Reinforcement Learning for Sparse-Reward Object-Interaction Tasks in a First-person Simulated 3D Environment

Wilka Carvalho[1*], **Anthony Liang**[1], **Kimin Lee**[2], **Sungryull Sohn**[1], **Honglak Lee**[1,3], **Richard Lewis**[1] and **Satinder Singh**[1]
[1]University of Michigan
[2]UC Berkeley
[3]LG AI Research

Abstract

Learning how to execute complex tasks involving multiple objects in a 3D world is challenging when there is no ground-truth information about the objects or any demonstration to learn from. When an agent only receives a signal from task-completion, this makes it challenging to learn the object-representations which support learning the correct object-interactions needed to complete the task. In this work, we formulate learning an attentive object dynamics model as a classification problem, using random object-images to define incorrect labels for our object-dynamics model. We show empirically that this enables object-representation learning that captures an object's category (is it a toaster?), its properties (is it on?), and object-relations (is something inside of it?). With this, our core learner (a relational RL agent) receives the dense training signal it needs to rapidly learn object-interaction tasks. We demonstrate results in the 3D AI2Thor simulated kitchen environment with a range of challenging food preparation tasks. We compare our method's performance to several related approaches and against the performance of an oracle: an agent that is supplied with ground-truth information about objects in the scene. We find that our agent achieves performance closest to the oracle in terms of both learning speed and maximum success rate.

1 Introduction

Consider a robotic home-aid agent that learns *object-interaction tasks* that involve using multiple objects together to accomplish various tasks such as chopping vegetables or heating meals. Such tasks are important for artificial intelligence (AI) to make progress on because of their large potential to impact our everday world: nursing robots can serve healthcare workers in hospitals, and home-aid robots can help busy families, the disabled, and the elderly.

Prior work on object-interaction tasks has focused on achieving strong training performance using expert demonstrations [Zhu *et al.*, 2017; Shridhar *et al.*, 2019]. Unfortunately, [Zhu *et al.*, 2017] found they were unable to learn relatively simple pick and place tasks when only learning from a sparse task-completion signal. Other work has relaxed the learning problem by relying on domain knowledge in the form of shaped rewards or object-affordance knowledge [Jain *et al.*, 2019; Gordon *et al.*, 2018].

Unfortunately, expert demonstrations and shaped rewards can be challenging to obtain for tasks novel to an agent. Additionally, it can be tedious or impossible to obtain ground-truth information about all novel objects an agent may encounter. Ideally, agents are capable of learning object-interaction tasks without this information. To work towards this, we focus on the setting where none of these are available.

Learning object-interaction tasks without expert demonstrations or shaped rewards is challenging because selecting between object-interactions induces a branching factor that scales with the number of visible objects, leading the agent choose from 50-100 actions at a given time-step. This leads the agent to infrequently experience a successful episode. When the agent does, task completion typically occurs after many hundred time-steps. Consider learning to toast bread. The agent should learn to turn on the toaster after a bread slice is placed inside, i.e. it needs to learn to represent *containment relationships* (the bread is inside the toaster) and *object properties* (the toaster is on or off). Without domain knowledge about objects, task-completion alone provides a weak learning signal for learning both to represent 3D object categories, properties, and relationships. When episodes last for hundreds of time-steps and the agent interacts with many objects, this makes it challenging to learn about about how the agent's object-interactions led to reward.

In this work, we find that we can achieve strong training performance on object-interaction tasks without expert demonstrations, shaped rewards, or ground-truth object-knowledge by incorporating inter-object attention and an object-centric model into a reinforcement learning agent.We call our agent the *Learning Object Attention & Dynamics* (or *LOAD*) agent. LOAD is composed of a base object-centric relational policy (*Attentive Object-DQN*, §4.1) that leverages inter-object attention to incorporate object-relationships when estimating object-interaction action-values. Without ground-truth information to identify object categories, properties, or relationships, LOAD learns object-representations with a novel

*Contact author: wcarvalh@umich.edu

learning objective that frames learning an object-model as a classification problem, where random object-embeddings are incorrect labels (*Attentive Object-Model*, §4.2). By doing so, we provide the object-model with a dense learning signal for learning represent both object categories, but also changes in object-properties caused by different object-interactions. Additionally, by sharing inter-object attention between the policy and the model, learning the model helps drive learning of inter-object attention helpful for speedening task learning.

In order to study object-interaction tasks and evaluate our agent, 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. We show that LOAD is able to significantly reduce sample complexity in this domain where no prior work has yet learned sparse-reward object-interaction tasks without expert demonstrations or shaped rewards.

In our main evaluation, we compare pairing Attentive Object-DQN with our Attentive Object-Model to alternative representation learning methods, and show that learning with our object-model best closes the performance gap to an agent supplied with ground-truth information about object categories, properties, and relationships (§5.1). Through an analysis of the learned object-representations and inter-object attention learned by each auxiliary task, we provide quantitative evidence that our Attentive Object-Model best learns representations that capture the ground-truth information present in our oracle (§5.2). We hypothesize that this is the source of our strong performance. Afterwards, we perform a series of ablations to study the importance of object-representations which capture object-properites and object-relations for reducing sample-complexity (§5.3).

In summary, the key contributions of our proposal are: (1) LOAD: an RL agent that demonstrates how to learn sparse-reward object-interaction tasks with first-persosn vision without expert demonstrations, shaped rewards, or ground-truth object-knowledge. (2) A novel Attentive Object-Model auxilliary task, which frames learning an object-model as a classification problem. With our analysis, we provide evidence that for our 3D, high-fidelty domain and our architecture, it is key to learn object-representations which not only capture object-categories but also object-properties and object-relations.

2 Related Work

Learning Object-interaction Tasks in 3D, First-person Environments. Due to the large branching factor induced by object-interactions, most work here has relied extensively on expert demonstrations [Zhu *et al.*, 2017; Shridhar *et al.*, 2019; Xu *et al.*, 2019] or avoided this problem by hard-coding object-selection [Jain *et al.*, 2019; Gordon *et al.*, 2018]. 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 and 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-Centric Relational RL. An intutive approach to tasks with objects is object-centric relational RL. Most work here has used hand-designed representations of objects and their relations, showing things like improved sample-efficiency [Xu *et al.*, 2020], improved policy quality [Zaragoza *et al.*, 2010], and generalization to unseen objects [Van Hoof *et al.*, 2015]. In contrast, we seek to learn object-representations and object-relations implicitly with our network. Most similar to our work is [Zambaldi *et al.*, 2018]–which applies attention to the feature vector outputs of a CNN. In this work, Attentive Object-DQN is a novel architecture extension for a setting with an object-centric observation- and action-space. Additionally, we show that learning an object-model as an auxilliary task can help drive learning of attention.

Learning an Object-model as an Auxiliary Task. Most prior work here has focused on how an object-model can be used in model-based reinforcement learning by enabling superior planning [Ye *et al.*, 2020; Veerapaneni *et al.*, 2020; Watters *et al.*, 2019]. In contrast, we do not use our object-model for planning and instead show that it can be leveraged to learn object-representation and inter-object attention to support faster policy learning in a model-free setting. Additionally, other work focused on domains where representation-learning only had to differntiate object-categories. We show that our method can additionally differentiate object-properties and does so significantly better than the object-model of [Watters *et al.*, 2019]. Our attentive object-model is most similar to the Contrastive Structured World Model (CSWM) [Kipf *et al.*, 2019], which uses a maximum margin contrastive learning objective [Hadsell *et al.*, 2006] to learn an object-model. Instead, we formulate a novel object-model contrastive objective as learning a classification problem. We note that they applied their model towards video-prediction and not reinforcement learning.

3 Sparse-reward Object-interaction Tasks in a First-person Simulated 3D Environment

Observations. We focus on an 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 $X^o = \{x^{o,i}\}$. Besides boxes around objects, no other information is extracted (i.e. no labels, identifiers, poses, etc.). We assume 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

Slice $\{X_i\}, n \in [1,3]$	Make Toamto & Lettuce Salad	Place Apple on Plate, Both on Table	Cook Potato on Stove	Fill Cup with Water	Toast Bread Slice
(A) recognize knife across angles (B) recognize 2-4 objects	(B) recognize 3 objects (C) use containment: plate with tomato/lettuce slice	(B) recognize 3 objects (C) use containment: apple on plate	(B) recognize 2 objects (C) use containment: potato on stove (D) changing properites: cooked potato	(A) recognize translucent cup across backgrounds (B) recognize 2 objects (C) use containment: cup in sink (D) changing properites: filled cup	(A) recognize toaster across angles (B) recognize 2 objects (C) use containment: bread inside toaster (D) changing properites: cooked bread

Table 1: Description of challenges associated with the tasks we study. See Figure 1 for example panels of 2 tasks.

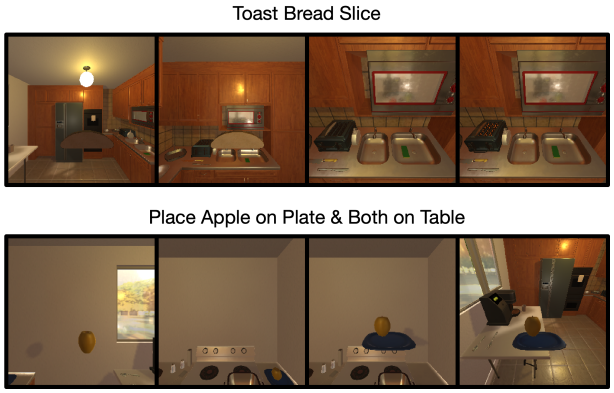


Figure 1: We present the steps required to complete two of our tasks. In "Toast Bread Slice", an agent must pickup a bread slice, bring it to the toaster, place it in the toaster, and turn the toaster on. In order to complete the task, the agent needs to recognize the toaster across angles, and it needs to recognize that when the bread is inside the toaster, turning the toaster on will cook the bread. In "Place Apple on Plate & Both on Table", agent must pickup an apple, place it on a plate, and move the plate to a table. 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.

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 $\{0^\circ, \pm 30^\circ, \pm 60^\circ\}$. 0° represents looking straight ahead. With $\{$*Rotate Left, Rotate Right*$\}$, the agent can rotate its body by $\{\pm 90^o\}$.

Tasks. We construct 8 tasks with the following 4 challenges. Challenge (A): the visual complexity of task objects (e.g. the cup is translucent). Challenge (B): the number of objects to be interacted with (e.g., "Slice Apple, Potato, Lettuce" requires the agent interact with 4 objects). Challenge (C): whether object-containment must be recognized and used (e.g. toasting bread in a toaster). Challenge (D): whether object-properties change (e.g. bread get's cooked). See Figure 1 for a description of the challenges associated with each task and Figure 1 for example panels of 2 tasks.

Reward. We consider a single-task setting where the agent receives a terminal reward of 1 upon task-completion.

4 LOAD: Learning Object Attention & Dynamics Agent

LOAD is a reinforcement learning agent composed of an object-centric relational policy, Attentive Object-DQN, and an Attentive Object-Model. LOAD 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 $\boldsymbol{z}^\kappa \in \mathbb{R}^d_\kappa$. LOAD treats state as the union of these variables: $s = \{\boldsymbol{z}^{o,i}\} \cup \{\boldsymbol{z}^\kappa\}$. Given object encodings, Attentive 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 A 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 A are trained to predict object-dynamics with an attentive object-model (see §4.2). See Figure 2 for an overview of the full architecture.

4.1 Attentive Object-DQN

Attentive 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.

Leveraging Inter-object Attention During 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*

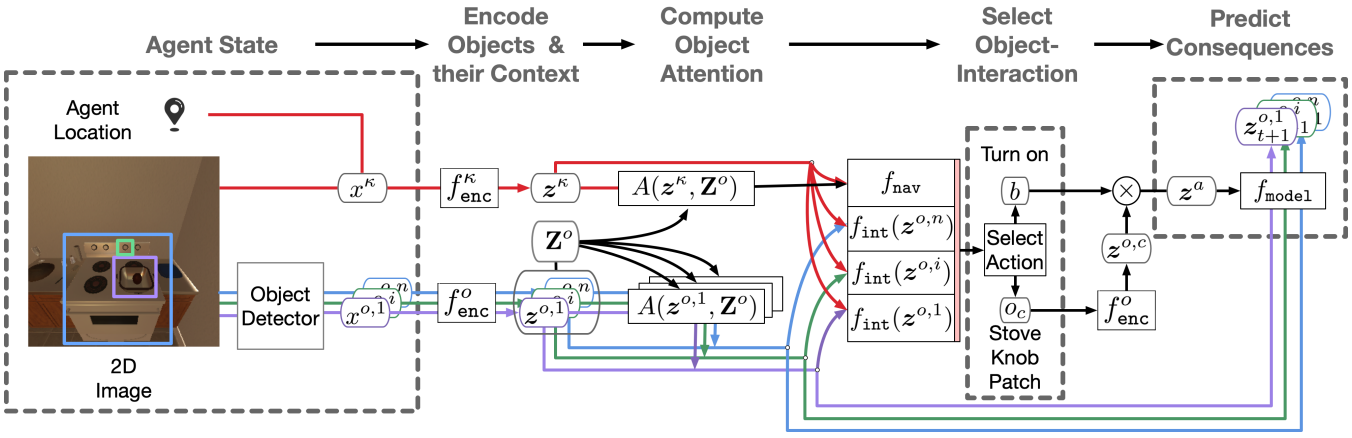


Figure 2: Full architecture and processing pipeline of LOAD. A scene is broken down into object-image-patches $\{x^{o,j}\}$ (e.g. of a pot, potato, and stove knob). The scene image is combined with the agent's location to define the *context* of the objects, x^κ. The objects $\{x^{o,j}\}$ and their context x^κ are processed by different encoding branches and then recombined by an attention module A that selects relevant objects for computing Q-value estimates. Here, A 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 A.

al., 2017] to incorporate inter-object attention into Q-value estimation. More formally, given an object-encoding $\boldsymbol{z}^{o,i}$, we can use attention to select relevant objects $A(\boldsymbol{z}^{o,i}, \boldsymbol{Z}^o) \in \mathbb{R}^{d_o}$ 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:

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

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 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:

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

Importantly, this enables us to compute Q-values for a variable number of *unlabeled* objects. We can similarly incorporate attention to compute Q-values for navigation actions by replacing $\boldsymbol{Z}^o W^{q_o}$ with $\left(W^{q_\kappa}\boldsymbol{z}^\kappa\right)^\top$ in equation 1. We estimate Q-values for navigation actions $b \in \mathcal{A}_N$ as follows:

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

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-Dynamics Model

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) and encodes it as $\boldsymbol{z}^{g,i}_t$. Given, an object-image-patch encoding $\boldsymbol{z}^{g,i}_t$ and a performed interaction a_t, we can define an object-dynamics model $D(\boldsymbol{Z}^o_t, \boldsymbol{z}^{g,i}_t, a_t)$ which produces the resultant encoding for $\rho^{g,i}_{t+1}$. We want $D(\boldsymbol{Z}^o_t, \boldsymbol{z}^{g,i}_t, a_t)$ to be closer to $\boldsymbol{z}^{g,i}_{t+1}$ than to encodings of other object-image-patches.

Classification Problem. We can formalize this by setting up a classification problem. For an object-image-patch encoding $\boldsymbol{z}^{g,i}_t$, we define the *prediction* as the output of our object-dynamics model $D(\boldsymbol{Z}^o_t, \boldsymbol{z}^{g,i}_t, a_t)$. We define the *label* as the encoding of a visible object-image-patch at the next time-step with the highest cosine similarity to the original encoding $\boldsymbol{z}^{g,i}_+ = \arg\max_{\boldsymbol{z}^{g,j}_{t+1}} \cos(\boldsymbol{z}^{g,i}_t, \boldsymbol{z}^{g,j}_{t+1})$. We can then select K random object-encodings $\{\boldsymbol{z}^o_{k,-}\}_{k=1}^K$ as *incorrect labels*. Rewriting $D(\boldsymbol{Z}^o_t, \boldsymbol{z}^{g,i}_t, a_t)$ as D, this leads to:

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

The set of indices corresponding to visible objects at time t is $v_t = \{i : \rho^{g,i}_t \text{ is visible at time } t\}$. The set of observed object-image-patch encodings is then $\boldsymbol{Z}^o_t = \{\boldsymbol{z}^{o,j}_t\} = \{\boldsymbol{z}^{g,i}_t\}_{i \in v_t}$.

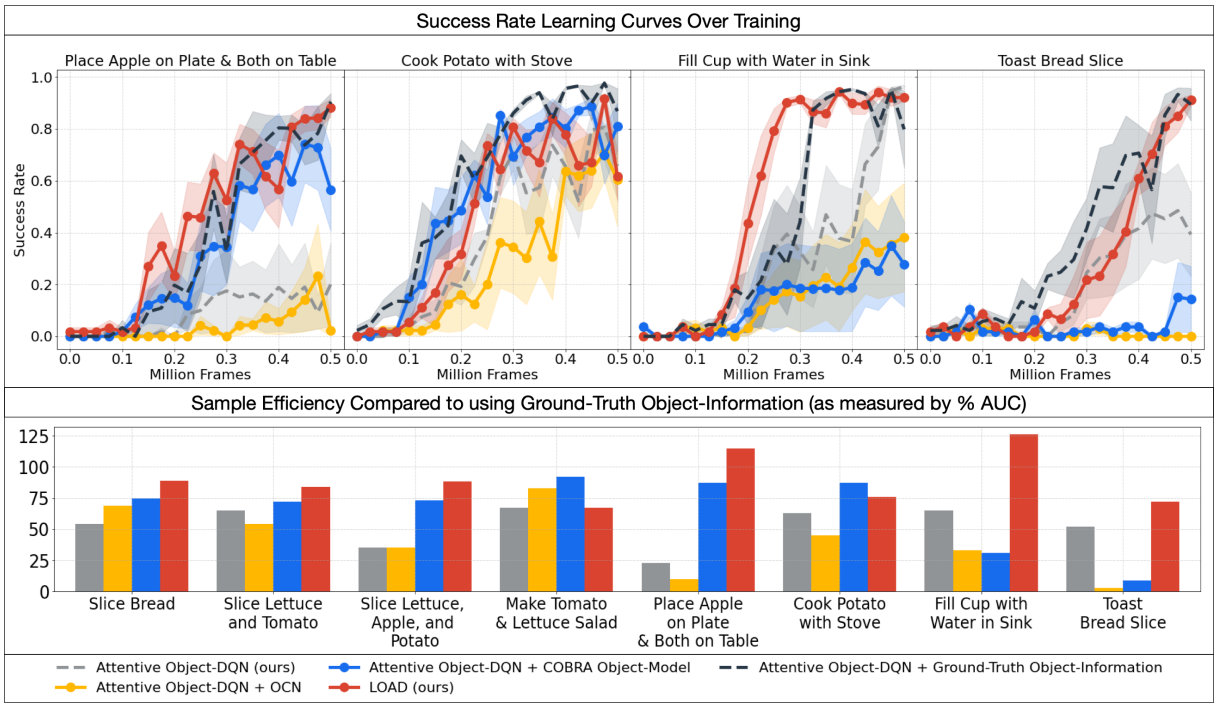


Figure 3: **Top-panel**: we present the success rate over learning for competing auxiliary tasks. We seek a method that best enables our Attentive Object-DQN (grey) to obtain the sample-efficiency it would from adding Ground-Truth Object-Information (black). We visually see that LOAD (red) is best able to learn more quickly on tasks that require using containment-relationships (e.g. a cup in a sink) or recognizing changing object properties (e.g. a toaster turning on with bread in it).
Bottom-panel: by measuring the % AUC achieved by each agent w.r.t to the agent with ground-truth information, we can measure how close each method is to the performance of an agent with ground-truth object-knowledge. We find LOAD (red), which learns an attentive object-model best closes the performance gap on $6/8$. We hypothesize that this is due to our object-model's ability to capture oracle object-information about object-categories, object-properties, and object-relations. We show evidence for this in Table 2.

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:

$$\begin{aligned}\mathcal{L}_{\texttt{model}} &= \mathbb{E}_{z_t,a_t,z_{t+1}}\left[-\log p(\boldsymbol{Z}^o_{t+1}|\boldsymbol{Z}^o_t, a_t)\right] \\ &= \mathbb{E}_{z_t,a_t,z_{t+1}}\left[-\sum_{i\in v_{t+1}} \log p(z^{g,i}_{t+1}|\boldsymbol{Z}^o_t, a_t)\right]. \end{aligned} \tag{6}$$

Our final objective becomes:

$$\mathcal{L} = \mathcal{L}_{\texttt{DQN}} + \beta^{\texttt{model}}\mathcal{L}_{\texttt{model}}. \tag{7}$$

Leveraging Inter-object Attention for Improved Accuracy. Consider slicing an apple with a knife. When selecting "slice" on the apple patch, learning to attend to the knife patch **both** enables more accurate estimation of Q-values and higher model-prediction accuracy. We can accomplish this by incorporating $A(\boldsymbol{z}^{g,i}, \boldsymbol{Z}^o)$ into our object-model as follows:

$$D(\boldsymbol{Z}^o_t, \boldsymbol{z}^{g,i}_t, a_t) = f_{\texttt{model}}([\boldsymbol{z}^{g,i}_t, A(\boldsymbol{z}^{g,i}_t, \boldsymbol{Z}^o_t), \boldsymbol{z}^a_t]). \tag{8}$$

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

$$\boldsymbol{z}^a_t = W^o \boldsymbol{z}^{g,c}_t \odot W^b b_t, \tag{9}$$

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 study how different auxilliary tasks for learning object-representations enable sample complexity comparable to an agent with oracle object-knowledge. We additionally study the degree to which each auxilliary task enables object-representation learning that captures the ground-truth knowledge present in our oracle agent. We conclude this section with ablation experiments studying the importance of different forms of object-knowledge in task learning.

Evaluation Settings. The agent's spawning location is randomized from 81 grid positions. The agent receives a terminal reward of 1 if its task is completed successfully and 0 otherwise. It receives a time-step penalty of -0.04. Episodes have a time-limit of 500 time-steps. The agent has a budget of $500K$ samples to learn a task. This was the budget needed by a relational agent with oracle object-information.

Baseline Methods for Comparison. In order to study the effects of competing object representation learning methods,

Representation Learning Method	Category	Object-Properties	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
LOAD	$\mathbf{88.6 \pm 3.5}$	$\mathbf{98.6 \pm 0.3}$	$\mathbf{94.3 \pm 0.6}$

Table 2: Performance of different unsupervised learning methods for learning object-features (see §5.2 for details). We find that our object-model best captures features present in the oracle agent, providing evidence that its strong object-representation learning is responsible for its strong task-learning performance.

we compare combining Attentive Object-DQN with the Attentive Object-Model against four baseline methods:

1. **Attentive Object-DQN**. This baseline has no auxiliary task and lets us study how well an agent can learn from the sparse-reward signal alone.
2. **Ground-Truth Object-Information**. This baseline has no auxiliary task. Instead, we supply the agent with 14 ground-truth features from the simulator. They roughly describe an object's category (is it a toaster?), its properties (e.g., is it on/off/etc.?), and relevant object-containment (e.g., what object is this object inside of?). We found that this hand-designed object-representation enabled Attentive Object-DQN to learn all our tasks within our sample-budget and it is our basis for comparing unsupervised object-representation learning methods. Please see §A.1 for detailed descriptions of these features.
3. **OCN**. The Object Contrastive Network [Pirk *et al.*, 2019]. This method also employs a classification-like contrastive learning objective to cluster object-images across time-steps. However, it doesn't use an object-model or incorporate action-information. This enables us to study the importance of incoporating an object-model and action information.
4. **COBRA Object-Model**. This is the 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 shapes that only needed differentiation by category. Their model had no mechanism for incorporating inter-object relations into its predictions.

To enable faster learning in a sparse-reward setting, all baselines sample training batches using a second self-imitation learning replay buffer of successful episodes [Oh *et al.*, 2018].

5.1 Task Performance

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 the Ground-Truth Object-Information mean success rate AUC each method achieved.

Performance

We present sample-efficiency bar plots for all 8 of our tasks in Figure 3. We found that using containment relationships and recognizing changing object-properties (Challenges C & D in §3) were most indicative of task difficulty. We only present learning curve results for 4 tasks which match this criteria. We present all learning curves in Figure 5 in appendix §C. We additionally present the maximum success rate achieved by each method in Table 6 in appendix §C.1.

We find that using Ground-Truth Object-Information is able to get the highest success rate on all tasks. Attentive Object-DQN performs below all methods besides OCN on $7/8$ tasks. Surprisingly, Attentive Object-DQN outperforms OCN on $5/8$ tasks. OCN doesn't incorporate action-information when learning to represent object-images across time-steps. We hypothesize that this leads it to learn degenerate object-representations that cannot discriminate object-propertiess that change due to actions, something important for our tasks.

In terms of sample-efficiency, our Attentive Object-Model comes closest to Ground-Truth Object-Information on $6/8$ tasks. 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, our Attentive Object-Model significantly improves over the COBRA Object-Model. Interestingly, sample-efficiency goes above 100% on 2 tasks. We suspect that this is because the object-model provides a learning signal for inter-object attention which is not provided by oracle information.

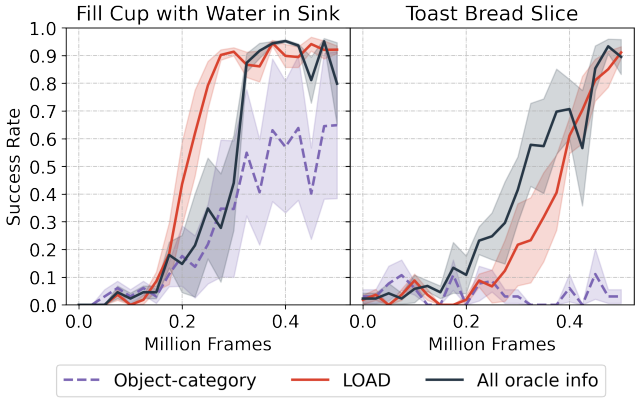
5.2 Analysis of Learned Object Representations

In Table 2, we explore our conjecture that the key to strong task-learning performance is an agent's ability to capture the information present in the oracle agent. To study this, we freeze the parameters of each encoding function, and add a linear layer to predict object-categories, object-properties, and containment relationships using a dataset of collected object-interactions we construct (see Appendix B.3 for details on the dataset and training). We find that our object-model best captures the information present in the oracle agent.

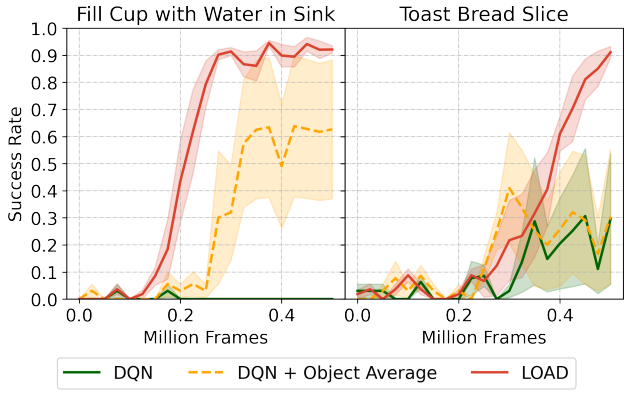
5.3 Ablations

Importance of Object-properties and Object-relations. To verify that capturing object-properties and -relations is key, we train an agent with only oracle object-category information. We find that this agent is not able to learn tasks that require using objects together as object-properties change in our sample-budget (see Figure 4(a)).

Importance of Inter-object Attention for Policy. In order to verify the utility of using attention as an inductive bias for capturing object-relations, we ablate attention from both Attentive Object-DQN and our Attentive Object-Model. First, we



(a) Ablation of object-properties and object-relations from oracle. With only oracle object-category information, the oracle can't learn these tasks in our sample budget.



(b) Ablation of inter-object attention in policy. Without this, DQN cannot learn these tasks in our sample-budget. See §5.3 for details.

Figure 4: Ablation Results.

look at two variants of Attentive Object-DQN without attention. The first is a regular DQN. In the second, we incorporate inter-object information by using the average of all present object-embeddings (DQN + Object Average). Neither learns our tasks in the sample-budget (see Figure 4).

Importance of Inter-object Attention for Model. Additionally, we look at performance where our policy can use inter-object attention but remove inter-object attention from our object-model. Without attention, we still get relatively good performance with 70% success rate; however, attention in the object-model helps increase this to 90%+ (see Figure 6 in our appendix for details).

6 Conclusion

We have shown that learning an attentive object-model 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 learning baselines, we have shown that our object-model best captures object-categories, object-properties, and containmennt-relationships. We believe that LOAD is a promising steps towards agents that can efficiently learn complex object-interaction tasks.

Acknowledgements

This work was supported in part by a grant from DARPA's L2M program and by a NSF CAREER IIS 1453651 Grant. Any opinions, findings, conclusions, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors. Additionally, WC was supported by an NSF and an RMF Fellowship.

References

[Gordon *et al.*, 2018] 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.

[Hadsell *et al.*, 2006] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[Higgins *et al.*, 2017] 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.

[Jain *et al.*, 2019] 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.

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

[Kolve *et al.*, 2017] 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.

[Mnih *et al.*, 2015] 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.

[Oh *et al.*, 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.

[Oh *et al.*, 2017] 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.

[Oh *et al.*, 2018] Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. *arXiv preprint arXiv:1806.05635*, 2018.

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

[Reed *et al.*, 2014] 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.

[Shridhar *et al.*, 2019] 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.

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

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

[Van Hoof *et al.*, 2015] 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.

[Vaswani *et al.*, 2017] 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.

[Veerapaneni *et al.*, 2020] 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.

[Watters *et al.*, 2019] 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.

[Xu *et al.*, 2019] Danfei Xu, Roberto Martín-Martín, De-An Huang, Yuke Zhu, Silvio Savarese, and Li F Fei-Fei. Regression planning networks. In *Advances in Neural Information Processing Systems*, 2019.

[Xu *et al.*, 2020] 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.

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

[Zambaldi *et al.*, 2018] 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.

[Zaragoza *et al.*, 2010] 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.

[Zhu *et al.*, 2017] 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.