REINFORCEMENT LEARNING FOR SPARSE-REWARD OBJECT-INTERACTION TASKS IN FIRST-PERSON SIMULATED 3D ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

First-person object-interaction tasks in high-fidelity, 3D, simulated environments such as the AI2Thor virtual home-environment pose significant sample-efficiency challenges for reinforcement learning (RL) agents learning from sparse task rewards. To alleviate these challenges, prior work has provided extensive supervision via a combination of reward-shaping, ground-truth object-information, and expert demonstrations. In this work, we show that one can learn object-interaction tasks from scratch without supervision by learning an attentive object-model as an auxiliary task during task learning with an object-centric relational RL agent. Our key insight is that learning an object-model that incorporates object-relationships into forward prediction provides a dense learning signal for unsupervised representation learning of both objects and their relationships. This, in turn, enables faster policy learning for an object-centric relational RL agent. We demonstrate our agent by introducing a set of challenging object-interaction tasks in the AI2Thor environment where learning with our attentive object-model is key to strong performance. Specifically, we compare our agent and relational RL agents with alternative auxiliary tasks to a relational RL agent equipped with ground-truth object-information, and show that learning with our object-model best closes the performance gap in terms of both learning speed and maximum success rate. Additionally, we find that incorporating object-attention into an object-model's forward predictions is key to learning representations which capture object-category and object-state.

1 INTRODUCTION

Consider a robotic home-aid agent given the task of cooking a potato. In order to perform this task, it needs to transport both a potato and a pot to the stove and place them appropriately so that turning on the stove will heat the pot, which in turn will cook the potato. Learning to perform such object-interaction tasks is challenging for a number of reasons: (A) the agent needs to transport objects to each other so they can be used together; (B) it needs to learn a view-invariant representation to facilitate object-recognition and object-selection; (C) it needs to recognize combinations of objects that affect each other; (D) it needs to learn to discriminate objects across object-state transitions. In our example, the agent must recognize the pot and potato from different viewpoints, pick them up, transport them to the stove, place the potato in the pot and turn on the stove, recognizing that this will heat both objects. In learning and performing such skills, manipulating object relationships and appropriately responding to them is paramount.

In order to study object-interaction tasks like the one above, we adopt the virtual home-environment AI2Thor (Kolve et al., 2017) (or *Thor*). Thor is an open-source environment that is high-fidelity, 3D, partially observable, and enables object-interactions. Thor poses significant learning challenges due to a large action space induced by many available object-interaction and navigation actions, and a relatively complex first-person visual input. No work has yet learned sparse-reward object-interaction tasks without imitation learning in this domain. Prior work has relied extensively on supervision in the form of reward-shaping, ground-truth object-information, or from expert demonstrations (Jain et al., 2019; Gordon et al., 2018; Zhu et al., 2017; Shridhar et al., 2019). We assume no access to supervision. Instead, we find that we can learn tasks without imitation learning by incorporating attention and an object-centric model into a reinforcement learning agent.

Our primary conceptual contribution is *ROMA*, a Relational, Object-Model Learning Agent. ROMA is composed of a base relational object-centric policy (*Relational Object-DQN*, §4.1) that leverages attention to integrate information about other important objects when estimating the action-value for interacting with a particular object. Without ground-truth information to identify objects, ROMA must learn object-representations that are both *invariant* across object-views and *discriminative* across object-states so that correct actions are chosen as object-states change over a task. To address this and the representation learning challenge induced by a sparse reward signal, ROMA learns an Attentive Object-Model (§4.2) with a contrastive learning loss (Arora et al., 2019; Sohn, 2016) that seeks to separate object-interactions based on their resultant transition. Critically, by incorporating object-attention into the model's forward predictions for a particular object, the object-attention is trained to attend to relevant objects that help predict how the interaction object will change.

We evaluate ROMA against Relational Object-DQN combined with alternative representation learning methods, and show that ROMA best closes the performance gap to an agent supplied with ground-truth object-information. Specifically, we find that learning an attentive object-model is key to achieving both the learning speed and the maximum success rate of the ground-truth-information-supplied agent. By analyzing the object-representations learned by various auxiliary tasks, we find that incorporating object-attention into forward prediction is key to learning representations discriminative of object-category and object-state, which we hypothesize is the source of our strong performance.

In summary, the key contributions of our proposal are: (1) ROMA: an RL agent that demonstrates how to learn sparse-reward object-interaction tasks with egocentric vision without imitation learning or access to other supervision. (2) a novel Attentive Object-Model that bootstraps representation learning of objects and attention over them. (3) Relational Object-DQN: a novel relational RL architecture for object-centric observation- and action-spaces.

2 Related Work

Reinforcement learning for 3D, egocentric object-interaction tasks. Jain et al. (2019) formulate a multi-agent reinforcement learning problem where agents coordinate picking up a large object, and Gordon et al. (2018) developed a hierarchical reinforcement learning agent for visual question-answering. In contrast to our work, both provide strong supervision from expert trajectories. The work most closely related to ours is Oh et al. (2017) (in Minecraft) and Zhu et al. (2017) (in Thor). Both develop a hierarchical reinforcement learning agent where a meta-controller provides goal object-interactions for a low-level controller to complete using ground-truth object-information. Both provide agents with knowledge of all objects, both assume lower-level policies pretrained to navigate to objects and to select interactions with a desired object. In contrast, we do not provide the agent with any ground-truth object information; nor do we pretrain navigation to objects or selection of them.

Object-models for improved sample-efficiency. Prior work here has focused on how an object-model can improve sample-efficiency in a model-based reinforcement learning setting by enabling superior planning (Ye et al., 2020; Veerapaneni et al., 2020; Watters et al., 2019). In contrast, we do not use our model for planning and instead show that it can be leveraged for object-representation and object-attention learning to support faster policy learning in a model-free setting. Our attentive object-model is most similar to the Contrastive Structured World Model (CSWM) (Kipf et al., 2019), which also uses contrastive learning to learn an object-model. However, they use a graph neural network (GNN) to learn relations between all objects in a scene, whereas we employ attention to attend to the objects most important for predicting dynamics. We note that these benefits are orthogonal and could be combined in future work for a GNN-based object-model with reduced connectivity between objects. We also note that they applied their model towards video-prediction and not reinforcement learning.

Relational RL. Most work here has used hand-designed relational representations. Xu et al. (2020) showed improved sample-efficiency, Zaragoza et al. (2010) showed improved policy quality, and Van Hoof et al. (2015) showed generalization to unseen objects. In contrast, we seek to learn object-relations implicitly via attention without our network. Most similar to our work is Zambaldi et al. (2018)–which applies attention to the feature vector outputs of a CNN. They then apply a max-pooling operation to produce a vector input for a policy that selects a coordinate on a screen with a corresponding modifier action. In this work, Relational Object-DQN is a novel architecture extension for a setting with an object-centric observation- and action-space. Additionally, we show that attention can further be exploited by an object-model to improve sample-efficiency.

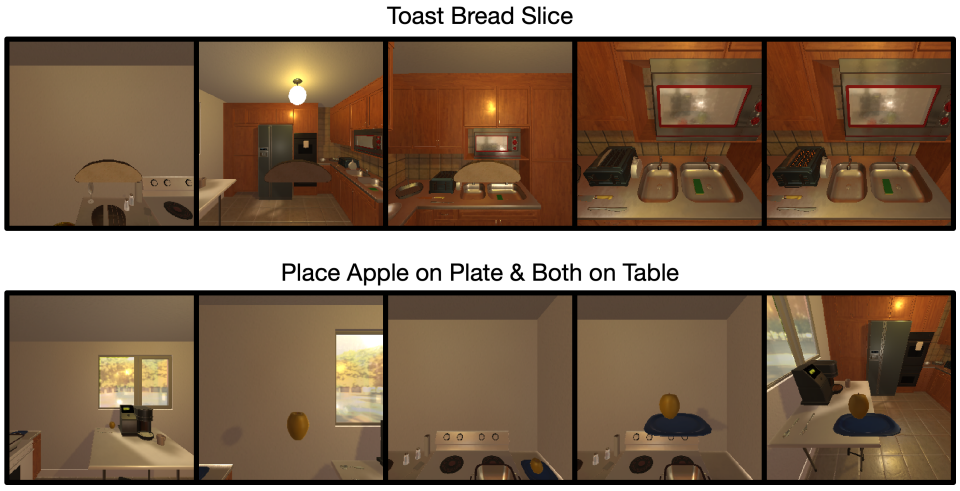

Figure 1: We present the steps required to complete two of our tasks. In the top panel, we present "Toast Bread Slice", where an agent must pickup a bread slice, bring it to the toaster, place it in the toaster, and turn the toaster on. In order to complete the task, the agent needs to recognize the toaster across angles, and it needs to recognize that when the bread is inside the toaster, turning the toaster on will cook the bread. In the bottom panel, we present "Place Apple on Plate & Both on Table", where an agent must pickup an apple, place it on a plate, and move the plate to a table. Like "Toast Bread Slice", it must recognize that because the objects are combined, moving the plate to the table will also move the apple. We observe that learning to use objects together such as in the tasks above poses a representation learning challenge – and thus policy learning challenge – when learning from only a task-completion reward.

3 Sparse-Reward Object-Interaction Tasks in a First-Person Simulated 3D environment

Observations. We focus on an embodied agent that has a 2D camera for experiencing *egocentric* observations $x^{\texttt{ego}}$ of the environment. Our agent also has a pretrained vision system that enables it to extract bounding box image-patches corresponding to the visible objects in its observation $X^o = \{x^{o,i}\}$ (but not object labels or identifiers). We also assume that the agent has access to its (x, y, z) location and body rotation $(\varphi_1, \varphi_2, \varphi_3)$ in a global coordinate frame, $x^{\texttt{loc}} = (x, y, z, \varphi_1, \varphi_2, \varphi_3)$.

Actions. In this work, we focus on the Thor environment. Here, the agent has 8 base object-interactions: $\mathcal{I} = \{$*Pickup, Put, Open, Close, Turn on, Turn off, Slice, Fill*$\}$. The agent interacts with objects by selecting (object-image-patch, interaction) pairs $a = (b, x^{o,c}) \in \mathcal{I} \times X^o$, where $x^{o,c}$ corresponds to the *chosen* image-patch. For example, the agent can Turn on the stove by selecting the image-patch containing the stove-knob and the *Turn on* interaction (see Figure 2 for a diagram). Each action is available at every time-step and can be applied to all objects (i.e. no affordance information is given/used). Interactions occur over one time-step, though their effect may occur over multiple. For the example above, when the agent applies "Turn on" to the stove knob, food on the stove will take several time-steps to heat.

In addition to object-interactions, the agent can select from 8 base navigation actions: $\mathcal{A}_N = \{$*Move ahead, Move back, Move right, Move left, Look up, Look down, Rotate right, Rotate left*$\}$. With $\{$*Look up, Look down*$\}$, the agent can rotate its head up or down in increments of $30°$ between angles $\{0°, \pm 30°, \pm 60°\}$ where $0°$ represents looking straight ahead. With $\{$*Rotate Left, Rotate Right*$\}$, the agent can rotate its body by $\{\pm 90^o\}$. While we forgo the complexity of learning to control actuators required for locomotion and dexterous manipulation, by defining interactions over visible objects, we maintain the complexity of needing to recognize and decide between visible objects and navigating towards other objects.

Reward. We consider a single-task setting where the agent receives a terminal reward of 1 upon task-completion. Due to the large action-space our agent acts in, this leads the agent to face a *sparse reward* problem. Please see appendix C for a description of the tasks we study.

4 ROMA: Relational, Object-Model Learning Agent

ROMA is a reinforcement learning agent composed of an object-centric relational policy, Relational Object-DQN, and an Attentive Object-Model. ROMA uses 2 perceptual modules. The first, $f^o_{\texttt{enc}}$, takes

in an observation x and produces object-encodings $\{\boldsymbol{z}^{o,i}\}_{i=1}^{n}$ for the n visible object-image-patches $X^o = \{x^{o,i}\}_{i=1}^{n}$, where $\boldsymbol{z}^{o,i} \in \mathbb{R}^d_o$. The second, $f^{\kappa}_{\texttt{enc}}$, takes in the egocentric observation and location $x^{\kappa} = [x^{\texttt{ego}}, x^{\texttt{loc}}]$ to produce the *context* for the objects $\boldsymbol{z}^{\kappa} \in \mathbb{R}^d_{\kappa}$. ROMA treats state as the union of these variables: $s = \{\boldsymbol{z}^{o,i}\} \cup \{\boldsymbol{z}^{\kappa}\}$. Given object encodings, Relational Object-DQN computes action-values $Q(s, a = (b, x^{o,i}))$ for interacting with an object $x^{o,i}$ and leverages an attention module $\mathcal{R}$ to incorporate information about other objects $x^{o,j\neq i}$ into this computation (see §4.1).

To address the representation learning challenge induced by a sparse-reward signal, object-representations $\boldsymbol{z}^{o,i}$ and object-attention $\mathcal{R}$ are trained to predict object-dynamics with an attentive object-model (see §4.2). We exploit the observation that when ROMA interacts with object-patch $x^{o,c}$, relevant object-patches $x^{o,j\neq c}$ that lead to more accurate object-dynamics predictions also lead to more accurate Q-value esimation. Thus, learning an *attentive* object-model enables faster learning of the Relational Object-DQN policy. See Figure 2 for an overview of the full architecture.

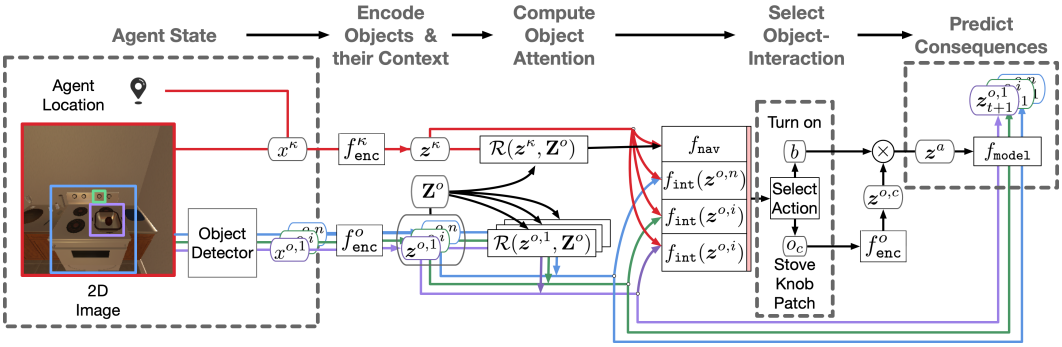

Figure 2: Full architecture and processing pipeline of ROMA. A scene is broken down into object-image-patches $\{x^{o,j}\}$ (e.g. of a pot, potato, and stove knob). The scene image is combined with the agent's location to define the *context* of the objects, x^{κ}. The objects $\{x^{o,j}\}$ and their context x^{κ} are processed by different encoding branches and then recombined by an attention module $\mathcal{R}$ that selects relevant objects for computing Q-value estimates. Here, $\mathcal{R}$ might select the pot image-patch when computing Q-values for interacting with the stove-knob image-patch. Actions are selected as (object-image-patch, base action) pairs $a = (b, x^{o,c})$. The agent then predicts the consequences of its interactions with our attentive object-model $f_{\texttt{model}}$ which reuses $\mathcal{R}$.

4.1 Relational Object-DQN

Relational Object-DQN uses $\widehat{Q}(s, a)$ to estimate the action-value function $Q^{\pi}(s, a) = \mathbb{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t | S_t = s, A_t = a]$, which maps state-action pairs to the expected return on starting from that state-action pair and following policy π thereafter. It leverages $\widehat{Q}$ by behaving according to a policy that is ϵ-greedy w.r.t. $\widehat{Q}(s, a)$: i.e. $\pi(a|s) = \arg\max_a \widehat{Q}(s, a)$ with probability $1 - \epsilon$ and is uniformly random otherwise.

Attending over objects relevant for action-value estimation. In many tasks, an agent must integrate information about multiple objects when estimating Q-values. For example, in the "toast bread" task, the agent needs to integrate information about the toaster and the bread when deciding to turn on the toaster. To accomplish this, we exploit the object-centric observations-space and employ attention Vaswani et al. (2017) to attend to objects that aid in estimating Q-values.

More formally, given an object-encoding $\boldsymbol{z}^{o,i}$, we can use attention to select relevant objects $\mathcal{R}(\boldsymbol{z}^{o,i}, \boldsymbol{Z}^o) \in \mathbb{R}^{d_o}$ for estimating $Q(s, a = (b, x^{o,i}))$. With a matrix of object-encodings, $\boldsymbol{Z}^o = \left[\boldsymbol{z}^{o,i}\right]_i \in \mathbb{R}^{n \times d_o}$, we can perform this computation efficiently for each object-image-patch via:

$$\begin{pmatrix} \mathcal{R}(\boldsymbol{z}^{o,1}, \boldsymbol{Z}^o) \\ \vdots \\ \mathcal{R}(\boldsymbol{z}^{o,n}, \boldsymbol{Z}^o) \end{pmatrix} = \text{Softmax}\left(\frac{(\boldsymbol{Z}^o W^{q_o})(\boldsymbol{Z}^o W^k)^{\top}}{\sqrt{d_k}}\right)\boldsymbol{Z}^o. \tag{1}$$

Here, $\boldsymbol{Z}^o W^{q_o}$ projects each object-encoding to a "query" space and $\boldsymbol{Z}^o W^k$ projects each encoding to a "key" space, where their dot-product determines whether a key is selected for a query. The softmax acts as a soft selection-mechanism for selecting an object-encoding in $\boldsymbol{Z}^o$.

Estimating object-interaction action-values. We can incorporate attention to estimate Q-values for selecting an interaction $b \in \mathcal{I}$ on an object $x^{o,i}$ as follows:

$$\widehat{Q}(s, a = (b, x^{o,i})) = f_{\texttt{int}}([\boldsymbol{z}^{o,i}, \mathcal{R}(\boldsymbol{z}^{o,i}, \boldsymbol{Z}^o), \boldsymbol{z}^{\kappa}]) \in \mathbb{R}^{|\mathcal{I}|} \tag{2}$$

Importantly, this enables us to compute Q-values for a variable number of *unlabeled* objects.

Estimating navigation action-values. We can similarly incorporate attention to compute Q-values for navigation actions by replacing $\boldsymbol{Z}^o W^{q_o}$ with $(W^{q_\kappa} \boldsymbol{z}^{\kappa})^\top$ in equation 1. This then selects object-image-patches in the observation important for selecting a navigation action (e.g. the agent might want to move ahead if it sees bread). We estimate Q-values for navigation actions $b \in \mathcal{A}_N$ as follows:

$$\widehat{Q}(s, a = b) = f_{\texttt{nav}}([\boldsymbol{z}^{\kappa}, \mathcal{R}(\boldsymbol{z}^{\kappa}, \boldsymbol{Z}^o)]) \in \mathbb{R}^{|\mathcal{A}_N|} \tag{3}$$

Learning. We estimate $\widehat{Q}(s, a)$ as a Deep Q-Network (DQN) by minimizing the following temporal difference objective:

$$\mathcal{L}_{\texttt{DQN}} = \mathbb{E}_{s_t, a_t, r_t, s_{t+1}} \left[||y_t - \widehat{Q}(s_t, a_t; \theta)||^2 \right], \tag{4}$$

where $y_t = r_t + \gamma \widehat{Q}(s_{t+1}, a_{t+1}; \theta_{\texttt{old}})$ is the target Q-value, and $\theta_{\texttt{old}}$ is an older copy of the parameters θ. To do so, we store trajectories containing transitions (s_t, a_t, r_t, s_{t+1}) in a replay buffer that we sample from Mnih et al. (2015). To stabilize learning, we use Double-Q-learning Van Hasselt et al. (2016) to choose the next action: $a_{t+1} = \arg\max_a \widehat{Q}(s_{t+1}, a; \theta)$.

4.2 Attentive Object-Model

Discriminating object-states based on their source interaction. In order to successfully complete a task, an agent needs to learn object-representations that are *discriminative* across object-states so that correct actions are chosen as object-states change over a task. To address this, we learn an object-model with a contrastive learning loss that contrasts transitions for a query object based on a given object-interaction.

Consider the global set of objects $\{o^g_{t,i}\}_{i=1}^m$, where m is the number of objects in the environment. At each time-step, each object-image-patch the agent observes corresponds to a 2D projection of $o^g_{t,i}$, $\rho(o^g_{t,i})$ (or $\rho^{g,i}_t$ for short). Given, an object-image-patch, $\rho^{g,i}_t$ and a performed interaction a_t, we can define an object-model as $F(X^o_t, \rho^{g,i}_t, a_t)$ that produces the resultant encoding for $\rho^{g,i}_{t+1}$. We want $F(X^o_t, \rho^{g,i}_t, a_t)$ to be closer to $\rho^{g,i}_{t+1}$ than to encodings of other object-image-patches.

Learning problem. We can formalize this by setting up a classification problem. For an object-image-patch $\rho^{g,i}_t$, we define its *anchor* (or query) by our object-model $F(X^o_t, \rho^{g,i}_t, a_t)$. We define the *positive* (or correct answer) as the encoding of a visible object-image-patch at the next time-step with the highest cosine similarity to the original encoding: $\boldsymbol{z}^{o,i}_+ = \arg\max_{\boldsymbol{z}^{o,j}_{t+1}} \cos(\boldsymbol{z}^{o,i}_t, \boldsymbol{z}^{o,j}_{t+1})$. We can then select K random object-encodings $\{\boldsymbol{z}^{o,i}_{k,-}\}_{k=1}^K$ as *negatives* (or incorrect answers). This leads to:

$$p(\rho^{g,i}_{t+1} | X^o_t, a_t) = \frac{\exp(F(X^o_t, \rho^{g,i}_t, a_t)^\top \boldsymbol{z}^{o,i}_+)}{\exp(F(X^o_t, \rho^{g,i}_t, a_t)^\top \boldsymbol{z}^{o,i}_+) + \sum_k \exp(F(X^o_t, \rho^{g,i}_t, a_t)^\top \boldsymbol{z}^o_{k,-})}. \tag{5}$$

The set of indices corresponding to visible objects at time t is $v_t = \{i : \rho^{g,i}_t \text{ is visible at time } t\}$. Assuming the probability of each object's next state is conditionally independent given the current set of objects and the action taken, we arrive at the following objective:

$$\begin{aligned} \mathcal{L}_{\texttt{model}} &= \mathbb{E}_{s_t, a_t, s_{t+1}} \left[-\log p(X^o_{t+1} | X^o_t, a_t) \right] \\ &= \mathbb{E}_{s_t, a_t, s_{t+1}} \left[-\sum_{i \in v_{t+1}} \log p(\rho^{g,i}_{t+1} | X^o_t, a_t) \right]. \\ \mathcal{L} &= \mathcal{L}_{\texttt{DQN}} + \beta^{\texttt{model}} \mathcal{L}_{\texttt{model}}. \end{aligned} \tag{6}$$

Leveraging object-attention for improved accuracy. Consider slicing an apple with a knife. When selecting "slice" on the apple patch, $\mathcal{R}(\boldsymbol{z}^{o,i}, \boldsymbol{Z}^o)$ must the select the knife patch to accurately estimate the Q-values via equation 2. We observe that $\mathcal{R}(\boldsymbol{z}^{o,i}, \boldsymbol{Z}^o)$ can also be employed by our object-model

$F(X_t^o, \rho_t^{g,i}, a_t)$ to support higher classification accuracy when predicting how the object-encoding will change post interaction. We can incorporate this to obtain an *attentive object-model* as follows:

$$F(X_t^o, \rho_t^{g,i}, a_t) = f_{\texttt{model}}([\boldsymbol{z}_t^{o,j}, \mathcal{R}(\boldsymbol{z}_t^{o,j}, \boldsymbol{Z}_t^o), \boldsymbol{z}_t^a]). \tag{7}$$

To learn an action encoding $\boldsymbol{z}_t^a$ for action a_t, following Oh et al. (2015); Reed et al. (2014), we employ multiplicative interactions so our learned action representation $\boldsymbol{z}_t^a$ compactly models the cartesian product of all base actions b and object-image-patch selections o_c as

$$\boldsymbol{z}_t^a = W^o \boldsymbol{z}_t^{o,c} \odot W^b b_t, \tag{8}$$

where $W^o \in \mathbb{R}^{d_a \times d_o}$, $W^b \in \mathbb{R}^{d_a \times |\mathcal{A}_I|}$, and $\odot$ is an element-wise hadamard product. In practice, $f_{\texttt{model}}$ is a small 1- or 2-layer neural network making this method compact and simple to implement.

5 EXPERIMENTS

The primary aim of our experiments is to examine how an object-centric observation- and action-space can best be exploited to improve sample-complexity for sparse-reward object-interaction tasks. We find that we can learn such tasks without imitation learning by imbuing our object-centric relational Q-learner with rich hand-designed object-representations. We study strong unsupervised object-representation learning techniques and find that learning to predict object dynamics with our attentive object-model best matches the success rate obtained using rich hand-designed features. The second aim of our experiments is to study the object representations learned by each method. We find quantitative evidence that learning with our attentive object-model yields representations that best discriminate object category, object state and object relations.

5.1 EVALUATION TASKS AND REWARD FUNCTION

Using common kitchen activities as inspiration, we constructed 8 tasks in the Thor environment that require an agent learn to collect objects and use them together using only a sparse task-completion reward. These tasks vary in their optimal length, the visually complexity of task objects, the number of objects to be used together, and in whether objects must be combined for further use. Across tasks, the agent's spawning location is randomized from 81 grid positions. The agent receives reward 1 if a task is completed successfully and a time-step penalty of -0.04. The agent has a budget of $500K$ samples to learn a task. We found that this was the budget needed by a relational agent with oracle object-information. We report results on the 8 tasks that had the highest optimal length. See descriptions of the tasks in appendix C.

5.2 BASELINE METHODS FOR COMPARISON

In order to study the effects of competing object representation learning methods, we compare combining Relational Object-DQN with the Attentive Object-Model against four baseline methods:

1. **No-Auxiliary Task**. This method has no representation learning method and lets us study how well an agent can learn from the sparse-reward signal alone.
2. **Ground-Truth Object-Information**. This method has no auxiliary task. Instead, we supply the agent with the following simulator information: object-category (i.e. an object's type), object-id (i.e. an object's token), object-state (e.g. on, off, etc.), and object-containment (i.e. is the object in/on another object and if so, which object). We found that this hand-designed object-representation enabled Relational Object-DQN to learn all our tasks within our sample-budget and it is our basis for comparing unsupervised object-representation learning methods.
3. **OCN**. The Object Contrastive Network (Pirk et al., 2019). This method also employs contrastive learning but seeks to cluster object-images across time-steps. This enables us to study whether contrasting object-transitions is more effective than contrasting object-images.
4. **COBRA Object-Model**. This is a *non-attentive* object-model employed by the COBRA RL agent (Watters et al., 2019). They also targeted improved sample-efficiency—though in a simpler, fully-observable 2D environment with basic shapes. This lets us verify our claim that incorporating attention into an object-model further improves sample-efficiency. We adapt COBRA's object-model to assume and predict object-image-patches.

5.3 LEARNING RESULTS

Metrics. We evaluate agent performance by measuring the agent's success rate over 5K frames every 25K frames of experience. The success rate is the proportion of episodes that the agent completes. We compute the mean and standard error of these values across 5 seeds. To study sample-efficiency, we compare each method to "Ground-Truth Object-Information" by computing what percent of

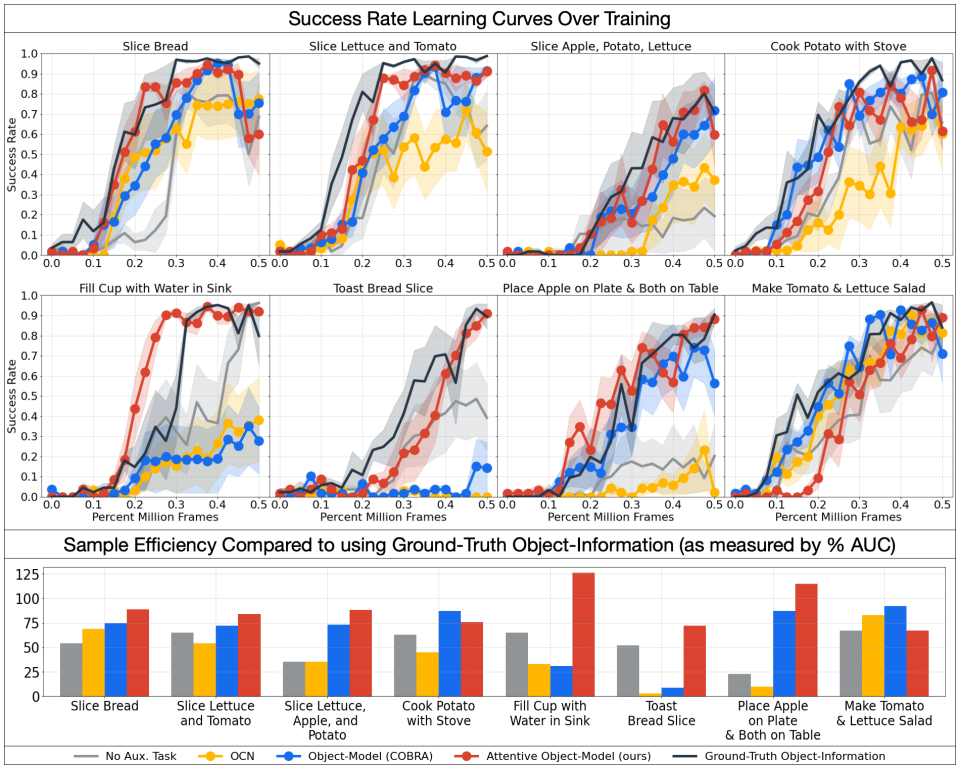

Figure 3: **Top-panel**: we present the success rate over learning for competing auxiliary tasks. We seek a method that best enables our Relational Object-DQN (grey) to obtain the sample-efficiency it would from adding Ground-Truth Object-Information (black). **Bottom-panel**: by measuring the % AUC achieved by each agent w.r.t to the agent with ground-truth information, we can more precisely measure how close each method is to ground-truth performance. We find learning with an Attentive Object-Model (ROMA, red) best closes the performance gap on $6/8$.

the Ground-Truth Object-Information mean success rate success rate AUC each method achieved. We present the success rate learning curves and a sample-efficiency bar-graph in Figure 3. We additionally present the maximum success rate achieved by each method in Table 1.

Performance. Looking at Table 1, we find that using Ground-Truth Object-Information is able to get the highest success rate on $7/8$ tasks and achieves a 90+% success rate for $7/8$ tasks. It, and all other methods, achieve 80% on "Slice Apple, Potato, Lettuce", a task that requires using 4 objects. We find that tasks that require more objects have a higher sample-complexity. No-Auxiliary Task performs below all methods besides OCN on $7/8$ tasks. Surprisingly, No-Auxiliary Task outperforms OCN on $5/8$ tasks. OCN learns an embedding space where object-images across time-steps are clustered together. We hypothesize that this leads it to lose the ability to discriminate object-states, which is important for our tasks.

auxiliary Task	Slice Bread	Slice Lettuce and Tomato	Slice Apple, Potato, Lettuce	Cook Potato on Stove	Fill Cup with Water	Toast Bread Slice	Apple on Plate, Both on Table	Make Salad
No-Auxiliary Task	80.6 ± 7.8	89.6 ± 3.0	23.5 ± 14.7	80.6 ± 12.9	$\mathbf{96.3 \pm 0.7}$	48.5 ± 18.2	20.2 ± 16.1	81.4 ± 7.3
Ground-Truth Object-Info	$\mathbf{98.6 \pm 0.2}$	$\mathbf{98.8 \pm 0.2}$	$\mathbf{80.2 \pm 10.8}$	$\mathbf{97.7 \pm 0.2}$	95.2 ± 0.4	$\mathbf{93.3 \pm 2.3}$	$\mathbf{90.5 \pm 3.2}$	$\mathbf{96.6 \pm 0.2}$
OCN	77.6 ± 13.9	72.0 ± 15.1	43.4 ± 16.3	70.9 ± 9.6	38.2 ± 20.9	3.0 ± 1.9	23.2 ± 18.0	90.2 ± 1.8
COBRA Object-Model	$\mathbf{95.3 \pm 1.2}$	$\mathbf{93.4 \pm 1.4}$	71.7 ± 16.2	88.6 ± 3.3	35.0 ± 19.3	15.1 ± 13.5	74.1 ± 14.8	$\mathbf{92.7 \pm 1.4}$
Attentive Object-Model	$\mathbf{94.4 \pm 1.8}$	$\mathbf{94.2 \pm 0.5}$	$\mathbf{81.9 \pm 4.4}$	$\mathbf{91.7 \pm 2.3}$	$\mathbf{94.5 \pm 1.0}$	$\mathbf{91.1 \pm 2.1}$	$\mathbf{88.1 \pm 3.4}$	$\mathbf{92.8 \pm 0.5}$

Table 1: Maximum success rate achieved by competing auxiliary tasks during training.

In terms of maximum success rate, looking at Table 1, our Attentive Object-Model comes closest to Ground-Truth Object-Information on $5/8$ tasks and is tied on $3/8$ tasks with the COBRA Object-

Model. However, for tasks that require using objects together, such as "Fill Cup with Water" where a cup must be used in a sink or "Toast Bread Slice" where bread must be cooked in a toaster, the COBRA Object-Model exhibits a higher sample-complexity. In terms of sample-efficiency, our Attentive Object-Model comes closest to Ground-Truth Object-Information on $4/8$ tasks and learns more quickly on $2/8$ tasks. We suspect that this is due to its ability to bootstrap object-attention.

5.4 Analysis of Learned Object Representations

We conjecture that incorporating attention into an object-model is key to having it accurately model object-transitions and thus learn object-representations that are discriminative of object-category, object-state, and object-relations. To probe these representations, we analyze each agent's object-encoding function $f^o_{\texttt{enc}}$. We freeze their parameters, and add a linear layer to predict some of the Ground-Truth Object-Information features using a dataset of collected object-interactions we construct. The dataset contains (s, a, s') tuples from an oracle agent we create to complete a tasks such as "*Cook X with Stove*", where $X \in \{\text{Potato, Potato Slice, Cracked Egg}\}$. We study the following features: *Category* is a multi-class label indicating an object's category. The following are binary labels. *Object-State* indicates whether objects *are* closed, turned on, etc. *Containment Relationship* indicates if an object is inside another object or has another object inside of it. For each feature, we present the mean average precision and standard error for each method across all 8 tasks in Table 2. See Appendix C.3 for more details. We find that an attentive object-model best captures these features, which we hypothesize drives our strong task-learning performance.

Representation Learning Method	Category	Object-State	Containment Relationship
OCN	39.2 ± 8.2	66.5 ± 8.5	69.1 ± 9.0
COBRA Object-Model	79.8 ± 2.8	73.4 ± 8.9	83.1 ± 5.8
Attentive Object-Model	$\mathbf{88.6 \pm 3.5}$	$\mathbf{98.6 \pm 0.3}$	$\mathbf{94.3 \pm 0.6}$

Table 2: Performance of different unsupervised learning methods for learning object-features (see text above for details). We find that incorporating attention into an object-model is key to capturing object-features useful for task-learning.

5.5 Object-Attention Ablation

We present results for object-DQN *without* a relational bias (i.e. without attention). This amounts to removing $\mathcal{R}(\cdot)$ from equation 2 and equation 3. Without attention, we see that object-DQN has a hard time learning tasks, including for "Fill Cup with Water", which it was able to excel in as seen in Figure 3 and Table 1. This indicates that it's not only good representations that's important for efficient learning but also the ability to relate objects via attention. We note that ROMA can excel in both of these settings.

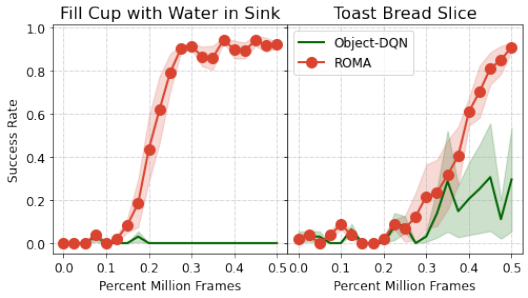

Figure 4: Ablation of Object-Attention. We show results for DQN conditioned only on object-images without object-attention.

6 Conclusion

With ROMA, we have shown that learning a attentive object-model in tandem with a relational object-centric policy can enable sample-efficient learning in high-fidelity, 3D, object-interaction domains without access to expert demonstrations or ground-truth object-information. Further, when compared to strong unsupervised object-representation learning baselines, we have shown that our attentive object-model is able to best capture ground-truth object information such as object categories, states of objects, and the presence of interesting object relationships. We believe ROMA and the components that power it—a relational object-centric policy and an attentive object-model—are promising steps towards agents that can efficiently learn complex object-interaction tasks.

References

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, 2019.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations*, 2017.

Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari, Svetlana Lazebnik, Ali Farhadi, Alexander G Schwing, and Aniruddha Kembhavi. Two body problem: Collaborative visual task completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, 2015.

Junhyuk Oh, Satinder Singh, Honglak Lee, and Pushmeet Kohli. Zero-shot task generalization with multi-task deep reinforcement learning. In *International Conference on Machine Learning*, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Sören Pirk, Mohi Khansari, Yunfei Bai, Corey Lynch, and Pierre Sermanet. Online object representations with contrastive learning. *arXiv preprint arXiv:1906.04312*, 2019.

Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *International Conference on Machine Learning*, 2014.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. *ArXiv*, abs/1912.01734, 2019.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, 2016.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI conference on artificial intelligence*, 2016.

Herke Van Hoof, Tucker Hermans, Gerhard Neumann, and Jan Peters. Learning robot in-hand manipulation with tactile features. In *International Conference on Humanoid Robots*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

Rishi Veerapaneni, John D Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning*, 2020.

David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *arXiv preprint arXiv:1811.11359*, 2018.

Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P Burgess, and Alexander Lerchner. Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration. *arXiv preprint arXiv:1905.09275*, 2019.

Tingting Xu, Henghui Zhu, and Ioannis Ch Paschalidis. Learning parametric policies and transition probability models of markov decision processes from data. *European Journal of Control*, 2020.

Yufei Ye, Dhiraj Gandhi, Abhinav Gupta, and Shubham Tulsiani. Object-centric forward modeling for model predictive control. In *Conference on Robot Learning*, 2020.

Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, et al. Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830*, 2018.

Julio H Zaragoza, Eduardo F Morales, et al. Relational reinforcement learning with continuous actions by combining behavioural cloning and locally weighted regression. *Journal of Intelligent Learning Systems and Applications*, 2(02):69, 2010.

Yuke Zhu, Daniel Gordon, Eric Kolve, Dieter Fox, Li Fei-Fei, Abhinav Gupta, Roozbeh Mottaghi, and Ali Farhadi. Visual semantic planning using deep successor representations. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.