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

Wilka Carvalho\textsuperscript{1*}, Anthony Liang\textsuperscript{1}, Kimin Lee\textsuperscript{2}, Sungryull Sohn\textsuperscript{1}, Honglak Lee\textsuperscript{1,3}, Richard Lewis\textsuperscript{1} and Satinder Singh\textsuperscript{1}
\textsuperscript{1}University of Michigan
\textsuperscript{2}UC Berkeley
\textsuperscript{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 to identify object categories, properties, or relationships. 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 everyday 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 \cite{zhu2017learning, shridhar2019object}. Unfortunately, \cite{zhu2017learning} 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 \cite{jain2019object, gordon2018learning}.

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 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
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 speeding up 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-properties 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-person vision without expert demonstrations, shaped rewards, or ground-truth object-knowledge. (2) A novel Attentive Object-Model auxiliary task, which frames learning an object-model as a classification problem. With our analysis, we provide evidence that for our 3D, high-fidelity 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 intuitive 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 Hooft 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 auxiliary task can help drive learning of attention.

Learning an Object-model as an Auxiliary Task. Most prior work 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 differentiate 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 $x^{egi}$ 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, $x_{loc} = (x, y, z, \varphi_1, \varphi_2, \varphi_3)$.

Actions. In this work, we focus on the Thor environment. Here, the agent has 8 base object-interactions: $I = \{\text{Pickup}, \text{Put}, \text{Open}, \text{Close}, \text{Turn on}, \text{Turn off}, \text{Slice}, \text{Fill}\}$. The agent interacts with objects by selecting (object-image-patch, interaction) pairs $a = (b, x^{oc}_i) \in I \times X^{o}$, where $x^{oc}_i$ 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
Table 1: Description of challenges associated with the tasks we study. See Figure 1 for example panels of 2 tasks.

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

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: $A_N = \{\text{Move ahead}, \text{Move back}, \text{Move right}, \text{Move left}, \text{Look up}, \text{Look down}, \text{Rotate right}, \text{Rotate left}\}$. With $\{\text{Look up}, \text{Look down}\}$, the agent can rotate its head up or down in increments of $30^\circ$ between angles $\{0^\circ, \pm 30^\circ, \pm 60^\circ\}$. $0^\circ$ represents looking straight ahead. With $\{\text{Rotate Left}, \text{Rotate Right}\}$, the agent can rotate its body by $\{\pm 90^\circ\}$.

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

To address the representation learning challenge induced by a sparse-reward signal, object-representations $z^{o,i}$ 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 $\hat{Q}(s, a)$ to estimate the action-value function $Q^\pi(s, a) = \mathbb{E}_r \sum_{t=0}^{\infty} \gamma^t r_t | S_t = s, A_t = a |$, which maps state-action pairs to the expected return on starting from that state-action pair and following policy $\pi$ 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
al., 2017] to incorporate inter-object attention into Q-value estimation. More formally, given an object-encoding \( z^{o_i} \), we can use attention to select relevant objects \( A(z^{o_i}, Z^o) \in \mathbb{R}^{d_z} \) for estimating \( Q(s, a) = (b, x^{o_i}) \). With a matrix of object-encodings, \( Z^o = [z^{o_i}]_{i=1}^{n} \in \mathbb{R}^{n \times d_z} \), we can perform this computation efficiently for each object-image-patch via:

\[
\begin{align*}
A(z^{o_i}, Z^o) &= \text{Softmax} \left( \frac{(Z^o W_q) (Z^o W_k)^T}{\sqrt{d_k}} \right) Z^o. \\
&= \begin{bmatrix}
A(z^{o_1}, Z^o) \\
\vdots \\
A(z^{o_n}, Z^o)
\end{bmatrix}
\end{align*}
\]

(1)

Here, \( Z^o W_q \) projects each object-encoding to a “query” space and \( Z^o W_k \) 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 \( Z^o \).

**Estimating action-values.** We can incorporate attention to estimate Q-values for selecting an interaction \( b \in I \) on an object \( x^{o_i} \) as follows:

\[
\hat{Q}(s, a = (b, x^{o_i})) = f_{\text{int}}([z^{o_i}, A(z^{o_i}, Z^o), z^a])
\]

(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 \( Z^o W_q \) with \( [W_q, z^a]^T \) in equation 1. We estimate Q-values for navigation actions \( b \in A_N \) as follows:

\[
\hat{Q}(s, a = b) = f_{\text{nav}}([z^a, A(z^a, Z^o)])
\]

(3)

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

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

(4)

where \( y_t = r_t + \gamma \hat{Q}(s_{t+1}, a_{t+1}; \theta_{\text{old}}) \) is the target Q-value, and \( \theta_{\text{old}} \) is an older copy of the parameters \( \theta \). 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: \( a_{t+1} = \arg \max_a \hat{Q}(s_{t+1}, a; \theta) \).

4.2 Attentive Object-Dynamics Model

Consider the global set of objects \( \{o_{i,j}\}_{i=1}^m \), 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 \( o_{i,j} \), \( p(o_{i,j}) \) (or \( p_{i,j} \) for short) and encodes it as \( z_{i,j}^{o_i} \). Given, an object-image-patch encoding \( z_{i,j}^{o_i} \) and a performed interaction \( a_t \), we can define an object-dynamics model \( D(Z^o_{i,j}, z_{i,j}^{o_i}, a_t) \) which produces the resultant encoding for \( p_{i,j}^{o_i} \). We want \( D(Z^o_{i,j}, z_{i,j}^{o_i}, a_t) \) to be closer to \( z_{i,j}^{o_i} \) 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 \( z_{i,j}^{o_i} \), we define the prediction as the output of our object-dynamics model \( D(Z^o_{i,j}, z_{i,j}^{o_i}, a_t) \). 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 \( z_{i,j}^{o_i} = \text{arg max}_{z_{i,k}^{o_i}} \cos(z_{i,j}^{o_i}, z_{i,k}^{o_i}) \). We can then select \( K \) random object-encodings \( \{z_{i,k}^{o_i}\}_{k=1}^K \) as incorrect labels. Rewriting \( D(Z^o_{i,j}, z_{i,j}^{o_i}, a_t) \) as \( D \), this leads to:

\[
p(z_{i,j}^{o_i} | Z^o_{i,j}, a_t) = \frac{\exp(D^T z_{i,j}^{o_i})}{\exp(D^T z_{i,j}^{o_i}) + \sum_k \exp(D^T z_{k,j}^{o_i})}.
\]

(5)

The set of indices corresponding to visible objects at time \( t \) is \( v_t = \{ i : p_{i,j}^{o_i} \text{ is visible at time } t \} \). The set of observed object-image-patch encodings is then \( Z_t^o = \{z_{i,j}^{o_i}\}_{j \in v_t} \).
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:

\[
L_{\text{model}} = \mathbb{E}_{z_t, a_t, z_{t+1}} \left[ -\log p(Z_{t+1}^o | Z_t^o, a_t) \right] = \mathbb{E}_{z_t, a_t, z_{t+1}} \left[ -\sum_{i \in t+1} \log p(z_{t+1}^{o,i} | Z_t^o, a_t) \right].
\]

(6)

Our final objective becomes:

\[
L = L_{\text{DQL}} + \beta_{\text{model}} L_{\text{model}}.
\]

(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 \( A(z_t^{b,i}, Z_t^o) \) into our object-model as follows:

\[
D(Z_t^o, z_t^{b,i}, a_t) = f_{\text{model}}([z_t^{b,i}, A(z_t^{b,i}, Z_t^o), z_t^a]).
\]

(8)

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

\[
z_t^a = W^o z_t^{b,c} \odot W^b b_t,
\]

where \( W^o \in \mathbb{R}^{d_o \times d_o} \), \( W^b \in \mathbb{R}^{d_o \times |A_t|} \), and \( \odot \) is an element-wise hadamard product. In practice, \( f_{\text{model}} \) 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 auxiliary tasks for learning object-representations enable sample complexity comparable to an agent with oracle object-knowledge. We additionally study the degree to which each auxiliary 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,
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.

- **OCN**: We hypothesize that this leads it to learn degenerate object-representations that cannot discriminate object-properties 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...
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 containment-relationships. We believe that LOAD is a promising step 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


