Unsupervised Object Interaction Learning with Counterfactual Dynamics Models

Jongwook Choi[*1], **Sungtae Lee**[*2], **Xinyu Wang**[1], **Sungryull Sohn**[3], **Honglak Lee**[1,3]
[1]University of Michigan
[2]Individual Researcher
[3]LG AI Research
jwook@umich.edu, sytb135@gmail.com, honglak@eecs.umich.edu

Abstract

We present COIL (Counterfactual Object Interaction Learning), a novel way of learning skills of object interactions on entity-centric environments. The goal is to learn primitive behaviors that can induce interactions without external reward or any supervision. Existing skill discovery methods are limited to locomotion, simple navigation tasks, or single-object manipulation tasks, mostly not inducing interaction between objects. Unlike a monolithic representation usually used in prior skill learning methods, we propose to use a structured goal representation that can query and scope which objects to interact with, which can serve as a basis for solving more complex downstream tasks. We design a novel counterfactual intrinsic reward through the use of either a forward model or successor features that can learn an interaction skill between a pair of objects given as a goal. Through experiments on continuous control environments such as Magnetic Block and 2.5-D Stacking Box, we demonstrate that an agent can learn object interaction behaviors (e.g., attaching or stacking one block to another) without any external rewards or domain-specific knowledge.

Introduction

Reinforcement learning (RL) has achieved remarkable progress at many application domains such as playing games (Mnih et al. 2013; Vinyals et al. 2019), and robotics control (Andrychowicz et al. 2020), etc. Very often RL agents are trained to specific tasks, with access to task-specific *extrinsic* rewards. A major drawback of task-specific training is that a proper reward function needs to be given, designed, and tuned so as to achieve desired behaviors, which can be often time-consuming and limits scalability in practice. It is important to be able to solve the task with a very sparse reward signal upon completion/failure of the task, or even without any external task rewards. Unsupervised RL such as task-agnostic exploration or pre-training of skills, aiming at learning interesting or useful behaviors without the use of task rewards or offline data, can provide better initialization or useful macro-actions (skills or options) for building a hierarchical agent to solve more complex and difficult tasks (Eysenbach et al. 2018; Zhang, Yu, and Xu 2021). Unsupervised learning often enables faster learning and achieves better generalization performance when multiple tasks are given after the skill acquiral or pre-training phase.

Despite a number of successes in unsupervised skill discovery (Eysenbach et al. 2018; Sharma et al. 2019; Park et al. 2022) or task-agnostic exploration based on state-entropy maximization or diversity (Pathak et al. 2017; Burda et al. 2019), relatively only a few attempts have been made on environments and tasks with *multiple entities* (e.g. objects in robotics manipulation). In the context of robotics manipulation or (discrete) entity-centric environments other than locomotion or maze navigation environments, exploration can be quite challenging because of this nature of multiple entities. One limitation of novelty-seeking exploration methods in the reward-free context is that exploration would easily converge to a low-hanging fruit behavior where exploration mostly focuses on one particular entity. For instance, in robotics manipulation environments, diversification or novelty seeking of the entire state can be easily dominated by that of the embodied agent itself (i.e., proprioceptive states) or some easy-to-control objects only, as observed and reported in (Zhao et al. 2021; Gu et al. 2021; Park et al. 2022). More interesting primitive behaviors would be interactions between many objects, for more realistic and challenging multi-object tasks such as block stacking (Lee et al. 2021; Sancaktar, Blaes, and Martius 2022) or furniture assembly (Lee et al. 2019; Ghasemipour et al. 2022). Notably, some recent works including (Sancaktar, Blaes, and Martius 2022; Cho, Kim, and Kim 2022) present reward-free exploration and skill learning in multi-object manipulation tasks.

In this work, we focus on learning a set of primitive skills that enable interaction between different objects in a task-agnostic, unsupervised fashion. Roughly speaking, interaction between two objects can be described as an action or event that occurs when two objects have a (mutual) effect on each other. Our work leverages an inductive bias that an interaction between objects learned in a task-agnostic manner can be a useful event and hence a useful primitive behavior for solving downstream tasks. Such object-object interactions (as well as agent-object interactions) are usually sparse and difficult to reach with naive exploration, but at the same time can be useful bottleneck states an agent would want to explore and visit often to achieve bigger tasks. In

[*]These authors contributed equally.

the kitchen, for instance, an interaction between a knife and various ingredients by slicing them with a knife can be one of the basic steps necessary for cooking; when assembling smaller building blocks to build a complex object like furniture, car, or electronic device, connecting matching pieces to form a composite body would be another type of indispensable interaction. As such, it will be important to learn *skills* or primitive behaviors that would induce object-object interactions, in the promise that a hierarchical control that acts upon the interaction skills (Zhang, Yu, and Xu 2021) or chaining of skills in sequence (Slivinski, Konidaris, and Marshall 2020) should solve complex tasks much faster than flat RL agents.

We study how to learn object interaction skills in a very challenging, online, reward-free setting while minimizing the use of domain and task-specific knowledge or task-agnostic offline data, which can be difficult to obtain. More specifically, we learn a goal-conditioned policy where a goal denotes an interaction of which objects is to be made. To enable this by *learning a reward function* (Zheng et al. 2020), we design a novel intrinsic reward that is computed by counterfactual reasoning on the dynamics model (forward model and successor features) (forward model and successor features), which we call Counterfactual Object Interaction Learning (COIL).

The concept of counterfactual reasoning, i.e., "*what if...?*" — predicting or inferring the outcome if something had happened differently (Mesnard et al. 2020; Gajcin and Dusparic 2022) — naturally aligns with an intuitive interpretation of interaction: interaction is when an object's future state would have been different if it were not for the presence of the other object. In the experiments, we show that the intrinsic reward derived by counterfactual reasoning on object states can efficiently induce the interaction of objects and enable an RL agent to learn such interaction behaviors without extrinsic rewards.

Our contribution can be summarized as follows:

- We study a setting of representing goals in terms of entities and objects to interact with, in the context of unsupervised skill-based and goal-conditioned RL.
- We present a novel intrinsic reward algorithm COIL (Counterfactual Object Interaction Learning) in a reward-free unsupervised exploration setting, which uses counterfactual reasoning on forward model or successor features. We show COIL can learn skills that make the goal objects interact with each other.
- We show that such an entity-centric interaction skill is generalizable to unseen, more object settings.

Approach

Preliminaries and Notations

Throughout the paper, we consider the task as an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $\mathcal{P}$ is a transition probability, $\mathcal{R}$ is a (extrinsic) task reward function, and $0 \leq \gamma < 1$ is a discount factor. Our goal is task-agnostic, unsupervised skill learning with no extrinsic rewards. We assume that the state space $\mathcal{S}$ can be explicitly factorized as the Cartesian product $(\mathcal{S}_{\text{object}})^N \times \mathcal{S}_{\text{agent}}$ where N is the number of objects, $\mathcal{S}_{\text{object}}$ is the object state space, and $\mathcal{S}_{\text{agent}}$ is the agent state space. We also assume the joint object space is permutation-invariant, i.e., $\{o_1, \cdots, o_N\}$ is a *set* (where $o_i \in \mathcal{S}_{\text{object}}$). Such a structural representation is common in robotics control (Keramati et al. 2018; Zhao et al. 2021; Sancaktar, Blaes, and Martius 2022) and is a mild assumption. However, our method is not necessarily limited to state-based control only, as one could combine with existing entity-centric representation learning methods from pixel observations (Watters et al. 2017; Greff et al. 2019; Xu et al. 2019; Veerapaneni et al. 2020; Locatello et al. 2020).

Goal representation. Skills are usually modeled in the form of goal-conditioned policies, $\pi(a|s, g)$, where $g \in \mathcal{G}$ represents a *goal*. Common choices for goal g include full state observation, a handcrafted goal with domain knowledge, or latent variables. Our particular choice is a pair of objects, namely A and B (among the N objects). A semantic meaning for this goal representation would be that two objects A and B should have an interaction (or some mutual effects) as a consequence of agent's actions. In our settings, for the sake of simplicity, we assume the reference to objects are simply categorical indices (or pointers), i.e., $A, B \in [N] = \{0, 1, \cdots, N-1\}$, respectively. However, more in general (e.g., for image observations), the goal representation for target objects can be replaced with a continuous vector to represent a reference to an *arbitrary* object in the current state, e.g., $g = (o^A, o^B)$ where $o^A, o^B \in \mathcal{S}_{\text{object}}$, which we leave as a future work.

Learning Interaction Skills with Counterfactual Forward Model

How can we learn interaction skills for two given objects, and how can we learn a reward function that would incentivize interactions between two objects? Our goal is to simultaneously learn such a reward function and object-object interaction skills in a *reward-free* setting.

Our main idea is to use a *counterfactual reasoning*; i.e., predict what would have happened instead if other objects involved in an interaction were not there or were in a different state. We argue that this form of inductive bias can provide us with a useful learning signal for interaction learning without relying on an external task reward.

Given a trajectory of observations as object states, we want to identify whether an interaction between two objects happened or not. Roughly speaking, we can say a (physical) interaction occurs between two objects A and B if and only if there exists a counterfactual state for B that would change the future state of A (and vice versa). (*Case* I) When an interaction between A and B happened, these two objects would have affected each other's state. In other words, the future state of an object would have been different without a specific configuration of the other object, provided that an interaction happened. (*Case* II) On the contrary, when there was no interaction between them, the future state of an object would remain almost the same or not dramatically different regardless of the counterpart object. A motivating example is depicted in Figure 1.

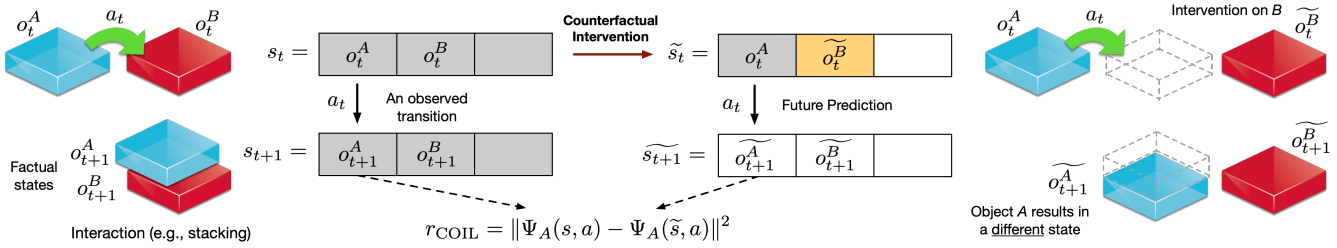


Figure 1: A high-level overview of COIL. (i) Suppose an interaction was made, then a counterfactual intervention on object B (e.g., putting it aside or change the object state randomly) would have made the future state of object A different. (ii) If no interaction was made, object A would remain in the same state regardless of the counterfactual intervention. (iii) We measure the discrepancy of object A with and without the counterfactual intervention, which becomes the intrinsic reward for interaction.

We can formalize this idea as follows. Consider a MDP transition observed by an agent, (s_t, a_t, s_{t+1}) where $s_t = \{o_t^A, o_t^B, \cdots\}$ and $s_{t+1} = \{o_{t+1}^A, o_{t+1}^B, \cdots\}$ (without the loss of generality) for a pair of objects A and B given as a goal g. We would want to tell whether an interaction was made in this transition.

(*Case* I) Suppose an interaction between object A and B happened, where A got affected by B in the interaction (without the loss of generality). Then, if we made an *counterfactual intervention* on the object B, i.e., changing the object state o_t^B randomly with $\widetilde{o_t^B}$ to obtain an *intervened state* $\widetilde{s}_t = \{o_t^A, \widetilde{o_t^B}, \cdots\}$, the same action a_t applied on $\widetilde{s}_t$ would have resulted in a different (counterfactual) next state $\widetilde{o_{t+1}^A}$ of object A than its (factual) next state o_{t+1}^A. In other words, the discrepancy between the factual next state o_{t+1}^A and the counterfactual next state $\widetilde{o_{t+1}^A}$ will be high.

(*Case* II) On the other hand, when there was no interaction happened between the two in this transition, we can expect that o_{t+1}^A would remain the same regardless of the intervention $\widetilde{o^B}$ on B, i.e., it would be that $\widetilde{o_{t+1}^A} = o_{t+1}^A$. To put together, the difference between o_{t+1}^A and $\widetilde{o_{t+1}^A}$ (e.g., $\|o_{t+1}^A - \widetilde{o_{t+1}^A}\|^2$) can quantize the degree of an interaction between objects A and B.

However, the counterfactual next state $\widetilde{s}_{t+1}$ is not observable by an agent. So we can instead predict the object A's next state by learning a forward dynamics model:

$$\widehat{o_{t+1}^A} = f_{\text{forward}}(o_t^A, \widetilde{o_t^B}, a_t, s^t \setminus \{o_t^A, o_t^B\}) \tag{1}$$

This gives us a counterfactual interaction reward function: computationally, we first make a random intervention $\widetilde{o_t^B}$ on object B, and plug it to the forward model to predict the next state $\widehat{o_{t+1}^A}$ of object A. Intervention on the object B can be implemented in many ways, such as random perturbation of the state vector by adding Gaussian noises, but an easy yet effective way to yield in-distribution randomization is to randomly sample an object state from the replay buffer.

Finally, we define the counterfactual interaction reward $r_{\text{COIL-Forward}}(s_t, a_t, s_{t+1}) = \|o_{t+1}^A - \widehat{o_{t+1}^A}\|^2$, which can be maximized by any underlying RL method (e.g., SAC or DQN) with a simultaneous learning of the forward model and the goal-conditioned policy π. We call this resulting algorithm **COIL** (Counterfactual Object Interaction Learning) and specifically this variant of using forward model **COIL-Forward**.

Learning Interaction Skills with Counterfactual Successor Features

In this section, we will present an improvement to COIL-Forward, called **COIL-SF**. One downside of the COIL-Forward is that it assumes the counterfactual intervention would have an immediate, easily distinguishable change within a single-step transition. In many realistic environments, the effect and consequence of interaction is *delayed* to be discernible enough; the change actually exists in the true world state but an observer would not be able to recognize the subtle difference until a few time step has elapsed. Therefore, it is practically important to take long-term futures into consideration so as to correctly evaluate the consequence of counterfactual interventions.

One natural way to deal with this problem would be to learn a multi-step, recurrent forward dynamics model (Oh et al. 2015). However, learning such a forward model can be challenging due to high uncertainty and the quick accumulation of prediction errors over a long horizon (Moerland, Broekens, and Jonker 2020; Lutter et al. 2021). Instead of learning a multi-step forward model, we propose to use the successor features (SF) (Dayan 1993; Barreto et al. 2016) to incorporate long-term futures that can still derive a reward signal for interaction learning.

A successor feature $\Psi^\pi(s, a)$ of a state s with respect to a policy π is an expected discounted sum of the feature of future states to be visited when starting from the state s and the action a, and following the policy π thereafter:

$$\Psi^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \Phi(s_t) \,\middle|\, s_0 = s, a_0 = a \right]. \tag{2}$$

where $\Phi(s_t)$ is called the cumulant, which is the feature of future states to accumulate. Successor features can be seen as an instance of generalized value functions (GVF) (Sutton et al. 2011) that *predicts the future* and *summarize* what will happen in the future for a state s in some specific form,

which can be easier than directly predicting the next states accurately. Successor features can be learned using simple TD learning methods like Q-learning (Dayan 1993).

To derive a reward function that tells whether an interaction is made or not, let's again consider two objects A and B given as a goal g, and focus on the future of object A when a counterfactual intervention is made on the object B. For this, we consider an entity-centric successor feature with an object cumulant function $\phi : \mathcal{S}_{\text{object}} \rightarrow \mathbb{R}^d$ for the target object o^A in the state observation s:

$$\Psi^\pi_A(s,a) = \Psi^\pi_A(\{o^A, o^B, \cdots\}, a) \tag{3}$$

$$= \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \phi(o^A) \,\middle|\, s_0 = s, a_0 = a \right] \tag{4}$$

for a query state $s = \{o^A, o^B, \cdots\}$. The SF $\Psi^\pi_A(s,a) \in \mathbb{R}^d$ summarizes the future state of object A with respect to the policy π. In general, one could learn the object representation $\phi(\cdot)$ with some auxiliary objectives (e.g., in pixel-based control), use some prior domain knowledge, or use a fixed random function as in (Zhang et al. 2019), but in a state-based control one can simply set $\phi(o^A) = o^A$ without an extra need to learn the cumulant feature function.

The reward function can be derived as in COIL-Forward: let's suppose we make a counterfactual intervention on object B at timestep t to get the intervened object state $\widetilde{o^B}$ from o_B. Denoting $\widetilde{s} = \{o^A, \widetilde{o^B}, \cdots\}$, the reward function for interaction can be written as

$$r_{\text{COIL-SF}}(s,a,s') = \|\Psi^\pi_A(s,a) - \Psi^\pi_A(\widetilde{s},a)\|^2. \tag{5}$$

We call this variant of using successor features for learning interactions **COIL-SF**. This reward also can be explained as follows: (*Case* II) When there was no interaction happened between objects A and B, the entity-centric successor features Ψ^π_A will be the same regardless of the intervention, in which case $r_{\text{COIL-SF}}$ would be 0. Note that, in practice, rewards for non-interaction transitions might be slightly bigger than 0 due to the epistemic uncertainty of the model. (*Case* I) On the other hand, if the future state of the object A would have changed much due to the intervention on object B, the SF values $\Psi^\pi_A(s,a)$ and $\Psi^\pi_A(\widetilde{s},a)$ will be different, in which $r_{\text{COIL-SF}}$ will evaluate to a higher scalar value. In the Experiments section, we present an analysis of the learned reward function for different types of transitions (e.g., a high reward is indeed given when interaction happens). Learning of COIL-SF also involves a simultaneous optimization of SF and policy;

Related Work

Object-Oriented RL Object-oriented RL (Diuk, Cohen, and Littman 2008) aims at improving data efficiency and generalization by leveraging representation of multiple objects and their relations. C-SWM (Kipf, van der Pol, and Welling 2019) proposes a GNN-based network to learn the world model of the object-based task using contrastive learning. Compared to models based on pixel reconstruction, C-SWM provides a rich representation of objects. CEE-US (Sancaktar, Blaes, and Martius 2022) utilizes the epistemic uncertainty of structured world model (Kipf, van der Pol, and Welling 2019) as an intrinsic reward and uses it to gather data for the world model training. The world model is then used for planning to solve downstream tasks. The behavior that emerges during world model training is mostly object manipulation rather than interactions between objects and their algorithm can learn object-object interaction only when an extrinsic reward is provided.

Exploration Cho, Kim, and Kim (2022) proposed a mutual-information (MI) based exploration algorithm to induce interactions between the *agent* and an object, which combines the MUSIC objective (Zhao et al. 2021), i.e., MI between agent and object, and the diversity term similar to DADS (Sharma et al. 2019) for the object's future state. Seitzer, Schölkopf, and Martius (2021) used object-centric causal action-influence as an intrinsic reward. However, interactions between different objects are not considered, and the skills are limited to simple control of a single target object specified by the task. Very recently, Sancaktar, Blaes, and Martius (2022) proposed curiosity-based exploration algorithm that learns a GNN-based world model, with the intrinsic reward being the epistemic uncertainty through ensemble disagreement (Pathak, Gandhi, and Gupta 2019). This work is the closest to our work, but despite GNN's ability to generalize to multiple objects during planning, their monolithic skill representation is limited to be useful for hierarchical learning or planning.

Several papers have proposed exploration methods using successor features (SF). Zhang et al. (2019) use the difference of SF between consecutive states as an intrinsic reward to efficiently explore bottleneck states. Machado, Bellemare, and Bowling (2020) propose an inverse of the L1-norm of the SF as a variant of count-based exploration. Hoang et al. (2021) utilize SF to define the distance function between states and learn a goal-conditioned policy to drive exploration. However, to the best of our knowledge, SF has not been used in object-centric environments and has not been combined with counterfactual reasoning.

Counterfactual Reasoning in RL Buesing et al. (2018) use a structural causal model in POMDP, which generates counterfactual trajectories for background planning, leading to a better sample efficiency and smaller bias of the prediction in guided policy search. Sharma and Kroemer (2020) utilize an inductive bias that, in similar scenes, if similar action has been taken it would give similar results. They utilize contrastive learning in object-centric tasks to acquire an object relation model, which is subsequently utilized in real-world precondition learning tasks. Counterfactual Credit Assignment (Mesnard et al. 2020) utilizes counterfactual reasoning on action to achieve unbiased, low variance credit assignment. Most approaches do counterfactual inference on the agent's action, i.e., concerns what would have happened if the agent made a *different decision* (i.e., action or goal); our approach differs in the sense that our counterfactual intervention is made on the object states instead of the agent's action.

Experiments

Environments

In the experiments, we test our proposed approach on multi-object continuous control environments: a toy environment (**StackingBox**) and more challenging environment (**Magnetic Blocks**).

Stacking Box. Stacking Box is a 2.5-D continuous control environment in which a cursor agent and multiple box-shaped objects of the same size are randomly spread throughout a fixed arena. The agent can move in any direction within the xy plane and can grab an object that overlaps with the agent. If the agent moves towards an object while holding another object, the object being held and moved will be placed on top of any other existing object. We assume that the height of each object is quantized to integer values (such as $0, 1, 2, \ldots$). The process of stacking one object onto another occurs instantly in a single MDP transition.

Magnetic Blocks. Magnetic Blocks is a continuous control environment in which an embodied cursor agent can interact with square-shaped block objects. The agent has a continuous action space that includes movement (translation), rotation, and grabbing through control of the joint's torque. The agent can move freely within the arena and can grab an adjacent object by slightly lifting up and moving around the object, or rotating it along with the agent. When the agent moves a held object close enough to another object such that the two objects become parallel, they will be connected by magnetic force. If the edges are not parallel, one object will push the other. A distinctive interaction in this environment is observed when two objects become connected through magnetic forces and then move together.

Implementation Details

The full network architecture for the policy and the model is shown in Figure 2. Taking the factorized state representation into consideration, we use a network with scaled dot-product attention architecture (Vaswani et al. 2017) to transform object states into desired outputs (actor, critic, and forward/-successor models). We note that the shared parameters for key and value matrices on the $N-2$ objects other than the goal objects allows the network to be permutation-invariant over their ordering, and that such an architecture allows generalization to a different number of objects.

COIL alternatingly updates the policy (actor and critic) and the model (forward model or successor features); for RL algorithm, we use SAC (Haarnoja et al. 2018) although COIL can be combined with any RL algorithms.

Performance of Learning Object Interaction Skills: Quantitative Results

We first study how well the proposed approach (COIL) can learn object interaction skills in a reward-free setting, with a comparison to strong exploration methods. At the beginning of every episode, a goal $g = (A, B)$ is chosen randomly to specify which objects should interact.

Baselines. (1) Sparse-GT: A SAC agent trained to maximize the *sparse* ground-truth interaction reward, where the per-step reward is 1 if a correct interaction between the target objects is made (e.g., stacking or magnetic connection) or 0 otherwise, which is the same as the success metric. (2) Forward-Curiosity: this maximizes the prediction error of the forward model for object A as an intrinsic reward: $\|o^A_{t+1} - f_{\text{forward}}(s_t, a_t, g_t)\|^2$. (3) SID (Zhang et al. 2019): this maximizes the "successor feature control" reward: $\|\Psi(o^A_{t+1}) - \Psi(o^A_t)\|^2$. (4) RND (Burda et al. 2019): this maximizes the prediction error of a randomly initialized network's feature representation of the target object's state as an intrinsic reward: $\|f_{\text{random}}(o^A_{t+1}) - f(o^A_{t+1})\|^2$.

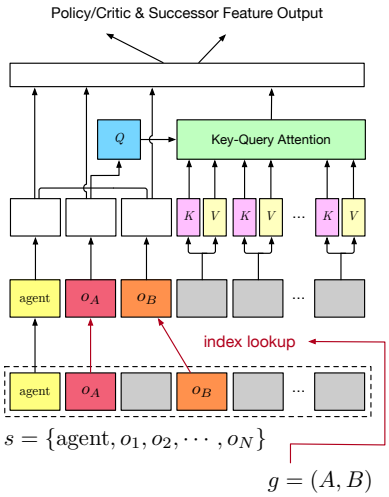
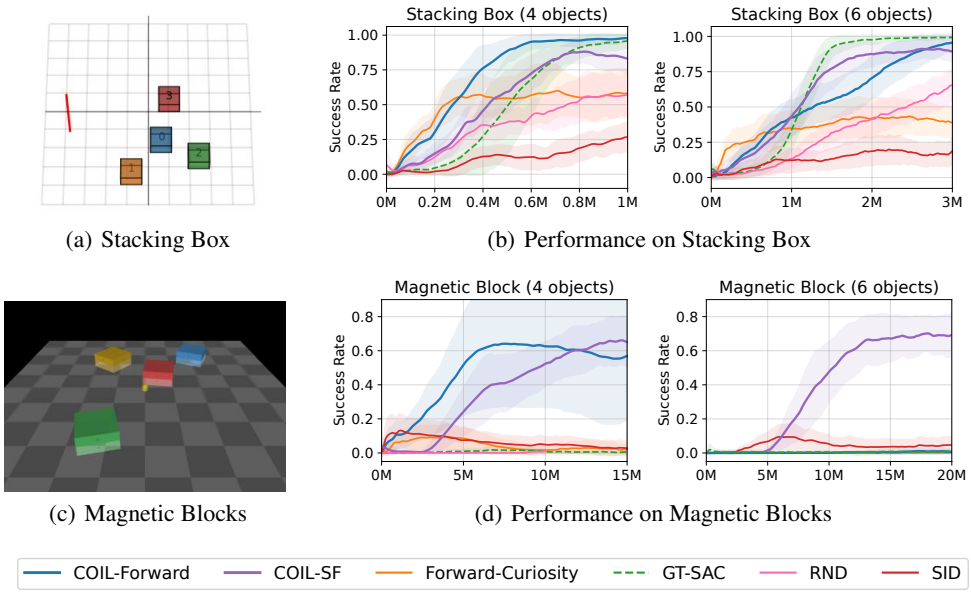


Figure 2: A network architecture used in the experiments.

For object-centric tasks, interactions can lead to significant changes in the object's state, making it desirable to employ curiosity-based exploration methods as baselines. RND is a state-of-the-art exploration method that seeks novel states, and Forward-Curiosity and SID are curiosity-based exploration techniques that use the Forward Model and Successor Feature, respectively.

Quantitative Results. The success rate of the algorithms is displayed in Figure 3, based on the evaluation episodes. Successful outcome is defined as the stacking of one object on the other in Stacking Box and the connection of the two selected goal blocks in Magnetic Blocks.

Stacking Box. COIL agents converge to a success rate of approximately 1.0, while curiosity-based exploration methods show limitations with upper bounds in their performance. One thing to note is that COIL-Forward outperforms COIL-SF in Stacking Box with 4 objects. In the Stacking Box environment, interactions occur instantaneously, enabling the 1-step forward model of COIL-Forward capture the occurrence of the interaction. This is supported by an analysis of the error of the dynamics model. (see Appendix). Transitions involving interactions exhibit a significantly higher ratio of the counterfactual prediction error (i.e., the prediction error when counterfactual intervention is made) to epistemic uncertainty, compared to transitions without events. On the other hand, Forward-Curiosity, SID, and RND are limited to manipulating individual objects



(a) Stacking Box (b) Performance on Stacking Box

(c) Magnetic Blocks (d) Performance on Magnetic Blocks

Figure 3: Progress of the success rate on the Stacking Box and the Magnetic Blocks environment. Runs are averaged over 5 random seeds. See "Quantitative Results." in the Experiments section for analyses and interpretation of the result.

without learning interaction stably (see Appendix).

Magnetic Blocks. COIL-SF is the only algorithm that successfully learns interaction skills between objects. Despite leveraging domain knowledge regarding the occurrence of interactions, Sparse-GT fails to learn even the basic task of grabbing an object. (see Appendix). Forward-Curiosity, SID, and RND can learn how to grab an object but interaction between the objects barely happen. This suggests that learning to induce interactions between objects in Magnetic Blocks is a challenging exploration problem, unlike the Stacking Box environment.

We find COIL-Forward is not effective enough to learn interactions in Magnetic Blocks, which accords with the motivation discussed above. In this environment, interactions make only a subtle difference in the object's state during a single-step transition and can be better discerned only in longer-term future; we verify this by analyzing the dynamics model errors (see Appendix) . When interactions occur, the counterfacutal prediction error is not significantly higher than the epistemic uncertainty in the forward model (in COIL-Forward). However, the counterfactual prediction error of the successor feature (in COIL-SF), is significantly higher than the epistemic uncertainty despite the counterfactual intervention, so the interaction reward could lead to learning interactions.

Qualitative Results

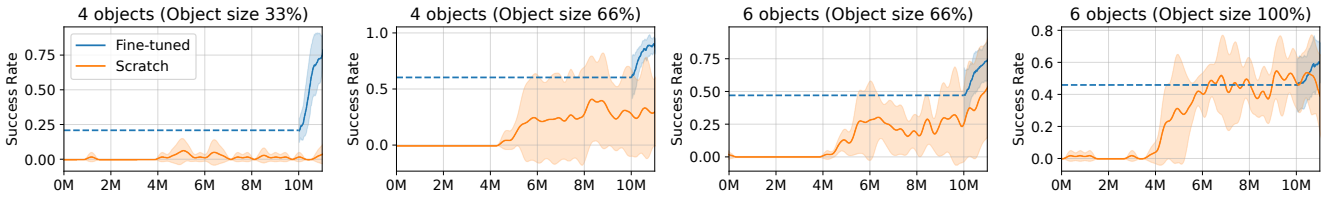
In Stacking Box, a typical interaction behavior for $g = (A, B)$ that COIL learns is to stack object A on object B. Note that A should be on top of B (i.e., bigger z coordinate) to say interaction happened. If the B were on top of A, changing or perturbing the state of B would not affect the A's state. An interesting finding was that the agent repeatedly stacked and unstacked the boxes, resulting in multiple interactions within a single episode.

In Magnetic Blocks, the interaction behavior is to grab object A and connect it to object B by making some movements and rotations as needed. Note that the agent needs to rotate objects accurately to connect the blocks, which makes the environment require some good exploration strategies to successfully learn object-object interactions. We present snapshots of COIL-SF's typical behaviors in Magnetic Blocks in Appendix. Typically, the agent grabs the object A and approaches the object B to make these two objects connected to each other, and pushing them further to move the compound around.

Analysis of COIL-SF Reward

To analyze what reward function COIL-SF has learned, we labeled each state with the following 7 categories on the **Magnetic Blocks** environment with 4 objects.

- Grab-A: the agent is grabbing the object A.
- Grab-B: the agent is grabbing the object B.
- Connect-AB: the objects A and B correctly connected. Note that when A and B are connected, the object A will be highly likely to be be affected by the object B, i.e., interaction occurs.
- Connect-AB-Only: a subset of Connect-AB states, excluding states where objects other than A and B are connected as well.
- Connect-AX: the object A is connected to a wrong object (X), i.e., anything but to B. This is a falsy interaction that does not conform to the goal given to the agent.
- Connect-BX: the object B is connected to a wrong object (X), i.e., anything but to A. This is also a falsy interaction that does not conform to the goal given to the agent.



(a) smaller size (b) smaller size (c) more objects, smaller size (d) more objects

Figure 4: Progress of the success rate when fine-tuning from a COIL-SF agent *pre-trained on the* 4 objects (object size 100%) setting in Magnetic Blocks environment. Runs are averaged over 3 random seeds.

State Labels	Average Reward	Relative Ratio
No-Event	0.7	0.040
Grab-A	14.21	0.803
Grab-B	10.7	0.604
Connect-AB	17.3	0.977
Connect-AB-Only	**17.7**	**1.0**
Connect-AX	0.51	0.029
Connect-BX	1.8	0.101

Table 1: Average COIL-SF reward given to the 7 types of states on the Magnetic Blocks environment. COIL-SF gives the highest reward to Connect-AB-Only.

- No-Event: all other states not included in the above 6 categories (e.g., the agent wanders around and does nothing)

Table 1 shows an average reward given to states with each label, for a successful instance of COIL-SF. Among the 7 labels, Connect-AB-Only receives the highest rewards. Connect-AB receives a slightly lower reward than Connect-AB-Only. Considering that Connect-AX or Connect-BX receive small rewards, we assume that a small portion of Connect-AB states are the states where objects other than A and B are also connected, and those states have small rewards. Grab-A and Grab-B receive high rewards compared to Connect-AX, Connect-BX, or No-Event. This may be due to Grab-A having an intersection with Connect-AB-Only, which is a set of states where objects A and B are connected and the agent is grabbing the object A.

Generalization to More/Unseen objects

We evaluate the object interaction skills learned by COIL-SF, testing whether they can be applied to environments with more and unseen objects. First, the policy and successor feature networks are pre-trained on Magnetic Blocks with 4 objects for 10 million steps and perform fine-tuning for 1 million additional steps. For each setup, the performance of COIL-SF fine-tuned from pre-trained networks is compared to that of COIL-SF trained from scratch for (10+1) million steps, ensuring a fair comparison. We tested the generalization ability on 4 different setups with varying object sizes and numbers: (a) 4 objects, 33% object size, (b) 4 objects, 66% object size, (c) 6 objects, 100% object size, and (d) 6 objects, 66% object size.

Unseen objects: (a), (b) To test the generalization ability of COIL-SF on unseen objects, we varied the scale (size) of the objects by 33% or 66%. The Figure 4 (a-b) show the performance of COIL-SF fine-tuned on pre-trained networks. When tested on the 66% scale, COIL-SF gets a high success rate even without any training. When tested on the 33% scale, the initial performance of COIL-SF is poor, but the performance improves rapidly within 1 million steps of further training while learning COIL-SF from the scratch fails.

More and Unseen objects: (c), (d) To test the generalization ability of COIL-SF on a different number of objects, we conducted experiments with more objects and varying scales (66%, 100%). The Figure 4 (c-d) show the performance of COIL-SF. Surprisingly COIL-SF fined-tuned on pre-trained networks performs better even in more and unseen objects settings indicating that skills learned from COIL-SF can be used as task-agnostic skills.

Overall, the successful learning of task-agnostic skills with COIL-SF has implications for future research, as these skills could potentially be incorporated into hierarchical reinforcement learning for more complex tasks.

Conclusion

In this paper, we introduce **COIL** (Counterfactual Object Interaction Learning), a novel approach to learning object-object interaction skills using intrinsic rewards, and the concept of counterfactual dynamics. Our results demonstrate that COIL can effectively learn to interact with objects in challenging continuous, object-centric environments outperforming all the baselines including Sparse-GT, which incorporates task-specific knowledge. We also showed a generalization ability of interaction skills learned by COIL.

Given the complexity and diversity of real-world tasks such as furniture assembly or complex robotics object manipulation, we believe that unsupervised learning of object-object interactions is important, and COIL presents a significant step towards this challenging goal. We note that COIL has some limitations that the method currently relies on a factorized state representation, and do not consider diverse modes of interaction skills. Considering that the real-world tasks contain multiple modes of interaction and complex state representation, combining diverse skill learning (Eysenbach et al. 2018; Sharma et al. 2019; Park et al. 2022) and object-centric representation learning methods (Locatello et al. 2020) will be an interesting future work.

Acknowledgements

This work was supported in part by grants from LG AI Research and NSF FW-HTF-R #2128623.

References

Andrychowicz, O. M.; Baker, B.; Chociej, M.; Jozefowicz, R.; McGrew, B.; Pachocki, J.; Petron, A.; Plappert, M.; Powell, G.; Ray, A.; et al. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1): 3–20.

Barreto, A.; Dabney, W.; Munos, R.; Hunt, J. J.; Schaul, T.; Silver, D.; and Hasselt, H. V. 2016. Successor Features for Transfer in Reinforcement Learning.

Buesing, L.; Weber, T.; Zwols, Y.; Racanière, S.; Guez, A.; Lespiau, J.; and Heess, N. 2018. Woulda, Coulda, Shoulda: Counterfactually-Guided Policy Search.

Burda, Y.; Edwards, H.; Storkey, A.; and Klimov, O. 2019. Exploration by Random Network Distillation.

Cho, D.; Kim, J.; and Kim, H. J. 2022. Unsupervised Reinforcement Learning for Transferable Manipulation Skill Discovery. *IEEE Robotics and Automation Letters*.

Dayan, P. 1993. Improving Generalization for Temporal Difference Learning: The Successor Representation. *Neural Computation*.

Diuk, C.; Cohen, A.; and Littman, M. 2008. An object-oriented representation for efficient reinforcement learning. *ICML*.

Eysenbach, B.; Gupta, A.; Ibarz, J.; and Levine, S. 2018. Diversity is All You Need: Learning Skills without a Reward Function. *ICLR*.

Gajcin, J.; and Dusparic, I. 2022. Counterfactual Explanations for Reinforcement Learning. *arXiv preprint arXiv:2210.11846*.

Ghasemipour, S. K. S.; Freeman, D.; David, B.; Gu, S. S.; Kataoka, S.; and Mordatch, I. 2022. Blocks Assemble! Learning to Assemble with Large-Scale Structured Reinforcement Learning. *arXiv preprint arXiv:2203.13733*.

Greff, K.; Kaufman, R. L.; Kabra, R.; Watters, N.; Burgess, C.; Zoran, D.; Matthey, L.; Botvinick, M.; and Lerchner, A. 2019. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, 2424–2433. PMLR.

Gu, S.; Diaz, M.; Freeman, D.; Furuta, H.; Ghasemipour, S. K. S.; Raichuk, A.; David, B.; Frey, E.; Coumans, E.; and Bachem, O. 2021. Braxlines: Fast and Interactive Toolkit for RL-driven Behavior Engineering beyond Reward Maximization. *ArXiv*.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. PMLR.

Hoang, C.; Sohn, S.; Choi, J.; Carvalho, W.; and Lee, H. 2021. Successor Feature Landmarks for Long-Horizon Goal-Conditioned Reinforcement Learning. volume 34, 26963–26975.

Keramati, R.; Whang, J.; Cho, P.; and Brunskill, E. 2018. Strategic object oriented reinforcement learning.

Kipf, T.; van der Pol, E.; and Welling, M. 2019. Contrastive Learning of Structured World Models. *ICLR*.

Lee, A. X.; Devin, C.; Zhou, Y.; Lampe, T.; Bousmalis, K.; Springenberg, J. T.; Byravan, A.; Abdolmaleki, A.; Gileadi, N.; Khosid, D.; Fantacci, C.; Chen, J. E.; Raju, A.; Jeong, R.; Neunert, M.; Laurens, A.; Saliceti, S.; Casarini, F.; Riedmiller, M. A.; Hadsell, R.; and Nori, F. 2021. Beyond Pick-and-Place: Tackling Robotic Stacking of Diverse Shapes. *ArXiv*.

Lee, Y.; Hu, E.; Yang, Z.; Yin, A.; and Lim, J. J. 2019. IKEA Furniture Assembly Environment for Long-Horizon Complex Manipulation Tasks. *IEEE International Conference on Robotics and Automation*.

Locatello, F.; Weissenborn, D.; Unterthiner, T.; Mahendran, A.; Heigold, G.; Uszkoreit, J.; Dosovitskiy, A.; and Kipf, T. 2020. Object-Centric Learning with Slot Attention.

Lutter, M.; Hasenclever, L.; Byravan, A.; Dulac-Arnold, G.; Trochim, P.; Heess, N.; Merel, J.; and Tassa, Y. 2021. Learning Dynamics Models for Model Predictive Agents. *ArXiv*.

Machado, M. C.; Bellemare, M. G.; and Bowling, M. 2020. Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 5125–5133.

Mesnard, T.; Weber, T.; Viola, F.; Thakoor, S.; Saade, A.; Harutyunyan, A.; Dabney, W.; Stepleton, T.; Heess, N.; Guez, A.; et al. 2020. Counterfactual credit assignment in model-free reinforcement learning. *arXiv preprint arXiv:2011.09464*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. A. 2013. Playing Atari with Deep Reinforcement Learning. *ArXiv*.

Moerland, T. M.; Broekens, J.; and Jonker, C. M. 2020. Model-based Reinforcement Learning: A Survey. *Found. Trends Mach. Learn.*, 16: 1–118.

Oh, J.; Guo, X.; Lee, H.; Lewis, R. L.; and Singh, S. 2015. Action-Conditional Video Prediction using Deep Networks in Atari Games. In *NeurIPS*.

Park, S.; Choi, J.; Kim, J.; Lee, H.; and Kim, G. 2022. Lipschitz-constrained Unsupervised Skill Discovery. *ICLR*.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-Driven Exploration by Self-Supervised Prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

Pathak, D.; Gandhi, D.; and Gupta, A. 2019. Self-Supervised Exploration via Disagreement.

Sancaktar, C.; Blaes, S.; and Martius, G. 2022. Curious Exploration via Structured World Models Yields Zero-Shot Object Manipulation. *arXiv preprint arXiv:2206.11403*.

Seitzer, M.; Schölkopf, B.; and Martius, G. 2021. Causal influence detection for improving efficiency in reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 22905–22918.

Sharma, A.; Gu, S.; Levine, S.; Kumar, V.; and Hausman, K. 2019. Dynamics-Aware Unsupervised Discovery of Skills. *ICLR*.

Sharma, M.; and Kroemer, O. 2020. Relational Learning for Skill Preconditions.

Slivinski, M.; Konidaris, G.; and Marshall, L. 2020. Robust Deep Skill Chaining. .

Sutton, R.; Modayil, J.; Delp, M.; Degris, T.; Pilarski, P.; White, A.; and Precup, D. 2011. Horde: a scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. *Adaptive Agents and Multi-Agent Systems*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Veerapaneni, R.; Co-Reyes, J. D.; Chang, M.; Janner, M.; Finn, C.; Wu, J.; Tenenbaum, J.; and Levine, S. 2020. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning*, 1439–1456. PMLR.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; Oh, J.; Horgan, D.; Kroiss, M.; Danihelka, I.; Huang, A.; Sifre, L.; Cai, T.; Agapiou, J.; Jaderberg, M.; Vezhnevets, A.; Leblond, R.; Pohlen, T.; Dalibard, V.; Budden, D.; Sulsky, Y.; Molloy, J.; Paine, T.; Gulcehre, C.; Wang, Z.; Pfaff, T.; Wu, Y.; Ring, R.; Yogatama, D.; Wünsch, D.; McKinney, K.; Smith, O.; Schaul, T.; Lillicrap, T.; Kavukcuoglu, K.; Hassabis, D.; Apps, C.; and Silver, D. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*.

Watters, N.; Zoran, D.; Weber, T.; Battaglia, P.; Pascanu, R.; and Tacchetti, A. 2017. Visual interaction networks: Learning a physics simulator from video. *Advances in neural information processing systems*, 30.

Xu, Z.; Liu, Z.; Sun, C.; Murphy, K.; Freeman, W. T.; Tenenbaum, J. B.; and Wu, J. 2019. Unsupervised discovery of parts, structure, and dynamics. *arXiv preprint arXiv:1903.05136*.

Zhang, J.; Wetzel, N.; Dorka, N.; Boedecker, J.; and Burgard, W. 2019. Scheduled intrinsic drive: A hierarchical take on intrinsically motivated exploration. *arXiv preprint arXiv:1903.07400*.

Zhang, J.; Yu, H.; and Xu, W. 2021. Hierarchical Reinforcement Learning By Discovering Intrinsic Options. *International Conference on Learning Representations*.

Zhao, R.; Gao, Y.; Abbeel, P.; Tresp, V.; and Xu, W. 2021. Mutual Information State Intrinsic Control. *ICLR*.

Zheng, Z.; Oh, J.; Hessel, M.; Xu, Z.; Kroiss, M.; van Hasselt, H.; Silver, D.; and Singh, S. 2020. What Can Learned Intrinsic Rewards Capture? arXiv:1912.05500.