dataset despite most samples having a differing CREST result than x_4. For details of SCALE skill selection, see App. I.

Domain shift experiment. To evaluate the out-of-distribution generalization capabilities of SCALE, we evaluate the skills on a test distribution that is significantly harder than the training distribution. While all approaches see a drop in performance, ours is much more robust than the baselines. See App. J for details.

7 Conclusion

We present SCALE, an approach for self-supervised learning of compact, diverse robot manipulation skills from causal interventions in simulation. These skills arise from the skill DGR: an underlying data generating process for how the robot should select parameters to solve a task in a particular context. We demonstrate the advantages of skill libraries discovered with SCALE for two simulation domains as well as on a real robot system.

Limitations and future work. SCALE assumes the robot has access to a causal reasoning engine. We provide this via simulation and scene structural causal models, but these models could be learned via causal discovery. SCALE primarily learns from batch dataset collection; active learning of skills would reveal useful behaviors that are statistically uncommon in the batch setting. SCALE requires that the task can be first solved in simulation (which, for our case, was learned using model-free policy search). Future work could instead leverage human demonstrations that can be analyzed using counterfactual reasoning. Lastly, SCALE assumes the context variables are disentangled. When this assumption does not hold, causal disentanglement offers a promising path forward.

References

[1] R. Desimone and J. Duncan. Neural Mechanisms of Selective Visual Attention. *Annual Review of Neuroscience*, 18(1):193–222, 1995.

[2] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Robot Learning from Demonstration by Constructing Skill Trees. *The International Journal of Robotics Research*, 31(3):360–375, 2012.

[3] Z. Su, O. Kroemer, G. E. Loeb, G. S. Sukhatme, and S. Schaal. Learning Manipulation Graphs from Demonstrations Using Multimodal Sensory Signals. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2758–2765, 2018.

[4] O. Kroemer, C. Daniel, G. Neumann, H. Van Hoof, and J. Peters. Towards Learning Hierarchical Skills for Multi-Phase Manipulation Tasks. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1503–1510, 2015.

[5] A. Chenu, N. Perrin-Gilbert, and O. Sigaud. Divide & Conquer Imitation Learning. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8630–8637, 2022.

[6] J. Achterhold, M. Krimmel, and J. Stueckler. Learning Temporally Extended Skills in Continuous Domains as Symbolic Actions for Planning. *arXiv*, abs/2207.05018, 2022.

[7] C. Devin, P. Abbeel, T. Darrell, and S. Levine. Deep Object-Centric Representations for Generalizable Robot Learning. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7111–7118, 2018.

[8] J. Z. Kolter and A. Y. Ng. Regularization and Feature Selection in Least-Squares Temporal Difference Learning. *International Conference on Machine Learning (ICML)*, pages 521–528, 2009.

[9] N. Jiang, A. Kulesza, and S. Singh. Abstraction Selection in Model-Based Reinforcement Learning. *International Conference on Machine Learning (ICML)*, pages 179–188, 2015.

[10] G. Konidaris and A. Barto. Efficient Skill Learning using Abstraction Selection. *Int'l Joint Conf. on AI (IJCAI)*, 2009.

[11] B. C. Da Silva, G. Konidaris, and A. G. Barto. Learning Parameterized Skills. *International Conference on Machine Learning (ICML)*, 2012.

[12] B. C. Da Silva, G. Baldassarre, G. Konidaris, and A. Barto. Learning Parameterized Motor Skills on a Humanoid Robot. *International Conference on Robotics and Automation (ICRA)*, 2014.

[13] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[14] L. P. Kaelbling and T. Lozano-Pérez. Learning Composable Models of Parameterized Skills. *International Conference on Robotics and Automation (ICRA)*, 2017.

[15] 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*, 40(6-7):866–894, 2021.

[16] J. Peters, J. Kober, K. Mülling, O. Krämer, and G. Neumann. Towards Robot Skill Learning: From Simple Skills to Table Tennis. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013.

[17] R. Pahič, Z. Lončarević, A. Gams, and A. Ude. Robot Skill Learning in Latent Space of a Deep Autoencoder Neural Network. *Robotics and Autonomous Systems*, 135:103690, 2021.

[18] K. Khetarpal, M. Klissarov, M. Chevalier-Boisvert, P.-L. Bacon, and D. Precup. Options of Interest: Temporal Abstraction with Interest Functions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(4):4444–4451, 2020.

[19] M. Abdulhai, D.-K. Kim, M. Riemer, M. Liu, G. Tesauro, and J. P. How. Context-Specific Representation Abstraction for Deep Option Learning. *arXiv*, abs/2109.09876, 2021.

[20] A. Bagaria and G. Konidaris. Option Discovery using Deep Skill Chaining. *International Conference on Learning Representations (ICLR)*, 2020.

[21] A. Bagaria, J. K. Senthil, and G. Konidaris. Skill Discovery for Exploration and Planning using Deep Skill Graphs. *International Conference on Machine Learning (ICML)*, 2021.

[22] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine. Diversity is All You Need: Learning Skills without a Reward Function. *International Conference on Learning Representations (ICLR)*, 2018.

[23] A. Sharma, S. S. Gu, S. Levine, V. Kumar, and K. Hausman. Dynamics-Aware Unsupervised Discovery of Skills. *International Conference on Learning Representations (ICLR)*, 2020.

[24] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman. *Causation, Prediction, and Search*. MIT Press, 2000.

[25] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

[26] J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

[27] G. W. Imbens and D. B. Rubin. *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge University Press, 2015.

[28] C. Glymour, K. Zhang, and P. Spirtes. Review of Causal Discovery Methods based on Graphical Models. *Frontiers in Genetics*, 10:524, 2019.

[29] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Towards Causal Representation Learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[30] J. Kaddour, A. Lynch, Q. Liu, M. J. Kusner, and R. Silva. Causal Machine Learning: A Survey and Open Problems. *arXiv preprint arXiv:2206.15475*, 2022.

[31] K. C. Stocking, A. Gopnik, and C. Tomlin. From Robot Learning to Robot Understanding: Leveraging Causal Graphical Models for Robotics. *Conference on Robot Learning (CoRL)*, 2022.

[32] O. Ahmed, F. Träuble, A. Goyal, A. Neitz, Y. Bengio, B. Schölkopf, M. Wüthrich, and S. Bauer. CausalWorld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning. *arXiv preprint arXiv:2010.04296*, 2020.

[33] T. E. Lee, J. Zhao, A. S. Sawhney, S. Girdhar, and O. Kroemer. Causal Reasoning in Simulation for Structure and Transfer Learning of Robot Manipulation Policies. *Int'l Conf. on Robotics and Automation (ICRA)*, 2021.

[34] Y. Li, A. Torralba, A. Anandkumar, D. Fox, and A. Garg. Causal Discovery in Physical Systems from Videos. *Advances in Neural Information Processing Systems*, 33:9180–9192, 2020.

[35] M. Diehl and K. Ramirez-Amaro. Why Did I Fail? A Causal-Based Method to Find Explanations for Robot Failures. *IEEE Robotics and Automation Letters*, 7(4):8925–8932, 2022.

[36] N. R. Ke, A. Didolkar, S. Mittal, A. Goyal, G. Lajoie, S. Bauer, D. Rezende, Y. Bengio, M. Mozer, and C. Pal. Systematic Evaluation of Causal Discovery in Visual Model Based Reinforcement Learning. *Proceedings of the NeurIPS 2021 Datasets and Benchmarks Track*, 2021.

[37] S. A. Sontakke, A. Mehrjou, L. Itti, and B. Schölkopf. Causal Curiosity: RL Agents Discovering Self-supervised Experiments for Causal Representation Learning. *International Conference on Machine Learning (ICML)*, 2021.

[38] A. Sonar, V. Pacelli, and A. Majumdar. Invariant Policy Optimization: Towards Stronger Generalization in Reinforcement Learning. *Learning for Dynamics and Control*, 2021.

[39] Z. Wang, X. Xiao, Z. Xu, Y. Zhu, and P. Stone. Causal Dynamics Learning for Task-Independent State Abstraction. *International Conference on Machine Learning (ICML)*, 2022.

[40] J. R. Kubricht, K. J. Holyoak, and H. Lu. Intuitive Physics: Current Research and Controversies. *Trends in Cognitive Sciences*, 21:749–759, 2017.

[41] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum. Simulation as an Engine of Physical Scene Understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

[42] D. Ha and J. Schmidhuber. World Models. *arXiv preprint arXiv:1803.10122*, 2018.

[43] D. Hafner, T. P. Lillicrap, M. Norouzi, and J. Ba. Mastering Atari with Discrete World Models. *International Conference on Learning Representations (ICLR)*, 2021.

[44] J. Wang, C. Hu, Y. Wang, and Y. Zhu. Dynamics Learning With Object-Centric Interaction Networks for Robot Manipulation. *IEEE Access*, 9:68277–68288, 2021.

[45] O. Kroemer, S. Niekum, and G. Konidaris. A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms. *Journal of Machine Learning Research*, 22: 30–1, 2021.

[46] R. S. Sutton, D. Precup, and S. Singh. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

[47] M. P. Deisenroth, G. Neumann, J. Peters, et al. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, 2(1–2):1–142, 2013.

[48] G. Konidaris, L. Kaelbling, and T. Lozano-Perez. Symbol Acquisition for Probabilistic High-Level Planning. *Int'l Joint Conf. on AI (IJCAI)*, 2015.

[49] J. Peters, K. Mülling, and Y. Altun. Relative Entropy Policy Search. *AAAI Conf. on Artificial Intelligence*, 2010.

[50] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, A. M. Agogino, A. Tamar, and P. Abbeel. Reinforcement Learning on Variable Impedance Controller for High-Precision Robotic Assembly. *Int'l Conf. on Robotics and Automation (ICRA)*, 2019.

[51] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox. GPU-Accelerated Robotic Simulation for Distributed Reinforcement Learning. *Conf. on Robot Learning (CoRL)*, 2018.

[52] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning. *arXiv preprint arXiv:2108.10470*, 2021.

[53] R. Brost. Automatic Grasp Planning in the Presence of Uncertainty. *Int'l Conf. on Robotics and Automation (ICRA)*, pages 1575–1581, 1986. doi:10.1109/ROBOT.1986.1087523.

[54] O. Kroemer and G. Sukhatme. Meta-level Priors for Learning Manipulation Skills with Sparse Features. *2016 International Symposium on Experimental Robotics*, 2017.

[55] S. Katz, A. Tal, and R. Basri. Direct Visibility of Point Sets. *ACM SIGGRAPH*, 2007.

[56] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the Second Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, 96(34):226–231, 1996.

[57] K. Zhang, M. Sharma, J. Liang, and O. Kroemer. A Modular Robotic Arm Control Stack for Research: Franka-Interface and FrankaPy. *arXiv preprint arXiv:2011.02398*, 2020.