Investigating Simple Object Representations in Model-Free Deep Reinforcement Learning

Guy Davidson (guy.davidson@nyu.edu)
Center for Data Science
New York University

Brenden M. Lake (brenden@nyu.edu)
Department of Psychology and Center for Data Science
New York University

Abstract

We explore the benefits of augmenting state-of-the-art model-free deep reinforcement learning with simple object representations. Following the Frostbite challenge posited by Lake et al. (2017), we identify object representations as a critical cognitive capacity lacking from current reinforcement learning agents. We discover that providing the Rainbow model (Hessel et al., 2018) with simple, feature-engineered object representations substantially boosts its performance on the Frostbite game from Atari 2600. We then analyze the relative contributions of the representations of different types of objects, identify environment states where these representations are most impactful, and examine how these representations aid in generalizing to novel situations.

Keywords: deep reinforcement learning; object representations; model-free reinforcement learning; DQN.

Deep reinforcement learning (deep RL) offers a successful example of the interplay between cognitive science and machine learning research. Combining neural network-based function approximation with psychologically-driven reinforcement learning, research in deep RL has achieved superhuman performance on Atari games (Mnih et al., 2015), board games such as chess and Go (Silver et al., 2017), and modern video games such as DotA 2 (OpenAI, 2018). While the successes of deep RL are impressive, its current limitations are becoming increasingly more apparent. The algorithms require orders of magnitude more training to learn how to play such games compared to humans. Tsividis et al. (2017) show that fifteen minutes allow humans to rival models trained for over 100 hours of human experience, and to an extreme, the OpenAI Five models (OpenAI, 2018) collect around 900 years of human experience *every day* they are trained. Beyond requiring unrealistic quantities of experience, the solutions these algorithms learn are highly sensitive to minor design choices and random seeds (Henderson et al., 2018) and often struggle to generalize beyond their training environment (Cobbe et al., 2019; Packer et al., 2018). We believe this suggests that people and deep RL algorithms are learning different kinds of knowledge, and using different kinds of learning algorithms, in order to master new tasks like playing Atari games. Moreover, the efficiency and flexibility of human learning suggest that cognitive science has much more to contribute to the development of new RL approaches.

In this work, we focus on the representation of objects as a critical cognitive ingredient toward improving the performance of deep reinforcement learning agents. Within a few months of birth, infants demonstrate sophisticated expectations about the behavior of objects, including that objects persist, maintain size and shape, move along smooth paths, and do not pass through one another (Spelke, 1990; Spelke et al., 1992). In artificial intelligence research, Lake et al. (2017) point to object representations (as a component of intuitive physics) as an opportunity to bridge the gap between human and machine reasoning. Diuk et al. (2008) utilize this notion to reformulate the Markov Decision Process (MDP; see below) in terms of objects and interactions, and Kansky et al. (2017) offer Schema Networks as a method of reasoning over and planning with such object entities. Dubey et al. (2018) explicitly examine the importance of the visual object prior to human and artificial agents, discovering that the human agents exhibit strong reliance on the objectness of the environment, while deep RL agents suffer no penalty when it is removed. More recently, this inductive bias served as a source of inspiration for several recent advancements in deep learning, such as Kulkarni et al. (2016), van Steenkiste et al. (2018), Kulkarni et al. (2019), Veerapaneni et al. (2019), and Lin et al. (2020). This work aims to contribute toward the Frostbite Challenge posited by Lake et al. (2017), of building algorithms that learn to play Frostbite with human-like flexibility and limited training experience. We aim to pinpoint the contribution of object representations to the performance of deep RL agents, in order to reinforce the argument for object representations (and by extension, other cognitive constructs) to such models.

To that end, we begin from Rainbow (Hessel et al., 2018), a state of the art model-free deep RL algorithm, and augment its input (screen frames) with additional channels, each a mask marking the locations of unique semantic object types. We confine our current investigation to the game of Frostbite, which allows crafting such masks from the pixel colors and locations in the image. While this potentially limits the scope of our results, it allows us to perform intricate analyses of the effects of introducing these object representations. We do not consider the representations we introduce to be cognitively plausible models of the representations humans reason with; they are certainly overly simple. Instead, our choice of representations offers a minimally invasive way to examine the efficacy of such representations within the context of model-free deep reinforcement learning. We find that introducing such representations aids the model in learning from a very early stage in training, surpassing in under 10M training steps the performance Hessel et al. (2018) report in 200M training steps. We then report several analyses as to *how* the models learn to utilize these representations: we examine how the models perform without access to particular channels, investigate variations in the value functions learned, and task the models with generalizing to novel situations. We find that adding such object representations tends to aid the models in identifying environment states and generalizing to a variety of novel situations, and we discuss several conditions in which

these models are more (and less) successful in generalization.

Methodology

Reinforcement Learning. Reinforcement learning provides a computational framework to model the interactions between an agent and an environment. At each timestep t, the agent receives an observation $S_t \in \mathcal{S}$ from the environment and takes an action $A_t \in \mathcal{A}$, where $\mathcal{S}$ and $\mathcal{A}$ are the spaces of possible states and actions respectively. The agent then receives a scalar reward $R_{t+1} \in \mathbb{R}$ and the next observation S_{t+1}, and continues to act until the completion of the episode. Formally, we consider a *Markov Decision Process* defined by a tuple $\langle \mathcal{S}, \mathcal{A}, T, R, \gamma \rangle$, where T is the environment transition function, $T(s,a,s') = \mathrm{P}(S_{t+1} = s' | S_t = s, A_t = a)$, R is the reward function, and $\gamma \in [0,1]$ is a discount factor. The agent seeks to learn a policy $\pi(s,a) = \mathrm{P}(A_t = a | S_t = s)$ that maximizes the expected return $\mathrm{E}[G_t]$, where the return $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$ is the sum of future discounted rewards. For further details, see Sutton and Barto (2018).

To construct a policy, reinforcement learning agents usually estimate one of two quantities. One option is to estimate the state-value function: $v_\pi(s) = \mathrm{E}_\pi[G_t | S_t = s]$. The state value reflects the expected discounted return under the policy π starting at a state s. Another oft-estimated quantity is the action-value (or Q-value) function: $q_\pi(s,a) = \mathrm{E}_\pi[G_t | S_t = s, A_t = a]$, which models the expected discounted return from taking action a at state s (under policy π). Note that the Q-values relate to the state values by $v_\pi(s) = \max_a q_\pi(s,a)$; the state value is equal to the action value of taking the optimal action.

Atari Learning Environment and Frostbite. We utilize the Atari Learning Environment (Bellemare et al., 2013; Machado et al., 2018), and specifically the game Frostbite, as a testbed for our object representations. The ALE offers an interface to evaluate RL algorithms on Atari games and has served as an influential benchmark for reinforcement learning methods. In Frostbite[1] , the player controls an agent attempting to construct an igloo (top-right corner of pixels panel in Figure 1) before they freeze to death when a time limit is hit. To construct the igloo, the agent must jump between ice floes, making progress for each unvisited (in the current round) set of ice floes, colored white (compared to the blue of visited floes). The ice floes move from one end of the screen to the other, each row in a different direction, in increasing speed as the player makes progress in the game. As the agent navigates building the igloo, they must avoid malicious animals (which cause loss of life, in orange and yellow in Figure 1), and may receive bonus points for eating fish (in green). To succeed, the player (or model) must plan how to accomplish subgoals (visiting a row of ice floes; avoiding an animal; eating a fish) while keeping track of the underlying goal of completing the igloo and finishing the current level.

DQN. As the Rainbow algorithm builds on DQN, we provide a brief introduction to the DQN algorithm, and refer the reader to Mnih et al. (2015) for additional details: DQN is a model-free deep reinforcement learning algorithm: *model-free* as it uses a parametric function to approximate Q-values, but does not construct an explicit model of the environment; *deep* as it uses a deep neural network (see Goodfellow et al. (2016) for reference) to perform the approximation. The model receives as state observations the last four screen frames (in lower resolution and grayscale), and emits the estimated Q-value for each action, using a neural network comprised of two convolutional layers followed by two fully connected layers. DQN collects experience by acting according to an ε-greedy policy, using Q-values produced by current network weights (denoted $Q(s,a;\theta)$ for a state s, action a, and weights θ). To update these weights, DQN minimizes the difference between the current Q-values predicted to bootstrapped estimates of the Q-value given the reward received and next state observed.

Rainbow. Rainbow (Hessel et al., 2018) is an amalgamation of recent advancements in model-free deep reinforcement learning, all improving on the DQN algorithm. Rainbow combines the basic structure of DQN with several improvements, such as Prioritized Experience Replay (Schaul et al., 2016), offering a better utilization of experience replay; Dueling Networks (Wang et al., 2016), improving the Q-value approximation; and Distributional RL (Bellemare et al., 2017), representing uncertainty by approximating distributions of Q-values, rather than directly estimating expected values. For the full list and additional details, see Hessel et al. (2018). Note that save for a minor modification of the first convolutional layer, to allow for additional channels in the state observation, we make no changes to the network structure, nor do we perform any hyperparameter optimization.

Object Segmentation Masks. We compute our object segmentation masks from the full-resolution and color frames. Each semantic category (see Figure 1 for categories and example object masks) of objects in Frostbite is uniquely determined by its colors and a subset of the screen it may appear in. Each mask is computed by comparing each pixel to the appropriate color(s), and black or white-listing according to locations. As the masks are created in full-resolution, we then reduce them to the same lower resolution used for the frame pixels. We pass these masks to the model as separate input channels, just as Rainbow receives the four screen frame pixels.

Experimental Conditions. Pixels: In this baseline condition, we utilize Rainbow precisely as in Hessel et al. (2018), save for a minor change in evaluation described below. **Pixels+Objects:** In this condition, the model receives both the four most recent screen frames, as well as the eight object masks for each frame. This requires changing the dimensionality of the first convolutional layer, slightly increasing the number of weights learned, but otherwise the model remains unchanged. **Objects:** We wanted to understand how much information the screen pixels contain beyond the object masks we compute. In this condition, we omit passing the screen frames, instead passing only the masks computed from each

[1]To play the game for yourself: `http://www.virtualatari.org/soft.php?soft=Frostbite`

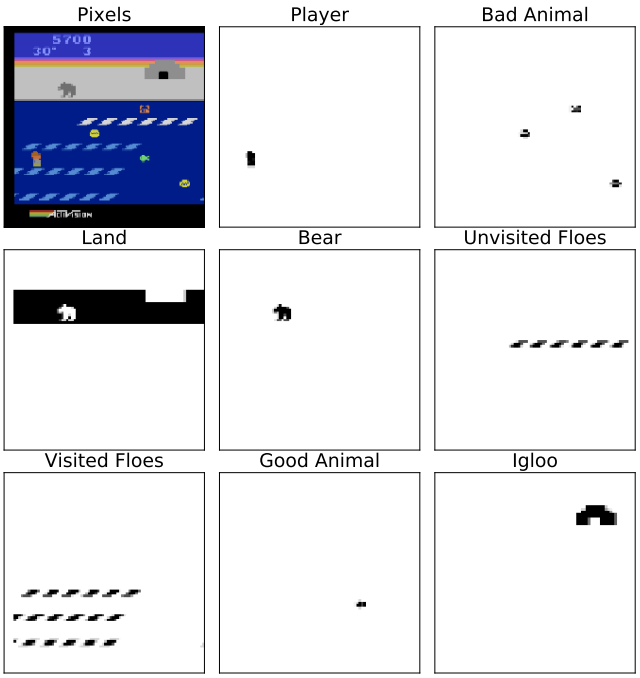

Figure 1: Object mask examples. A sample screen frame encountered during Frostbite gameplay (top left), and the object masks computed from it (remaining panels).

frame. **Grouped-Moving-Objects:** We wished to investigate the importance of separating objects to separate semantic channels. Loosely inspired by evidence that animacy influences human object organization (Konkle and Caramazza, 2013), we combined all object masks for moving objects (all except for the land and the igloo) to a single binary mask, which we passed the model alongside the state pixels. For the remainder of this paper, we shall refer to this condition as *Grouped*.

Training and evaluation. Each model is trained for ten million steps, or approximately 185 human hours[2] identically to the models reported in Hessel et al. (2018). Our evaluation protocol is similar to *no-op starts* protocol used by Hessel et al. (2018), except that rather than evaluating models for 500K frames (and truncate episodes at 108K), we allow each model ten evaluation attempts without truncation (as Frostbite as a relatively challenging game, models rarely exceed 20K frames in a single episode. We perform ten replications of the baseline *Pixels* condition, recovering similar results to those reported by Hessel et al. (2018), and ten replications of each of our experimental conditions.

Implementation. All models were implemented in PyTorch (Paszke et al., 2017). Our modifications of Rainbow built on Github user @Kaixhin's implementation of the Rainbow algorithm[3].

[2]At 60 frames per second, one human hour is equivalent to 216,000 frames, or 54,000 steps, as each step includes four new frames. See Mnih et al. (2015) for additional details.

[3]https://github.com/Kaixhin/Rainbow

Training	Pixels	Pixels+Objects	Objects	Grouped
1 days	2443 ± 661	2352 ± 617	2428 ± 612	4426 ± 585
2 days	4636 ± 792	6220 ± 618	5366 ± 863	7318 ± 560
3 days	6460 ± 1132	8915 ± 730	7612 ± 904	8417 ± 627
4 days	7102 ± 1027	10066 ± 876	9761 ± 1028	9688 ± 614
5 days	8011 ± 1112	11239 ± 1301	11677 ± 1559	10111 ± 843
6 days	8236 ± 1532	13403 ± 1157	13033 ± 1613	11570 ± 927
7 days	9748 ± 1652	14277 ± 1282	14698 ± 2038	13169 ± 1209

Table 1: Mean evaluation results. We report mean evaluation results for each condition after each day of human experience (approximately 1.3M training steps[2]). Errors reflect the standard error of the mean.

Results

Figure 2 summarizes the learning curves, depicting the mean evaluation reward received in each experimental condition over the first 10M frames (approximately 185 human hours) of training (see results for each human day of training in Table 1). Both conditions receiving the object masks show markedly better performance, breaking past the barrier of approximately 10,000 points reported by state-of-the-art model-free deep RL methods (including Hessel et al. (2018)), and continuing to improve. We also validated that this improvement cannot be explained solely by the number of parameters.[4] We find that including the semantic information afforded by channel separation offers a further benefit to performance beyond grouping all masks together, as evidenced by the *Grouped* condition models falling between the baseline *Pixels* ones and the ones with object information. We find these results promising: introducing semantically meaningful object representations enable better performance in a fraction of the 200M frames that Hessel et al. (2018) train their models for. However, the results for the first 24 human hours (see **??** in the Appendix) indicate that initial learning performance is not hastened, save for some advantage provided by the *Grouped* models over the remainder. One interpretation of this effect is that the single channel with moving objects best mirrors early developmental capacities for segmenting objects based on motion (Spelke, 1990). We take this as evidence that object representations alone will not bridge the gap to human-like learning.

We are less concerned with the improvement per se and more with what these object representations enable the model to learn, and how its learning differs from the baseline model. To that end, we dedicate the remainder of the results section to discussing several analyses that attempt to uncover how the models learn to utilize the object masks and what benefits they confer.

Omitted Objects Analysis

Our first analysis investigates the value of each type of object mask. To do so, we took models trained with access to all

[4]We evaluated a variant of the *Pixels* model with additional convolutional filters, to approximately match the number of parameters the *Pixels+Objects* has. The results were effectively identical to the baseline *Pixels* ones.

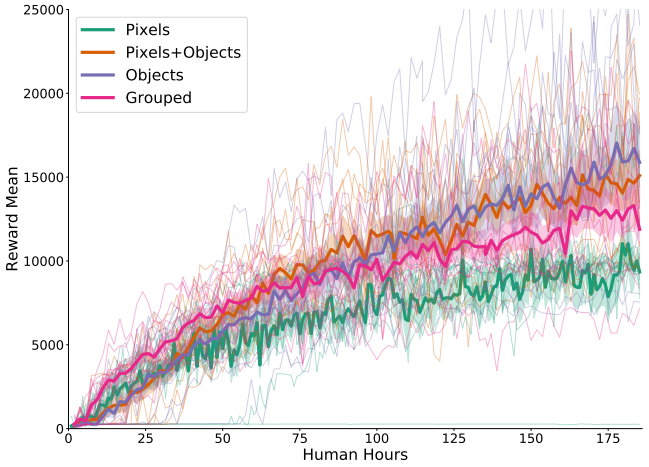

Figure 2: Mean evaluation reward per condition. Mean evaluation reward per condition. Bold lines plot the mean per condition. Shaded areas reflect the standard error of the mean. Thin lines mark individual replications.

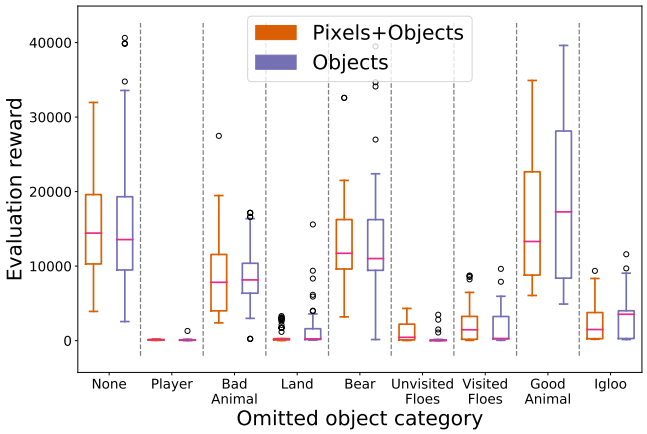

Figure 3: Omitted object comparison. Comparing the resilience of models in the *Pixels+Objects* and *Objects* conditions to omitting specific object masks. Each pair of box plots represents omitting a particular semantic category, with the full (no objects omitted) on the far left. Pink lines mark the median of each distribution, boxes the first and third quartiles, whiskers to any points with 1.5 the interquartile range, and circles to any outliers.

masks and evaluated them with one mask at a time omitted. As to not interfere with input dimensionality, we omitted each mask by zeroing it out. In each experimental condition (*Pixels+Objects* and *Objects*), we evaluated each replication ten times without each mask. The boxplots in Figure 3 mark the distributions of evaluation rewards obtained across the different replications and evaluations. With minor exceptions, we find that performance is similar in the two conditions. It appears that while the *Pixels+Objects* model receives the pixels, it primarily reasons based on the objects, and therefore sees its performance decline similarly to the *Objects* models when the object masks are no longer available.

We find that some of the object masks are much more crucial than others: without the representations identifying the agent, the land, the ice floes, and the igloo (which is used to complete a level), our models' performance drops tremendously. The two representations marking antagonist animals ('bad animals' and bear) also contribute to success, albeit on a far smaller magnitude. Omitting the mask for the bear induces a smaller penalty as the bear is introduced in a later game state than other malicious animals. Finally, omitting the mask for the good (edible for bonus points) animals increases the variability in evaluation results, and, and in the *Objects* condition, results in a higher median score. We interpret this as a conflict between short-term reward (eating an animal) and long-term success (finishing a level and moving on to the next state). Omitting information about the edible animals eliminates this tradeoff, and allows the model to 'focus' on finishing levels to accrue reward.

These results indicate that models with both pixels and object representations lean on the object representations, and do not develop any redundancy with the pixels. This analysis also validates the notion that the most important objects are the ones that guide the model to success, and that some objects may be unhelpful even when directly predictive of reward.

State-value difference comparison

Next we highlight game states such that the object-augmented networks predict substantially different state values than networks based on pixels, adapting the t-SNE (van der Maaten and Hinton, 2008) methodology (using an implementation by Ulyanov (2016)) utilized by Mnih et al. (2015) to visualize state representations. We report a single comparison, between the replications in the baseline *Pixels* condition and the replications in the *Pixels+Objects* condition. We collected sample evaluation states from the most successful (by evaluation scores) random seed of our *Pixels* condition models. Utilizing the identity $v_\pi(s) = \max_a q_\pi(s,a)$, we passed each state through each of the models compared and took the maximal action value as the state value. We then colored the embeddings plotted in Figure 4 using the (normalized) difference in means, and examined clusters of states with extreme differences to understand differences in what the models succeeded in learning. We plot the screen frames corresponding to these telling states on the circumference of Figure 4.

States (2) and (3) highlight one failure mode of the *Pixels* model: failure to recognize impending successful level completion. The baseline *Pixels* models miss the completed igloo (which terminates a level) on the dark background, and therefore fail to identify these states as highly rewarding – even though the igloo itself remains identical to previous levels–the sole change, which *Pixels* models fail to generalize to, is the change in background color. Note that the *Pixels* models do encounter the igloo on a dark background during training, but fail to recognize it–which appears to contribute to the score difference between these models and the ones that receive objects. States (1) and (6) pinpoint another source of confusion: at some stage of the game, the ice floes begin splitting and moving relative to each other, rather than only relative to the overall environment. Unlike the previous case, the *Pixels*

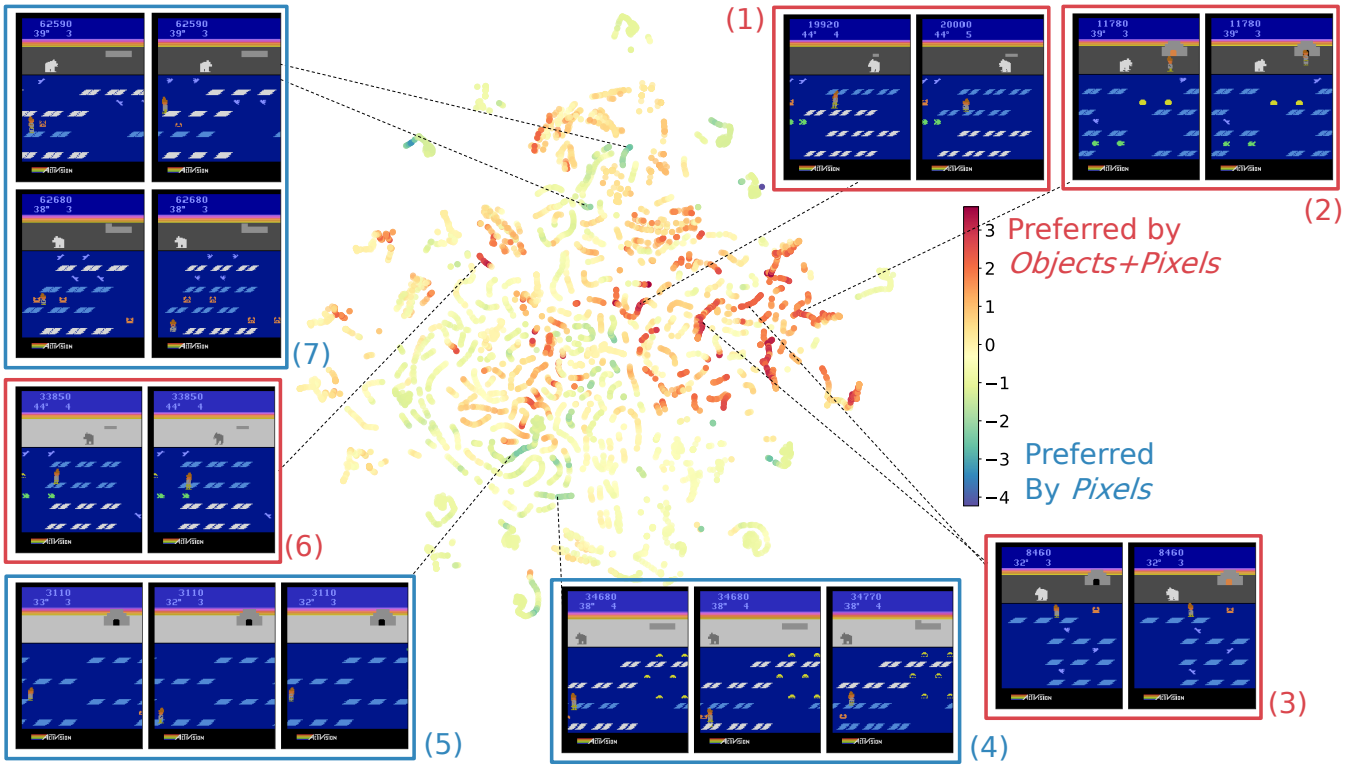

Figure 4: t-SNE visualization of internal model representations. We use a single *Pixels* condition model to compute latent representations of states collected during an evaluation, and embed them in 2D using t-SNE. We color the embeddings by the normalized difference in mean state-values [6] between models in the *Pixels+Objects* condition and models in the *Pixels* condition. Positive differences (in red) indicate higher state values in *Pixels+Objects* models, negative differences (in blue) are states valued higher by *Pixels* models. Sample states plotted (labeled 1-7 clockwise) represent observed clusters where object representations most affect the learned value functions. Border colors denote which condition predicted higher state values.

models do not encounter the splitting floes during training, as they appear in the level following the dark background igloos of states (2) and (3). Perhaps with training on these states the *Pixels* models would learn to act around them; but as it stands, these models fail to generalize to this condition. This motion appears to confuse the baseline *Pixels* models; we conjecture that the object representations enable the *Pixels+Objects* models to reason better about this condition.

States (4), (5), and (7) depict situations where the baseline *Pixels* models erroneously fail to notice dangerous situations. For example, in state (5), although the igloo is complete, the *Pixels+Objects* models identify that the agent appears to be stuck without a way to return to land. Indeed, the agent successfully jumps down, but then cannot return up and jumps into the water (and to its death). States (4) and (7) represent similarly dangerous environment states, which the model used to collect the data survives in state (4) and succumbs to in state (7)[5]. In these states, the *Pixels+Objects* models appear to identify the precarious situation the agent is in, while the *Pixels* models without object representations do not.

Generalization to novel situations

We now investigate the utility of our object representations in generalizing to situations radically different from anything encountered during training, in hopes of investigating the capacity for systematic, human-like generalization. In Figure 5 and Figure 6, we augmented a state with additional features, adjusting both the pixels and object masks accordingly. We report a boxplot of state values predicted by models in each condition, both for the original state before any manipulation and for each manipulated state.

In Figure 5, we attempt to overwhelm the agent by surrounding it with animals–during normal gameplay, the model does not see anything resembling this simultaneous quantity of animals. We also tested how our models would react to a novel object (an alien from Space Invaders), marked either as a 'good animal' or as a 'bad animal.' The *Pixels* models respond negatively to all of the overwhelming conditions, which is incorrect behavior in the Fish scene and correct behavior otherwise. Both *Pixels+Objects* models and *Objects* models show much stronger robustness to this change in distribution, reacting favorably to being surrounded with edible animals, and negatively to being surrounded with dangerous animals, demonstrating the generalization benefit conferred by our ob-

[5]Note that states (4) and (7), like states (1) and (6), feature the splitting ice floes; the baseline *Pixels* models' failure to recognize those might contribute to the differences in states (4) and (7) as well.

[6]We pass each state through the ten replications in each condition, compute the expected state value (expected as Rainbow utilizes distributional RL), average across each condition for each state, compute the difference of means for each state, and normalize the collection of differences.

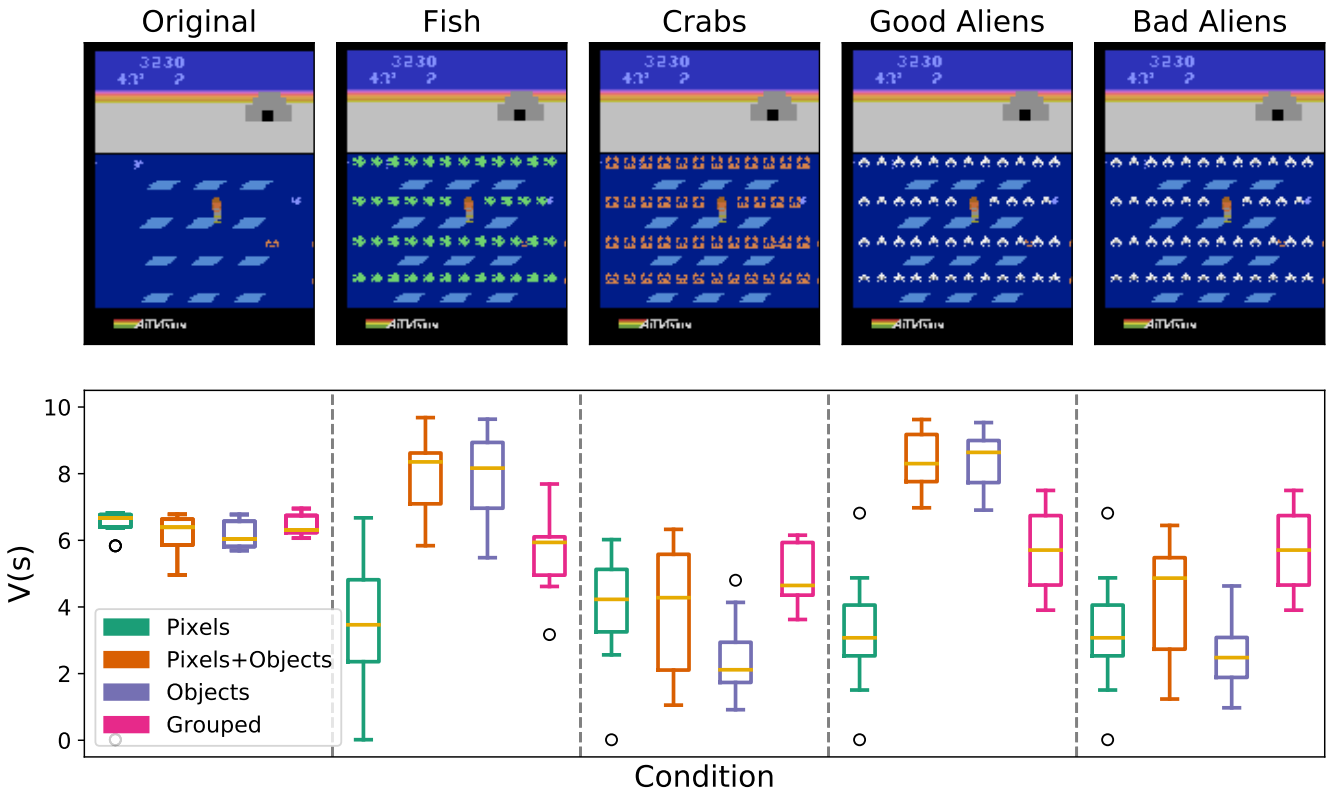

Figure 5: Generalization to being surrounded. We examined how each model would react to being surrounded. In the first two manipulations, "Fish" and "Crabs," we surround the agent with existing creatures from Frostbite. In the second two, we borrow an alien sprite from the Atari game Space Invaders and label it in different object masks: first as edible fish, then as dangerous crabs.

ject representations. Some of these differences in state-value distributions proved statistically significant: in a two-sample t-test, the differences in between the *Pixels* and *Pixels+Objects* models were significant ($P < 0.001$) in the 'Fish' and 'Good Aliens' condition. The same held true for the *Pixels* and *Objects* models. Comparisons between the *Grouped* model and *Pixels+Objects* showed the same pattern of significance, in the 'Fish' and 'Good Aliens' cases, while the difference between the *Grouped* and *Objects* models proved significant ($P < 0.002$) in all four generalization scenarios. When comparing within a model between the different manipulations, results were equally stark: for the *Pixels+Objects* and *Objects* models, *all* differences were significant, save for the two with identical valence ('Fish' and 'Good Aliens,' 'Crabs' and 'Bad Aliens'). Note that these are entirely novel objects to the agents, not merely unusual configurations of prior objects. The entirely novel alien pixels appear to have little impact on the *Pixels+Objects* models, providing further evidence that these models rely on the object masks to reason (rather than on the raw pixel observations).

We offer a second, potentially harder example, in which object representations do not seem sufficient to enable generalization. In Figure 6, we examine how the models respond to another situation far off the training data manifold: on a solitary ice floe, surrounded by crabs, with nowhere to go. Only in one scenario ('Escape Route') does the agent have a path to survival, and once arriving on land, could immediately finish the level and receive a substantial bonus. While some of the models show a preference to that state over the two guaranteed death states, the differences are fairly minor, and only one (for the *Pixels+Objects* model, between 'Death #2' and 'Escape Route') proves statistically significant. This scenario, both its original state and the various manipulations performed, fall far off the data manifold, which may account for the higher variance in state values assigned to original state (compared to the one exhibited in Figure 5).

Discussion

We report a careful analysis of the effect of adding simple object masks to the deep reinforcement learning algorithm Rainbow evaluated on the Atari game Frostbite. We find that these representations enable the model to learn better and achieve higher overall scores in a fraction of the training time. We then dissect the models to find the effect of these object representations. We analyze their relative contributions to the model's success, we find states where object representations drastically change the learned value function, and we examine how these models would respond to novel scenarios. While we believe this provides strong preliminary evidence to the value of object representations for reinforcement learning, we acknowledge there is plenty of work ahead.

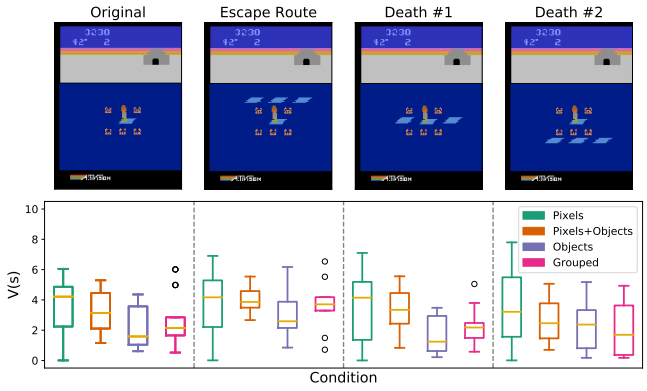

Figure 6: Generalization to being stranded among crabs. We examined how each model would react when stranded and surrounded with crabs, with introducing additional ice floes that either serve as an escape route or a distraction.

We aim to investigate utilizing similar representations in other reinforcement learning tasks and explore methods towards learning robust, cognitively-plausible object representations, rather than providing them externally. Furthermore, we believe that additional core ingredients of human cognition may prove as useful (if not more so) to the development of flexible and rapidly-learning reinforcement learning algorithms. We hope to explore the benefits offered by an intuitive physical understanding of the world can confer, as it can allow an agent to predict the physical dynamics of the environment and plan accordingly. We also believe that the capacity to develop causal models explaining observations in the environment, and the ability to meta-learn such models across different scenarios and games, would provide another step forward in the effort to build genuinely human-like reinforcement learning agents.

Acknowledgements

BML is grateful for discussions with Tomer Ullman, Josh Tenenbaum, and Sam Gershman while writing "Building machines that learn and think like people." Those ideas sit at the heart of this paper. We also thank Reuben Feinman and Wai Keen Vong for helpful comments on this manuscript.

References

Bellemare, M. G., Dabney, W., and Munos, R. (2017). A Distributional Perspective on Reinforcement Learning. In *Proceedings of the 34 th International Conference on Machine Learning, Sydney, Australia, PMLR 70*. 2

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279. 2

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. (2019). Quantifying Generalization in Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, PMLR 97*. 1

Diuk, C., Cohen, A., and Littman, M. L. (2008). An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 240–247, New York, NY, USA. Association for Computing Machinery. 1

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T. L., and Efros, A. A. (2018). Investigating human priors for playing video games. In *ICML*. 1

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. 2

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. (2018). Deep Reinforcement Learning that Matters. In *Thirthy-Second AAAI Conference On Artificial Intelligence (AAAI)*. 1

Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. (2018). Rainbow: Combining Improvements in Deep Reinforcement Learning. In *AAAI'18*. 1, 2, 3

Kansky, K., Silver, T., Mély, D. A., Eldawy, M., Lázaro-Gredilla, M., Lou, X., Dorfman, N., Sidor, S., Phoenix, S., and George, D. (2017). Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1809–1818. JMLR.org. 1

Konkle, T. and Caramazza, A. (2013). Tripartite organization of the ventral stream by animacy and object size. *Journal of Neuroscience*, 33(25):10235–10242. 3

Kulkarni, T. D., Gupta, A., Ionescu, C., Borgeaud, S., Reynolds, M., Zisserman, A., and Mnih, V. (2019). Unsupervised Learning of Object Keypoints for Perception and Control. In *Advances in Neural Information Processing Systems*, pages 10723–10733. 1

Kulkarni, T. D., Narasimhan, K. R., Saeedi, A., and Bcs, J. B. T. (2016). Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. In *Advances in Neural Information Processing Systems 29*, pages 3675–3683. 1

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40(e253). 1

Lin, Z., Wu, Y.-F., Vishwanath Peri, S., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. (2020). SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition. In *International Conference on Learning Representations*. 1

Machado, M. C., Bellemare, M. G., Talvitie, E., Veness, J., Hausknecht, M. J., and Bowling, M. (2018). Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562. 2

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533. 1, 2, 3, 4

OpenAI (2018). OpenAI Five. 1

Packer, C., Gao, K., Kos, J., Krähenbühl, P., Koltun, V., and Song, D. (2018). Assessing Generalization in Deep Reinforcement Learning. 1

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS*. 3

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016). Prioritized Experience Replay. In *International Conference on Learning Representations*. 2

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359. 1

Spelke, E. S. (1990). Principles of object perception. *Cognitive Science*, 14(1):29–56. 1, 3

Spelke, E. S., Breinlinger, K., Macomber, J., and Jacobson, K. (1992). Origins of Knowledge. *Psychological Review*, 99(4):605–632. 1

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA. 2

Tsividis, P. A., Pouncy, T., Xu, J. L., Tenenbaum, J. B., and Gershman, S. J. (2017). Human learning in Atari. In *2017 AAAI Spring Symposium Series, Science of Intelligence: Computational Principles of Natural and Artificial Intelligence*. 1

Ulyanov, D. (2016). Multicore-tsne. `https://github.com/DmitryUlyanov/Multicore-TSNE`. 4

van der Maaten, L. and Hinton, G. (2008). Visualizing high-dimensional data using t-sne. 4

van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. (2018). Relational Neural Expectation Maximization: Unsupervised Discovery of Objects and their Interactions. In *International Conference on Learning Representations*. 1

Veerapaneni, R., Co-Reyes, J. D., Chang, M., Janner, M., Finn, C., Wu, J., Tenenbaum, J. B., and Levine, S. (2019). Entity Abstraction in Visual Model-Based Reinforcement Learning. 1

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., and De Frcitas, N. (2016). Dueling Network Architectures for Deep Reinforcement Learning. In *33rd International Conference on Machine Learning, ICML 2016*, volume 4, pages 2939–2947. International Machine Learning Society (IMLS). 2