arXiv:1904.07482v1 [cs.LG] 16 Apr 2019

Object-Oriented Dynamics Learning through Multi-Level Abstraction

Guangxiang Zhu [*1] **Jianhao Wang** [*1] **Zhizhou Ren** [*1] **Chongjie Zhang** [1]

Abstract

Object-based approaches for learning action-conditioned dynamics has demonstrated promise for generalization and interpretability. However, existing approaches suffer from structural limitations and optimization difficulties for common environments with multiple dynamic objects. In this paper, we present a novel self-supervised learning framework, called *Multi-level Abstraction Object-oriented Predictor* (MAOP), which employs a three-level learning architecture that enables efficient object-based dynamics learning from raw visual observations. We also design a spatial-temporal relational reasoning mechanism for MAOP to support instance-level dynamics learning and handle partial observability. Our results show that MAOP significantly outperforms previous methods in terms of sample efficiency and generalization over novel environments for learning environment models. We also demonstrate that learned dynamics models enable efficient planning in unseen environments, comparable to true environment models. In addition, MAOP learns semantically and visually interpretable disentangled representations.

1. Introduction

Model-based deep reinforcement learning (DRL) has recently attracted much attention for improving sample efficiency of DRL (Heess et al., 2015; Schmidhuber, 2015; Gu et al., 2016; Racanière et al., 2017; Finn & Levine, 2017). One of the core problems for model-based DRL is to learn action-conditioned dynamics models through interacting with environments. Pixel-based approaches have been proposed for such dynamics learning from raw visual perception, achieving remarkable performance in training environments (Oh et al., 2015; Watter et al., 2015; Chiappa et al., 2017).

To unlock sample efficiency of model-based DRL, learning action-conditioned dynamics models that generalize over unseen environments is critical yet challenging. Finn et al. (2016) proposed a dynamics learning method that takes a step towards generalization over object appearances. Zhu et al. (2018) developed an object-oriented dynamics predictor to support efficient learning and generalization. However, due to structural limitations and optimization difficulties, these methods do not efficiently generalize over environments with multiple controllable and uncontrollable dynamic objects and different static object layouts.

To address these limitations, this paper presents a novel self-supervised, object-oriented dynamics learning framework, called *Multi-level Abstraction Object-oriented Predictor* (MAOP). This framework simultaneously learns disentangled object representations and predicts object motions conditioned on their historical states, their interactions to other objects, and an agent's actions. To reduce the complexity of such concurrent learning and improve sample efficiency, MAOP employs a three-level learning architecture from the most abstract level of motion detection, to dynamic instance segmentation, and to dynamics learning and prediction. A more abstract learning level solves an easier problem and has lower learning complexity, and its output provides a coarse-grained guidance for a less abstract learning level, improving its speed and quality of learning. This multi-level architecture is inspired by humans' multi-level motion perception from cognitive science studies (Johansson, 1975; Lu & Sperling, 1995; Smith et al., 1998) and multi-level abstraction search in constraint optimization (Zhang & Shah, 2016). In addition, we design a novel CNN-based spatial-temporal relational reasoning mechanism for MAOP, which includes a Relation Net to reason about spatial relations between objects and an Inertia Net to learn temporal effects. This mechanism offers a disentangled way to handle physical reasoning in settings with partial observability.

Our results show that MAOP significantly outperforms previous methods for learning dynamics models in terms of sample efficiency and generalization over novel settings with multiple controllable and uncontrollable dynamic objects and different object layouts. MAOP enables model

[*]Equal contribution [1]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China. Correspondence to: Guangxiang Zhu <guangxiangzhu@outlook.com>, Jianhao Wang <jh-wang15@mails.tsinghua.edu.cn>, Zhizhou Ren <rzz16@mails.tsinghua.edu.cn>, Chongjie Zhang <chongjie@tsinghua.edu.cn>.

Preprint. Work in progress.

learning from few interactions with environments and accurately predicting the dynamics of objects as well as raw visual observations in previously unseen environments. The learned dynamics model enables an agent to directly plan in unseen environments without retraining. In addition, MAOP learns disentangled representations and gains visually and semantically interpretable knowledge, including meaningful object masks, accurate object motions, disentangled relational reasoning process, and controllable factors. Last but not least, MAOP provides a general multi-level framework for learning object-based dynamics model from raw visual observations, offering opportunities to easily leverage well-studied object detection methods (e.g., Mask R-CNN (He et al., 2017)) in the area of computer vision.

2. Related Work

Object-oriented reinforcement learning has received much research attention, which exploits efficient representations based on objects and their interactions. This learning paradigm is close to that of human cognition in the physical world and the learned object-level knowledge can be efficiently generalized across environments. Early work on object-oriented RL requires explicit encodings of object representations, such as *relational MDPs* (Guestrin et al., 2003), *OO-MDPs* (Diuk et al., 2008), object focused q-learning (Cobo et al., 2013), and Schema Networks (Kansky et al., 2017). In this paper, we present an end-to-end, self-supervised neural network framework that automatically learns object representations and dynamics conditioned on actions and object relations from raw visual observations.

Action-conditioned dynamics learning aims to address one of the core problems for model-based DRL, i.e., constructing an environment dynamics model. Several pixel-based approaches have been proposed for learning how an environment changes in response to actions through unsupervised video prediction and achieve remarkable performance in training environments (Oh et al., 2015; Watter et al., 2015; Chiappa et al., 2017). Fragkiadaki et al. (2016) propose an object-centric prediction method to learn the dynamics model when given the object localization and tracking. Finn et al. (2016) propose an action-conditioned video prediction method that explicitly models pixel motion and learns invariance to object appearances. Recently, Zhu et al. (2018) propose an object-oriented dynamics learning paradigm to support efficient learning. However, it focuses on environments with a single dynamic object. In this paper, we take a further step towards efficient learning of object-oriented dynamics model in more general environments with multiple dynamic objects and also demonstrate its usage for model-based planning. In addition, we design an instance-aware dynamics mechanism to support instance-level dynamics learning and handle partial observations.

Relation-based deep learning approaches make significant progress in a wide range of domains such as physical reasoning (Chang et al., 2016; Battaglia et al., 2016; van Steenkiste et al., 2018), computer vision (Watters et al., 2017; Wu et al., 2017), natural language processing (Santoro et al., 2017), and reinforcement learning (Zambaldi et al., 2018; Zhu et al., 2018). Relation-based nets introduce relational inductive biases into neural networks, which facilitate generalization over entities and relations and enables relational reasoning (Battaglia et al., 2018). This paper proposes a novel spatial-temporal relational reasoning mechanism, which includes an Inertia Net for learning temporal effects in addition to a CNN-based Relation Net for reasoning about spatial relations.

Instance Segmentation has been one of the fundamental problems in computer vision. Instance segmentation can be regarded as the combination of semantic segmentation and object localization. Many approaches have been proposed for instance segmentation, including DeepMask (Pinheiro et al., 2015), InstanceFCN (Dai et al., 2016), FCIS (Li et al., 2017), and Mask R-CNN (He et al., 2017). Most models are supervised learning and require a large labeled training dataset. Liu et al. (2015) proposes a weakly-supervised approach to infer object instances in foreground by exploiting dynamic consistency in video. In this paper, we design a self-supervised, three-level approach for learning dynamic rigid object instances. At the most abstract level, foreground detection produces region proposals for instance segmentation. The instance segmentation level then learns coarse dynamic instance segmentation. This coarse instance segmentation provides a guidance for learning accurate instances at the dynamics learning level, whose instance segmentation considers not only object appearances but also motion prediction conditioned on object-to-object relations and actions.

3. Multi-level Abstraction Object-oriented Predictor (MAOP)

In this section, we will present a novel self-supervised deep learning framework, aiming to learn object-oriented dynamics models that are able to efficiently generalize over unseen environments with different object layouts and multiple dynamic objects. Such a generalized object-oriented dynamics learning approach requires simultaneously learning object representations and motions conditioned on their historical states, their interactions to other objects, and an agent's actions. This concurrent learning is challenging for an end-to-end approach in complex environments. Evidences from cognitive science studies (Johansson, 1975; Lu & Sperling, 1995; Smith et al., 1998) show that human beings are born with prior motion perception ability (in Cortical area MT) of perceiving moving and motionlessness, which enables

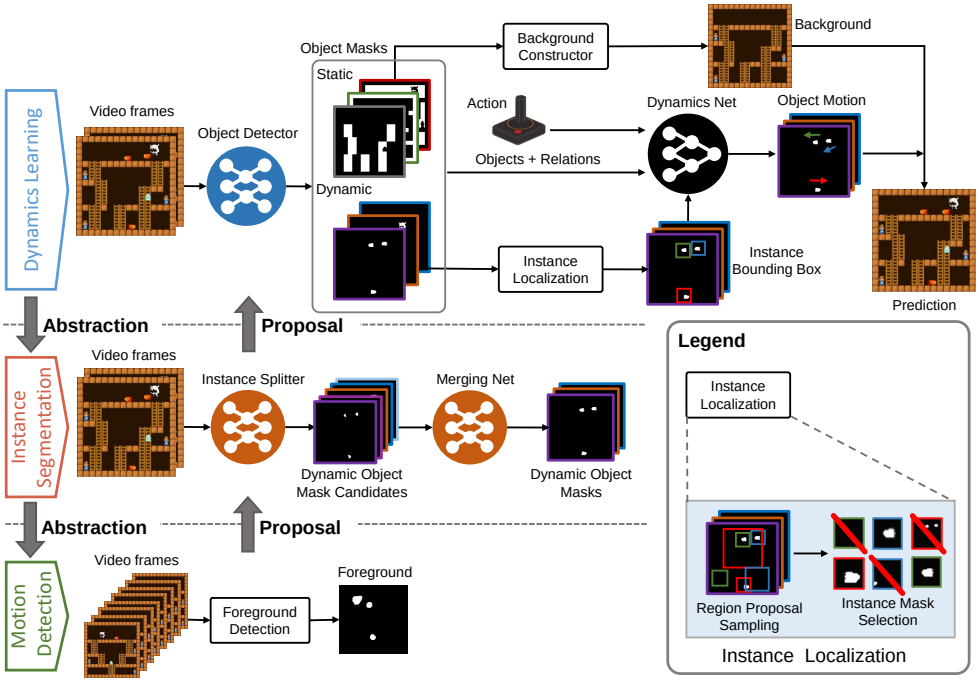


Figure 1. Multi-level dynamics learning framework. From a bottom-up view, we first perform motion detection to produce foreground masks. Then, we utilize the foreground masks as dynamic region proposals to guide the learning of dynamic instance segmentation. Finally, we use the learned dynamic instance segmentation networks (Instance Splitter and Merging Net) as a guiding network to generate region proposals of dynamic instances and guide the learning of Object Detector at the level of dynamics learning. We provide a pseudocode that sketches out this multi-level framework in Appendix Algorithm A1.

learning more complex knowledge, such as object-level dynamics prediction. Inspired by these studies, we design a multi-level learning framework, called *Multi-level Abstraction Object-oriented Predictor* (MAOP), which incorporates motion perception levels to assist in dynamics learning.

Figure 1 illustrates three levels of MAOP framework: dynamics learning, dynamic instance segmentation, and motion detection. Here we present them from a top-down decomposition view. The dynamics learning level is an end-to-end, self-supervised neural network, aiming to learn object representations and instance-level dynamics, and predict the next visual observation conditioned on object-to-object relations and an agent's action. To guide the learning of the object representations and instance localization at the level of dynamics learning, the more abstracted level of dynamic instance segmentation learns a guiding network in a self-supervised manner, which can provide coarse mask proposals of dynamic instances. This level exploits spatial-temporal information of locomotion property and appearance patterns to capture region proposals of dynamic instances. To facilitate the learning of dynamic instance segmentation, MAOP employs the more coarse-grained level of motion detection, which detects changes in image sequences and provides guidance on proposing regions potentially containing dynamic instance. As the learning proceeds, the knowledge distilled from the more coarse-grained level are gradually refined at the more fine-grained level by considering additional information. When the training is finished, the coarse-grained levels of dynamic instance segmentation and motion detection will be removed at the testing stage. In the rest of this section, we will describe in detail the design of each level and their connections.

3.1. Object-Oriented Dynamics Learning Level

The semantics of this level is formulated as learning an object-based dynamics model with region proposals generated from the more abstracted level of dynamic instance segmentation. Its architecture is shown at the top level of Figure 1, which is an end-to-end neural network and can be trained in a self-supervised manner. It takes a sequence of video frames and an agent's actions as input, learns disentangled representations (including objects, relations and effects) and dynamics of controllable and uncontrollable dynamic object instances conditioned on actions and object relations, and produces predictions of raw visual observations. The whole architecture includes four major components: A) an Object Detector that learns to decompose the input image into objects; B) an Instance Localization module responsible for localizing dynamic instances; C) a Dynamics Net for learning the motion of each dynamic instance conditioned on the effects from actions and object-level spatial-temporal relations; and D) a Background Constructor that computes a background image from learned static object masks. In addition to Figure 1, we further provide Appendix Algo-

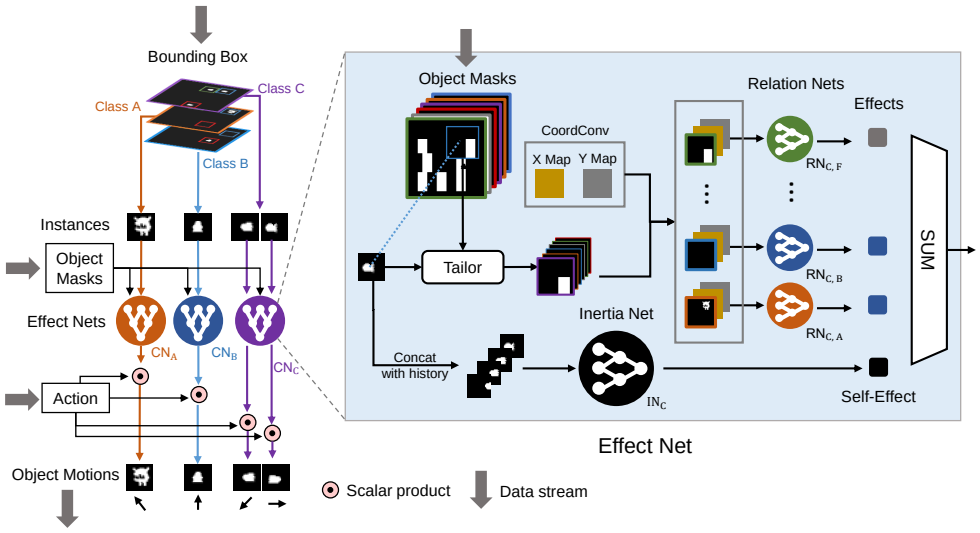


Figure 2. Architecture of Dynamics Net (left) and its component Effect Net (right). Different classes of objects are distinguished by different letters (e.g., A, B, ... , F). Dynamics Net has one Effect Net for each class of objects. An Effect Net consists of one Inertia Net and several Relation Nets.

rithm A2 to describe interactions of these components and the learning paradigm of object-based dynamics, which is a general framework and agnostic to the concrete form of each component. In the following paragraphs, we will describe detailed design of each component.

Object Detector and Instance Localization Module. Object Detector is a CNN module aiming to learn object masks from a sequence of input images. An object mask describes a spatial distribution of a class of objects, which forms the fundamental building block of our object-oriented framework. Considering that instances of the same class are likely to have different motions, we append an Instance Localization Module to Object Detector to localize each dynamic instance to support instance-level dynamics learning. Class-specific object masks in conjunction with instance localization bridge visual perception (Object Detector) with dynamics learning (Dynamics Net), which allows learning objects based on both appearances and dynamics.

Specifically, Object Detector receives image $\mathbf{I}^t \in \mathbb{R}^{H\times W\times 3}$ at timestep t and then outputs object masks $\mathbf{O}^t \in [0,1]^{H\times W\times n_O}$, including dynamic object masks $\mathbf{D}^t \in [0,1]^{H\times W\times n_D}$ and static object masks $\mathbf{S}^t \in [0,1]^{H\times W\times n_S}$, where H and W denote the height and width of the input image, n_D and n_S denotes the maximum possible number of dynamic and static object classes respectively, and $n_O = n_D + n_S$. Entry $\mathbf{O}_{u,v,i}$ indicates the probability that pixel $\mathbf{I}_{u,v,:}$ belongs to the i-th object class. The Instance Localization module uses learned dynamic object masks to identify each object instance mask $\mathbf{X}^t_{:,:,i} \in [0,1]^{H_M\times W_M} (1 \le i \le n_M)$, where H_M and W_M denote the height and width of the bounding box of this instance and n_M denotes the maximum possible number of localized instances. As shown in Figure 1, Instance Localization first samples a number of bounding boxes on dynamic object masks and then select the regions, each of which contains only one dynamic instance. As we focus on the motion of rigid objects, the affine transformation is approximatively consistent for all pixels of each dynamic instance mask. Inspired by this, we define a discrepancy loss $\mathcal{L}_{\text{instance}}$ for a sampled region that measures motion consistence of its pixels and use it as a selection score for selecting instance masks. To compute this loss, we first compute an average rigid transformation of a sampled region between two time steps, then apply this transformation to this region at the previous time step by using Spatial Transformer Network (STN) (Jaderberg et al., 2015), and finally compared this predicted region with the region at the current time (the difference is measured by l_2 distance). Obviously, when a sampled region contains exactly one dynamic instance, this loss will be extremely small, and even zero when object masks are perfectly learned. More details of region proposal sampling and instance mask selection can be found in Appendix Section 3.

Dynamics Net. Dynamics Net is designed to learn instance-level motion effects of actions, object-to-object spatial relations (Relation Net) and temporal relations of spatial states (Inertia Net), and to reason about the motion of each dynamic instance based on these effects. Its architecture is illustrated as Figure 2, where the motion of each dynamic instance is individually computed. We take as an example the computation of the motion of the i-th instance $\mathbf{X}^t_{:,:,i}$ to illustrate the detailed structure of the Effect Net.

As shown in the right subfigure of Figure 2, Effect Net first uses a sub-differentiable tailor module introduced by Zhu et al. (2018) to enable the inference of dynamics focusing on the relations with neighbour objects. This module

crops a w-size "horizon" window from the concatenated masks of all objects $\mathbf{O}^t$ centered on the expected location of $\mathbf{X}^t_{:,:,i}$, where w denotes the maximum effective range of relations. Then, the cropped object masks are concatenated with constant x-coordinate and y-coordinate meshgrid map (to make networks more sensitive to the spatial information) and fed into corresponding Relation Nets (RN) according to their classes. We use $\mathbf{C}^t_{:,:,i,j}$ to denote the cropped mask that crops the j-th object class $\mathbf{O}^t_{:,:,j}$ centered on the expected location of the i-th dynamic instance (the class it belongs to is denoted as $c_i, 1 \leq c_i \leq n_D$). The effect of object class j on class c_i, $E^t(c_i, j) \in \mathbb{R}^{2\times n_a}$ (n_a denotes the number of actions) is calculated as, $E^t(c_i, j) = \text{RN}_{c_i,j}\Big(\text{concat}\big(\mathbf{C}^t_{:,:,i,j}, \text{Xmap}, \text{Ymap}\big)\Big)$. Note that there are total $n_D \times n_O$ RNs for $n_D \times n_O$ pairs of object classes that share the same architecture but not their weights. To handle the partial observation problem, we add an Inertia Nets (IN) to learn self-effects of an object class through historical states, $E^t_{\text{self}}(c_i) = \text{IN}_{c_i}\Big(\text{concat}\big(\mathbf{X}^t_{:,:,i}, \mathbf{X}^{t+1}_{:,:,i}, \ldots, \mathbf{X}^{t+h}_{:,:,i}\big)\Big)$, where h is the history length. There are total n_D INs for n_D dynamic object classes, which share the same architecture but not their weights. To predict the motion vector $\boldsymbol{m}^t_i \in \mathbb{R}^2$ for the i-th dynamic instance, all these effects are summed up and then multiplied by the one-hot coding of action $\boldsymbol{a}^t \in \{0, 1\}^{n_a}$, that is, $\boldsymbol{m}^t_i = \Big(\big(\sum_j E^t(c_i, j)\big) + E^t_{\text{self}}(c_i)\Big) \cdot \boldsymbol{a}^t$.

Background Constructor. This module constructs the static background of an input image based on the static object masks learned by Object Detector, which is then combined with the predicted dynamic instances to predict the next visual observation. As Object Detector can decompose its observation into objects in an unseen environment with a different object layout, Background Constructor is able to generate a corresponding static background and support the visual observation prediction in novel environments. Specifically, Background Constructor maintains a background memory $\mathbf{B} \in \mathbb{R}^{H\times W\times 3}$ which is continuously updated with the static object mask learned by Object Detector. Denoting α as the decay rate, the updating formula is given by, $\mathbf{B}^t = \alpha\mathbf{B}^{t-1} + (1-\alpha)\mathbf{I}^t \sum_i \mathbf{S}^t_{:,:,i}$, and $\mathbf{B}^0 = 0$.

Prediction and Training Loss. Based on the learned masks and motions of the object instances, we propose an object-oriented prediction loss, $\mathcal{L}_{\text{pred-object}} = \sum_i \Big\| \text{STN}\Big((\bar{u}_i, \bar{v}_i)^t, \boldsymbol{m}^t_i\Big) - (\bar{u}_i, \bar{v}_i)^{t+1} \Big\|^2_2$, where $(\bar{u}_i, \bar{v}_i)^t$ is the excepted location of i-th instance mask $\mathbf{X}^t_{:,:,i}$ at timestep t. To utilize the information of ground-true future frames, we also use a conventional image prediction loss. In our model, the prediction of the next frame is produced by merging the learned object motions and the background $\mathbf{B}^t$. The pixels of a dynamic instance can be calculated by masking the raw image with the corresponding instance mask and we can use STN to apply the learned instance motion vector $\boldsymbol{m}^t_i$ on these pixels. First, we transform all the dynamic instances according to the learned instance-level motions. Then, we merge all the transformed dynamic instances with the background image calculated from Background Constructor to generate the prediction of the next frame. In this paper, we use the pixel-wise l_2 loss to restrain image prediction error, denoted as $\mathcal{L}_{\text{pred-image}}$. In addition, we add a proposal loss to utilize the dynamic instance proposals for guiding the learning, which is given by $\mathcal{L}_{\text{proposal}} = \Big\| \sum_i (\mathbf{D}^t_{:,:,i} - \mathbf{P}^t_{:,:,i}) \Big\|^2_2$, where $\mathbf{P}$ denotes the dynamic instance region proposals provided by the level of dynamic instance segmentation. Therefore, the total loss of the dynamics learning level is given by, $\mathcal{L}_{\text{DL}} = \mathcal{L}_{\text{pred-object}} + \lambda_1\mathcal{L}_{\text{pred-image}} + \lambda_2\mathcal{L}_{\text{proposal}}$.

3.2. Dynamic Instance Segmentation Level

This level aims to generate region proposals of dynamic instances to guide the learning of object masks and facilitate instance localization at the level of dynamics learning. The architecture is shown in Figure 1. Instance Splitter aims to identify regions, each of which potentially contains one dynamic instance. To learn to divide different dynamic object instances onto different masks, we use the discrepancy loss $\mathcal{L}_{\text{instance}}$ described in Section 3.1 to train Instance Splitter. Considering that one object instance may be split into smaller patches on different masks, we append a Merging Net (i.e., a two-layer CNN with 1 kernel size and 1 stride) to Instance Splitter to learn to merge masks. This module uses a merging loss $\mathcal{L}_{\text{merge}}$ that aims to merge mask candidates that are adjacent and share the same motion. In addition, we add a foreground proposal loss $\mathcal{L}_{\text{forground}}$ to encourage attentions on dynamic regions provided by the level of motion detection, which is defined similar to $\mathcal{L}_{\text{proposal}}$ at the level of dynamics learning. The total loss of this level is given by, $\mathcal{L}_{\text{DIS}} = \mathcal{L}_{\text{instance}} + \lambda_3\mathcal{L}_{\text{merge}} + \lambda_4\mathcal{L}_{\text{forground}}$.

Although the network structure of this level is similar to Object Detector in the level of dynamics learning, we do not integrated them together as a whole network because concurrent learning of both object representations and dynamics model is not stable. Instead, we first learn the coarse object representations only based on the spatial-temporal consistency of locomotion and appearance pattern, and then use them as proposal regions to guide object-oriented dynamics learning at the more fine-grained level, which in turn fine-tunes the object representations. In addition, MAOP is also readily to incorporate Mask R-CNN (He et al., 2017) or other off-the-shelf supervised object detection methods (Liu et al., 2018) as a plug-and-play module into our framework to generate region proposals of dynamic instances.

3.3. Motion Detection Level

At this level, we employ foreground detection to detect potential regions of dynamic objects from a sequence of image frames and provide coarse dynamic region proposals for assisting in dynamic instance segmentation. In our experiments, we use a simple unsupervised foreground detection approach proposed by Lo & Velastin (2001). Our framework is also compatible with many advanced unsupervised foreground detection methods (Lee, 2005; Maddalena et al., 2008; Zhou et al., 2013; Guo et al., 2014) that are more efficient or more robust to moving camera. These complex foreground detection methods have the potential to improve the performance but are not the focus of this work.

4. Experiments

We compare MAOP with state-of-the-art action-conditioned dynamics learning baselines, AC Model (Oh et al., 2015), CDNA (Finn et al., 2016), and OODP (Zhu et al., 2018). AC Model adopts an encoder-LSTM-decoder structure, which performs transformations in hidden space and constructs pixel predictions. CDNA explicitly models pixel motions to achieve invariance to appearance. OODP is designed for class-level dynamics and tries to simultaneously learn object-based representations, relations and motion effects.

4.1. Generalization Ability and Sample Efficiency

We first evaluate zero-shot generalization and sample efficiency on *Monster Kong* from Pygame Learning Environment (Tasfi, 2016), which allows us to test generalization ability over various scenes with different layouts. It is the advanced version of that used by Zhu et al. (2018), which has a more general and complex setting. The monster wanders around and breathes out fires randomly, and the fires also move with some randomness. The agent randomly explores with actions *up*, *down*, *left*, *right*, *jump* and *noop*. All these dynamic objects interact with the environment and other objects according to the underlying physics engine. Moreover, gravity and jump model has a long-term dynamics effects, leading to a partial observation problem. To test whether our model can truly learn the underlying physical mechanism behind the visual observations and perform relational reasoning, we set the k-to-m zero-shot generalization experiment (Figure 3), where we use k different environments for training and m different unseen environments for testing.

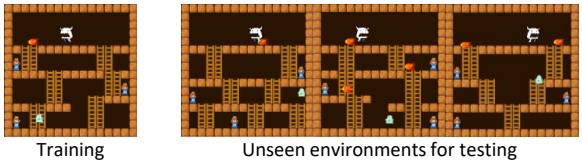


Figure 3. An Example of 1-to-3 zero-shot generalization.

To make a sufficient comparison with previous methods on object dynamics learning and video prediction, we conduct 1-5, 2-5 and 3-5 generalization experiments with a variety of evaluation indices. We use n-error accuracy to measure the performance of object dynamics prediction, which is defined as the proportion that the difference between the predicted and ground-true agent locations is less than n pixel. We also add an extra pixel-based measurement (denoted by object RMSE), which compares the pixel difference near dynamic objects between the predicted and ground-truth images.

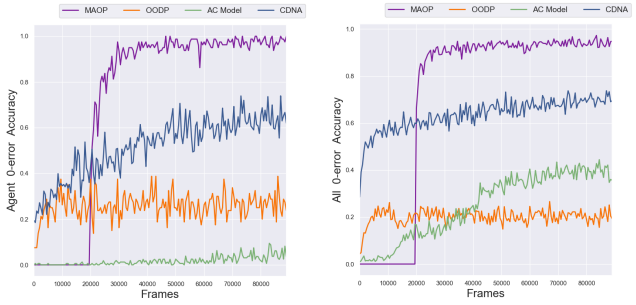


Figure 4. The performance of object dynamics prediction in unseen environments as training progresses on 3-to-5 generalization problem of *Monster Kong*. Since we use the first 20k samples to train the level of dynamic instance segmentation, the curve of MAOP starts at iteration 20001.

As shown in Table 1, MAOP significantly outperforms other methods in all experiment settings in terms of generalization ability and sample efficiency of object dynamics learning. It can achieve 90% 0-error accuracy in unseen environments even trained with 3k samples from a single environment, while other methods have a much lower accuracy (less than 45%). In addition, MAOP with only 3k training samples outperforms CDNA using 300k samples. Although AC Model achieves high accuracy in training environments, its performance in unseen scenes is much worse, which is probably because its pure pixel-level inference easily leads to overfitting. CDNA performs better than AC Model, but still cannot efficiently generalize with limited training samples. Since OODP has structural limitation and optimization difficulty, it has innate difficulty on frames with multiple dynamic objects. In Figure 4 and Appendix Figure A3, we also plot the learning curve of these models. Compared to other models, MAOP achieves higher prediction accuracy for unseen environments at a faster rate during the training process. We further add a video (`https://github.com/maop2019/maop2019/blob/master/PredictionVideo/video.avi`) for better perceptual understanding of prediction performance in unseen environments.

We also evaluate MAOP on *Flappy Bird* and *Freeway*. *Flappy Bird* is a side-scroller game with a moving camera. *Freeway* is an Atari game, which has a large number

Table 1. Prediction performance on *Monster Kong*. k-m means the k-to-m generalization problem. § indicates training with only 3000 samples. ALL represents all dynamic objects. The first column shows the number of samples used for training the models.

	Models	Training environments								Unseen environments							
		1-5§		1-5		2-5		3-5		1-5§		1-5		2-5		3-5	
		Agent	All	Agent	All	Agent	All	Agent	All	Agent	All	Agent	All	Agent	All	Agent	All
Training Samples	MAOP	3k		100k		100k		100k		-		-		-		-	
	OODP	3k		200k		200k		200k		-		-		-		-	
	AC Model	3k		500k		500k		500k		-		-		-		-	
	CDNA	3k		300k		300k		300k		-		-		-		-	
0-error accuracy	MAOP	0.95	0.92	0.98	0.95	0.99	0.96	0.99	0.95	**0.94**	**0.90**	**0.97**	**0.92**	**0.98**	**0.93**	**0.99**	**0.94**
	OODP	0.15	0.15	0.18	0.16	0.22	0.17	0.26	0.20	0.14	0.15	0.20	0.15	0.18	0.15	0.26	0.18
	AC Model	0.01	0.36	0.87	0.94	0.80	0.93	0.77	0.92	0.01	0.20	0.08	0.16	0.30	0.48	0.45	0.66
	CDNA	0.28	0.62	0.77	0.84	0.78	0.82	0.78	0.84	0.26	0.44	0.79	0.80	0.78	0.78	0.81	0.83
Object RMSE	MAOP	24.58		21.96		21.97		23.04		**29.67**		**27.22**		**25.55**		**24.30**	
	OODP	65.63		66.44		66.66		64.73		65.46		67.41		67.78		64.95	
	AC Model	71.02		18.88		22.39		21.30		77.24		57.41		55.45		43.48	
	CDNA	40.92		24.52		24.37		24.18		51.08		37.15		27.67		25.33	

of dynamic objects. Since the testing environments will be similar with the training ones without limitation of samples, we limit the training samples to form a sufficiently challenging generalization task. MAOP still outperforms existing baseline methods (Table 2), which demonstrates that MAOP is effective for the concurrent dynamics prediction of a large number of objects. In addition, we conduct a modular test to better understand the contribution of each learning level (see Appendix Section 4). The results show that each level of MAOP can independently perform well and has a good robustness to the proposals generated by the more abstracted level. Taken together, the above results demonstrates that MAOP has superiority of sample efficiency and generalization ability, which suggests MAOP is good at relational reasoning and learns the object-level dynamics, rather than learn some patterns from mass data to recover the dynamics as the conventional neural networks do.

Table 2. Accuracy of dynamics prediction on *Flappy Bird* and *Freeway*. Since only the agent's ground-true location is accessible in Arcade Learning Environment, we just show the dynamics prediction of the agent. Actually, we observe that predictions of other dynamic objects are also accurate by comparing predicted with ground-true images (Appendix Figure A5).

Models	Flappy Bird (0-acc)				Freeway (Agent)		
	100 samples		300 samples		100 samples		
	Agent	All	Agent	All	0-acc	1-acc	2-acc
MAOP	**0.83**	**0.89**	**0.83**	**0.92**	**0.78**	**0.84**	**0.89**
OODP	0.01	0.18	0.02	0.15	0.26	0.33	0.42
AC Model	0.03	0.18	0.04	0.23	0.31	0.38	0.42
CDNA	0.13	0.77	0.30	0.81	0.42	0.43	0.47

4.2. Model-Based Planning in Unseen Environments

Although RL has achieved considerable successes, most RL researches tend to "train on the test set" (Nichol et al., 2018; Pineau, 2018). It is critical yet challenging to develop model-based RL approaches that support generalization over unseen environments. Monte Carlo tree search (MCTS) (Browne et al., 2012) is developed to leverage the environment models to conduct efficient lookahead search, which has shown remarkable effectiveness on long-term planning, such as AlphaGo (Silver et al., 2016). Considering that our learned dynamics model can efficiently generalize to unseen environments, we can directly use our learned model to perform MCTS in unseen environments. To perform long-range planning, we first test our performance of long-range prediction, as shown in Table 3. MAOP only trained for 1-step prediction can achieve 90% 2-error accuracy in unseen environments when predicting 3 steps of the future, while the accuracy is 73% when predicting 6 steps of the future, which is also a satisfactory performance for lookahead search. Appendix Figure A7 illustrates a case visualizing the 6-step prediction of MAOP in unseen environments.

Table 3. Long-range prediction of MAOP in unseen environments on *Monster Kong*. MAOP is trained for 1-step prediction in 3 environments and tested for 3-step and 6-step prediction in 5 unseen environments.

	0-acc		1-acc		2-acc	
	Agent	All	Agent	All	Agent	All
3-steps	0.81	0.81	0.89	0.87	0.93	0.90
6-steps	0.50	0.53	0.66	0.67	0.74	0.73

We evaluate our performance of model-based planning on *Monster Kong*. In this game, the goal of the agent is to approach the princess and a reward will be given when the straight-line distance from agent to princess gets smaller than that in the agent's history. The value of such a reward is proportional to the shrinking distance. The agent will win with an extra reward +5 when touching the princess, and lose with an extra reward -5 when hitting the fires. To gain a better understanding of the contribution of MAOP to the MCTS agent, we compare MCTS in conjunction with MAOP to DQN (Mnih et al., 2015) and to an ablation (i.e., using the real simulator of the unseen environments in MCTS). We provide the same ground-true reward functions for all dynamics model during MCTS. We conduct random experiments in 5 unseen environments, where the agent and the princess randomly generate. Such a set-

ting is extremely hard for DQN even testing in the training environments, thus we reduce the difficulty for DQN by fixing the princess and shaping the reward function from a POMDP to MDP. As shown in Table 4, MAOP achieves almost the same performance with the true environment model for model-based planning in unseen environments and significantly outperforms DQN. The model-free approach DQN tends to overfit the training environments and cannot learn to plan in unseen environments, leading to a much higher death rate and a much lower score. In addition, we observe that MCTS in conjunction with MAOP acquires intriguing forward-looking skills, such as jumping over the fires and jumping across the big gap that are critical for survival and reaching the goal (see the video https://github.com/maop2019/maop2019/tree/master/MCTSVideo).

Table 4. The performance of using MCTS with different dynamics models, and DQN in unseen environments. REAL indicates the real simulator. Time Out indicates exceeding 100 steps. Reward indicates the accumulative reward that is averaged over 21 runs.

Methods	Reward	Win	Lose	Time Out
MCTS + MAOP	38.19	47.62%	9.52%	42.86%
MCTS + REAL	38.41	52.38%	9.52%	38.10%
DQN	13.67	26.7%	23.8%	49.5%

4.3. Interpretable Representations and Knowledge

MAOP takes a step towards interpretable dynamics model learning. Through interacting with environments, it learns visually and semantically interpretable knowledge in a self-supervised manner, which contributes to unlocking the "black box" of the dynamics prediction and potentially opens the avenue for further researches on object-oriented RL, model-based RL, and hierarchical RL.

Visual Interpretability. To demonstrate the model interpretability of MAOP in unseen environments, we visualize the learned masks of dynamic and static objects. We highlight the attentions of object masks by multiplying the raw images by the binarized masks. Note that MAOP does not require the actual number of objects but a maximum number and some learned object masks may be redundant. Thus, we only show the informative object masks. As shown in Figure 5, our model captures all the key objects in the environments including the controllable agents (cowboy, bird, and chicken), the uncontrollable dynamic objects (monster, fires, pipes and cars), and the static objects that have effects on the motions of dynamic objects (ladders, walls and the free space), which demonstrates that model can learn disentangled object representations and distinguish the objects by both appearance and dynamic property.

Dynamical Interpretability. To show the dynamical interpretability behind image prediction, we test our predicted motions by comparing RMSEs between the predicted and ground-truth motions in unseen environments (Appendix Table A2). Intriguingly, most predicted motions are quite accurate, with the RMSEs less than 1 pixel. Such a visually indistinguishable error also verifies the accuracy of our dynamics learning.

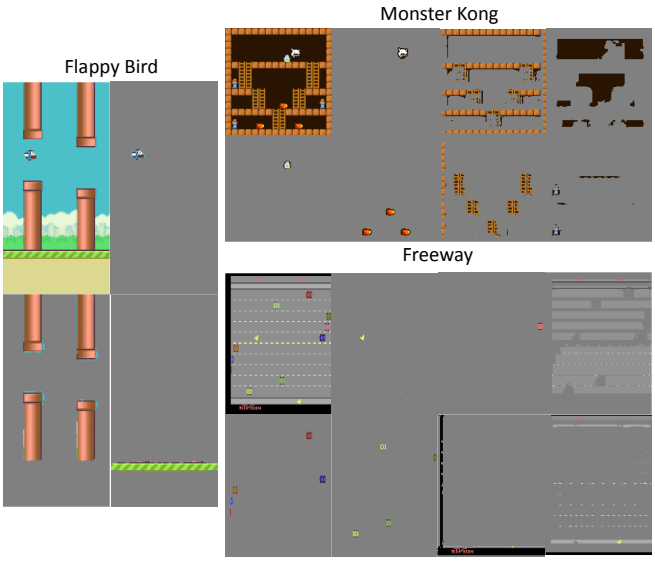


Figure 5. Visualization of the masked images in unseen environments. Top left corner is the raw image.

Discovery of the Controllable Agent. With the learned knowledge in MAOP, we can easily uncover the action-controlled agent from all the dynamic objects, which is useful semantic information that can be used in heuristic algorithms. Specifically, the object that has the maximal variance of total effects over actions is the action-controlled agent. Denote the total effects as $E_i^t = (\sum_j E^t(c_i, j)) + E_{\text{self}}^t(c_i)$, the label of the action-controlled agent is calculated as, $\arg\max_i \sum_t Var_{\boldsymbol{a}^t}(E_i^t)$. We observe that our discovery of the controllable agent achieves right or near 100% accuracy in unseen environments (see Appendix Table A3).

5. Conclusion and Discussion

This paper presents a self-supervised multi-level learning framework for learning action-conditioned object-based dynamics. It enables sample-efficient and interpretable model learning, and achieves zero-shot generalization over novel environments with multiple dynamic objects and different static object layouts. The learned dynamics model enables an agent to directly plan in unseen environments. Our future work includes extending our model for deformation prediction (e.g., object appearing, disappearing and non-rigid deformation) and incorporating a camera motion prediction network module introduced by (Vijayanarasimhan et al., 2017) for applications such as FPS games and autonomous driving. Learning 3D dynamics from 2D video is extremely challenging. Conventional neural networks try to learn such

3D dynamics by remembering some patterns from 2D data as they do for the non-rigid deformation, such as AC Model (Oh et al., 2015) and CDNA (Finn et al., 2016). This approach achieves good performance in training environments, but it requires a large number of data and does not really recover the true 3D dynamics model. To learn generalized 3D dynamics model, object-oriented learning paradigm in conjunction with 3D CNN (3D data input) is necessary, which is an important direction for future work.

References

Battaglia, P., Pascanu, R., Lai, M., Rezende, D. J., et al. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pp. 4502–4510, 2016.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

Chang, M. B., Ullman, T., Torralba, A., and Tenenbaum, J. B. A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*, 2016.

Chiappa, S., Racaniere, S., Wierstra, D., and Mohamed, S. Recurrent environment simulators. *International Conference on Learning Representations*, 2017.

Cobo, L. C., Isbell, C. L., and Thomaz, A. L. Object focused q-learning for autonomous agents. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 1061–1068. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

Dai, J., He, K., Li, Y., Ren, S., and Sun, J. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, pp. 534–549. Springer, 2016.

Diuk, C., Cohen, A., and Littman, M. L. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 240–247. ACM, 2008.

Finn, C. and Levine, S. Deep visual foresight for planning robot motion. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 2786–2793. IEEE, 2017.

Finn, C., Goodfellow, I., and Levine, S. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pp. 64–72, 2016.

Fragkiadaki, K., Agrawal, P., Levine, S., and Malik, J. Learning visual predictive models of physics for playing billiards. *International Conference on Learning Representations*, 2016.

Girshick, R. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Gu, S., Lillicrap, T., Sutskever, I., and Levine, S. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, pp. 2829–2838, 2016.

Guestrin, C., Koller, D., Gearhart, C., and Kanodia, N. Generalizing plans to new environments in relational mdps. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pp. 1003–1010. Morgan Kaufmann Publishers Inc., 2003.

Guo, X., Wang, X., Yang, L., Cao, X., and Ma, Y. Robust foreground detection using smoothness and arbitrariness constraints. In *European Conference on Computer Vision*, pp. 535–550. Springer, 2014.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2980–2988. IEEE, 2017.

Heess, N., Wayne, G., Silver, D., Lillicrap, T., Erez, T., and Tassa, Y. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, pp. 2944–2952, 2015.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456, 2015.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pp. 2017–2025, 2015.

Johansson, G. Visual motion perception. *Scientific American*, 232(6):76–89, 1975.

Kansky, K., Silver, T., Mély, D. A., Eldawy, M., Lázaro-Gredilla, M., Lou, X., Dorfman, N., Sidor, S., Phoenix, S., and George, D. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. *arXiv preprint arXiv:1706.04317*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Lee, D.-S. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (5):827–832, 2005.

Li, Y., Qi, H., Dai, J., Ji, X., and Wei, Y. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4438–4446, 2017.

Liu, B., He, X., and Gould, S. Multi-class semantic video segmentation with exemplar-based object reasoning. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1014–1021. IEEE, 2015.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., and Pietikäinen, M. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018.

Lo, B. and Velastin, S. Automatic congestion detection system for underground platforms. In *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pp. 158–161. IEEE, 2001.

Lu, Z.-L. and Sperling, G. The functional architecture of human visual motion perception. *Vision research*, 35(19): 2697–2722, 1995.

Maddalena, L., Petrosino, A., et al. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7):1168, 2008.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.

Nichol, A., Pfau, V., Hesse, C., Klimov, O., and Schulman, J. Gotta learn fast: A new benchmark for generalization in rl. *arXiv preprint arXiv:1804.03720*, 2018.

Oh, J., Guo, X., Lee, H., Lewis, R. L., and Singh, S. Action-conditional video prediction using deep networks in atari games. In *Advances in Neural Information Processing Systems*, pp. 2863–2871, 2015.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*, volume 2017, 2017.

Pineau, J. Reproducible, reusable, and robust reinforcement learning (Invited Talk). *Advances in Neural Information Processing Systems*, 2018.

Pinheiro, P. O., Collobert, R., and Dollár, P. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pp. 1990–1998, 2015.

Racanière, S., Weber, T., Reichert, D., Buesing, L., Guez, A., Rezende, D. J., Badia, A. P., Vinyals, O., Heess, N., Li, Y., et al. Imagination-augmented agents for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 5694–5705, 2017.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pp. 4974–4983, 2017.

Schmidhuber, J. On learning to think: Algorithmic information theory for novel combinations of reinforcement learning controllers and recurrent neural world models. *arXiv preprint arXiv:1511.09249*, 2015.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Smith, A. T., Greenlee, M. W., Singh, K. D., Kraemer, F. M., and Hennig, J. The processing of first-and second-order motion in human visual cortex assessed by functional magnetic resonance imaging (fmri). *Journal of Neuroscience*, 18(10):3816–3830, 1998.

Tasfi, N. Pygame learning environment. https://github.com/ntasfi/PyGame-Learning-Environment, 2016.

Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. *arXiv preprint arXiv:1802.10353*, 2018.

Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., and Fragkiadaki, K. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pp. 2746–2754, 2015.

Watters, N., Zoran, D., Weber, T., Battaglia, P., Pascanu, R., and Tacchetti, A. Visual interaction networks. In *Advances in Neural Information Processing Systems*, pp. 4540–4548, 2017.

Wu, J., Lu, E., Kohli, P., Freeman, B., and Tenenbaum, J. Learning to see physics via visual de-animation. In *Advances in Neural Information Processing Systems*, pp. 152–163, 2017.

Zambaldi, V., Raposo, D., Santoro, A., Bapst, V., Li, Y., Babuschkin, I., Tuyls, K., Reichert, D., Lillicrap, T., Lockhart, E., et al. Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830*, 2018.

Zhang, C. and Shah, J. A. Co-optimizating multi-agent placement with task assignment and scheduling. In *IJCAI*, pp. 3308–3314, 2016.

Zhou, X., Yang, C., and Yu, W. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):597–610, 2013.

Zhu, G., Huang, Z., and Zhang, C. Object-oriented dynamics predictor. *Advances in Neural Information Processing Systems*, 2018.

6. Appendix

6.1. Multi-Level Abstraction Framework

Algorithm A1 shows a pseudocode that summarizes the overall architecture of our multi-level abstraction framework (Section 3 in the main body).

6.2. Object-Oriented Dynamics Learning Paradigm

Algorithm A2 illustrates the learning paradigm of object based dynamics and the interactions of its components (Section 3.1 in the main body).

6.3. Instance Localization

Instance localization is a common technique in context of supervised region-based object detection (Girshick et al., 2014; Girshick, 2015; Ren et al., 2015; He et al., 2017; Liu et al., 2018), which localizes objects on raw images with regression between the predicted bounding box and the ground truth. Here, we propose an unsupervised approach to perform dynamic instance localization on dynamic object masks learned by Object Detector. Our objective is to sample a number of region proposals on the dynamic object masks and then select the regions, each of which has exactly one dynamic instance. In the rest of this section, we will describe these two steps in detail.

Region proposal sampling. We design a learning-free sampling algorithm for sampling region proposals on object masks. This algorithm generates multi-scale region proposals with a full coverage over the input mask. Actually, we adopt multi-fold full coverage to ensure that pixels of the potential instances are covered at each scale. The detailed algorithm is described in Algorithm A3.

Instance mask selection. Instance mask selection aims at selecting the regions, each of which contains exactly one dynamic instance, based on the discrepancy loss $\mathcal{L}_{\text{instance}}$ (Section 3.1 in the main body). To screen out high-consistency, non-overlapping and non-empty instance masks at the same time, we integrate Non-Maximum Suppression (NMS) and Selective Search (SS) (Uijlings et al., 2013) in the context of region-based object detection (Girshick et al., 2014; Girshick, 2015; Ren et al., 2015; He et al., 2017; Liu et al., 2018) into our algorithm.

6.4. Modular Test

We conduct a modular test to better understand the contribution of each learning level. First, we investigate whether the level of dynamics learning can learn the accurate dynamics model when the coarse region proposals of dynamic instances are given. We remove the other two levels and replace them by the artificially synthesized coarse proposals of dynamic instances to test the independent performance of the dynamics learning level. Specifically, the synthesized data are generated by adding standard Gaussian or Poisson noise on ground-true dynamic instance masks (Figure A5). As expected, the level of dynamics learning can learn accurate dynamics of all dynamic objects given coarse proposals of dynamic instances (Table A1). Similarly, we test the independent performance of the dynamics instance segmentation level. We replace the foreground proposal generated by the motion detection level with the artificially synthesized noisy foreground proposal. Figure A6 demonstrates our learned dynamic instances in the level of dynamic instance segmentation, which demonstrates the competence of the dynamic instance segmentation level. Taken together, the modular test shows that each level of MAOP can independently perform well and has a good robustness to the proposals generated by the more abstracted level.

6.5. Supplementary Tables and Figures

In addition to the above mentioned tables and figures, here we supplement the rest of supplementary tables and figures, that is, Table A2 (mentioned in Section 4.3 of the main body), Table A3 (mentioned in Section 4.3 of the main body), Figure A3 (mentioned in Section 4.1 of the main body), Figure A7 (mentioned in Section 4.2 of the main body), and Figure A4 (mentioned in Table 2 of the main body).

6.6. Implementation Details for Experiments

The neural network architecture of the dynamic instance segmentation level (consisting of Instance Splitter and Merging Net) is shown as Figure A1. Object Detector in the dynamics learning level has the similar architecture with Instance Splitter. The CNNs in Object Detector are shown as Figure A2.

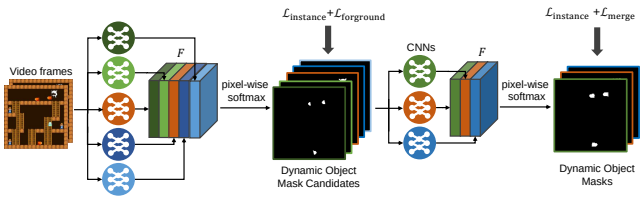


Figure A1. Architecture of the dynamic instance segmentation level, which consists of Instance Splitter and Merging Net.

Denote $Conv(F, K, S)$ as the convolutional layer with the number of filters F, kernel size K and stride S. Let $R(), S()$ and $BN()$ denote the ReLU layer, sigmoid layer and batch normalization layer (Ioffe & Szegedy, 2015). The CNNs in Merging Net are connected in the order: $R(BN(Conv(32, 1, 1)))$, $R(BN(Conv(1, 1, 1)))$. The 6 convolutional layers (Figure A2) in Object Detector can be indicated as $R(BN(Conv(32, 3, 1)))$,

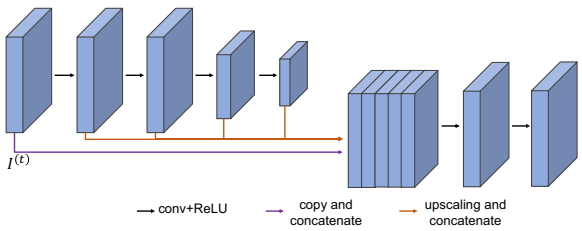


Figure A2. Architecture of the CNNs in Object Detector.

$R(BN(Conv(32,3,1)))$, $R(BN(Conv(16,3,2)))$, $R(BN(Conv(16,3,2)))$, $R(BN(Conv(32,1,1)))$ and $BN(Conv(1,3,1))$, respectively. The CNNs in Relation Net are connected in the order: $R(BN(Conv(16,3,2)))$, $R(BN(Conv(16,3,2)))$, $R(BN(Conv(16,3,2)))$, and $R(BN(Conv(16,3,2)))$. The last convolutional layer is reshaped and fully connected by the 64-dimensional hidden layer and the 2-dimensional output layer successively. Inertia Net has the same architecture and hyperparameters as Relation Net.

The detailed experimental settings and hyperparameters for training MAOP on *Monster Kong*, *Flappy Bird* and *Freeway* are listed as follows:

- We use random exploration on *Monster Kong*. We adopt an expert guided random exploration on *Flappy Bird* and *Freeway*, because a totally random exploration will lead to an early death of the agent even at the very beginning. Although we use these exploration methods in our experiments, our framework can support smarter exploration strategies, such as curiosity-driven exploration (Pathak et al., 2017).
- The weights of losses, λ_1, λ_2, λ_3, λ_4 are 100, 1, 10, and 10, respectively. And in these three games, we consider that static mask is a dynamic object whose motion is 0 and the weight of $\mathcal{L}_{\text{instance}}$ for this 0-motion dynamic object mask is 100. In addition, all the l_2 losses are divided by HW to keep invariance to the image size.
- The decay rate α in background memory is 0.9.
- Batch size is 8 and the maximum number of training steps of Dynamics Learning and Instance Segmentation are set to 5×10^4 and 1×10^5, respectively.
- The optimizer is Adam (Kingma & Ba, 2014) with learning rate 1×10^{-3}.
- The raw images of *Monster Kong*, *Flappy Bird* and *Freeway* are resized to $160 \times 160 \times 3$, $160 \times 80 \times 3$, and $160 \times 120 \times 3$, respectively.
- The size of the horizon window w is 33 on *Monster Kong*, 41 on *Flappy Bird*, and 33 on *Freeway*.
- The maximum number of static masks is 8 on *Monster Kong*, 3 on *Flappy Bird* and 6 on *Freeway*.
- The maximum number of dynamic object masks (the output masks of Object Detector and Merging Net) is 5 on *Monster Kong*, 6 on *Flappy Bird* and 12 on *Freeway*. To encourage Instance Splitter to generate more dynamic object mask candidates, we set the maximum number of dynamic object masks outputted by Instance Splitter to be 8 on *Monster Kong*, 15 on *Flappy Bird* and 20 on *Freeway*.
- The maximum instance number of each dynamic object class is set to 6 on *Monster Kong*, 15 on *Flappy Bird* and 6 on *Freeway*.
- The size of mutli-sacle region proposals are $10 \times 10, 20 \times 20, 40 \times 40, 80 \times 80$, respectively.
- To augment the interactions of instances when training Instance Splitter, we random sample two region proposals and combine them into a single region proposal with double size.

The detailed hyperparameters for running MCTS with MAOP, OODP, CDNA, AC Model, and real simulator on *Monster Kong* are listed as follows:

- The number of trajectories is 500.
- The maximum-depth of each trajectory is 6.
- The exploration parameter used in Upper Confidence Bounds for Trees (UCT) is 5.
- The number of rollouts in each simulation is 8.
- At the end of each search, the agent selects the action with maximum visit count.

Algorithm A1 Overall Multi-Level Abstraction Framework.

1: Initialization. Initialize the parameters of all neural networks with random weights respectively.
2: Motion Detection Level. Perform foreground detection to produce dynamic region proposals, which potentially have moving objects.
3: Instance Segmentation Level. Train the dynamic instance segmentation network (including Instance Splitter and Merging Net) by minimizing $\mathcal{L}_{\text{DIS}}$, which includes a proposal loss to focus the dynamic instance segmentation on the dynamic region proposals from Step 2.
4: Dynamic learning Level. Train the dynamics learning network (whose forward process is shown as Algorithm A2) by minimizing $\mathcal{L}_{\text{DL}}$, which includes a proposal loss to utilize the dynamic instance proposals generated by the trained dynamic instance segmentation network in Step 3 to facilitate the learning of Object Detector.

Algorithm A2 Object-Oriented Dynamics Learning.

Input: A sequence of video frames $\mathbf{I}^{t-h:t}$ with length h, input action $\boldsymbol{a}^t$ at time t.

Object masks $\mathbf{O}^{t-h:t} \leftarrow$ ObjectDetector($\mathbf{I}^{t-h:t}$), $\mathbf{O}$ include dynamic and static masks $\mathbf{D}, \mathbf{S}$
Instance masks $\mathbf{X}^{t-h:t} \leftarrow$ InstanceLocalization($\mathbf{I}^{t-h:t}, \mathbf{D}^{t-h:t}$)
Predicted instance masks $\hat{\mathbf{X}}^{t+1} \leftarrow \varnothing$
for each instance mask $\boldsymbol{x}$ in $\mathbf{X}$ **do**
  Effects from spatial relations $\boldsymbol{m}_1^t \leftarrow$ RelationNet($\boldsymbol{x}^t, \mathbf{O}^t, \boldsymbol{a}^t$)
  Effects from temporal relations $\boldsymbol{m}_2^t \leftarrow$ InertiaNet($\boldsymbol{x}^{t-h:t}, \boldsymbol{a}^t$)
  Total effects $\boldsymbol{m}^t \leftarrow \boldsymbol{m}_1^t + \boldsymbol{m}_2^t$
  Predicted instance mask $\hat{\boldsymbol{x}}^{t+1} \leftarrow$ Transformation($\boldsymbol{x}^t, \boldsymbol{m}^t$)
  $\hat{\mathbf{X}}^{t+1} \leftarrow \hat{\mathbf{X}}^{t+1} \bigcup \hat{\boldsymbol{x}}^{t+1}$
end for
Background image $\mathbf{B}^{t+1} \leftarrow$ BackgroundConstructor($\mathbf{I}^t, \mathbf{S}^t$)
Predicted next frame $\hat{\mathbf{I}}^{t+1} \leftarrow$ Merge($\hat{\mathbf{X}}^{t+1}, \mathbf{B}^{t+1}$)

Algorithm A3 Region Proposal Sampling.

Input: Dynamic object mask $\mathbf{D} \in [0,1]^{H \times W}$, the number of region proposal scales n_S, the folds of full coverage T .

Initialize proposal set $\mathbb{P} = \varnothing$.
Binarize $\mathbf{D}$ to get the indicator for the existence of objects
for $l = 1 \ldots n_S$ **do**
  Select scale dx, dy depend on the level l.
  for $t = 1 \ldots T$ **do**
    Initialize candidate set $\mathbb{C} = \{(i,j) | \mathbf{D}_{i,j} = 1\}$.
    while $\mathbb{C} \neq \varnothing$ **do**
      Sample a pixel coordinate (x, y) from $\mathbb{C}$.
      Get a box $\mathbb{B} = \{(i,j) |\, |i - x| \leq dx, |j - y| \leq dy\}$.
      if $\mathbb{B}$ is not empty **then**
        Insert $\mathbb{B}$ into the proposal set $\mathbb{P} \leftarrow \mathbb{P} \cup \{\mathbb{B}\}$.
      end if
      Update the remain candidate set $\mathbb{C} \leftarrow \mathbb{C} \setminus \mathbb{B}$.
    end while
  end for
end for

Return: $\mathbb{P}$

Table A1. Prediction performance of the dynamic instance level with different region proposals of dynamic instances on 3-to-5 generalization problem of *Monster Kong*. "All" represents all dynamic objects. "Computed by DIS" refers to using the proposal regions of dynamic instances computed from the level of dynamic instance segmentation in MAOP.

Noise type of proposals	Training environments						Unseen environments					
	0-acc		1-acc		2-acc		0-acc		1-acc		2-acc	
	Agent	All	Agent	All	Agent	All	Agent	All	Agent	All	Agent	All
Computed by DIS	0.99	0.95	1.00	0.97	1.00	0.97	0.99	0.94	1.00	0.96	1.00	0.97
Gaussian Noise	0.63	0.57	0.94	0.89	0.99	0.95	0.60	0.57	0.93	0.89	0.98	0.95
Poisson Noise	0.93	0.91	0.98	0.95	0.99	0.96	0.93	0.91	0.99	0.96	0.99	0.96

Table A2. Average motion prediction error in two experiment environments. †, ‡ and § indicate training with only 100, 300 and 3000 samples. ALL represents all dynamic objects.

Model	Monster Kong				Flappy Bird	
	1-5§	1-5	2-5	3-5	1-5†	1-5‡
MAOP	0.34	0.15	0.14	0.12	0.30	0.34

Table A3. Accuracy of our discovery of controllable agent in unseen environments.

Model	MonsterKong	FlappyBird	Freeway
MAOP	100%	100%	98.75%

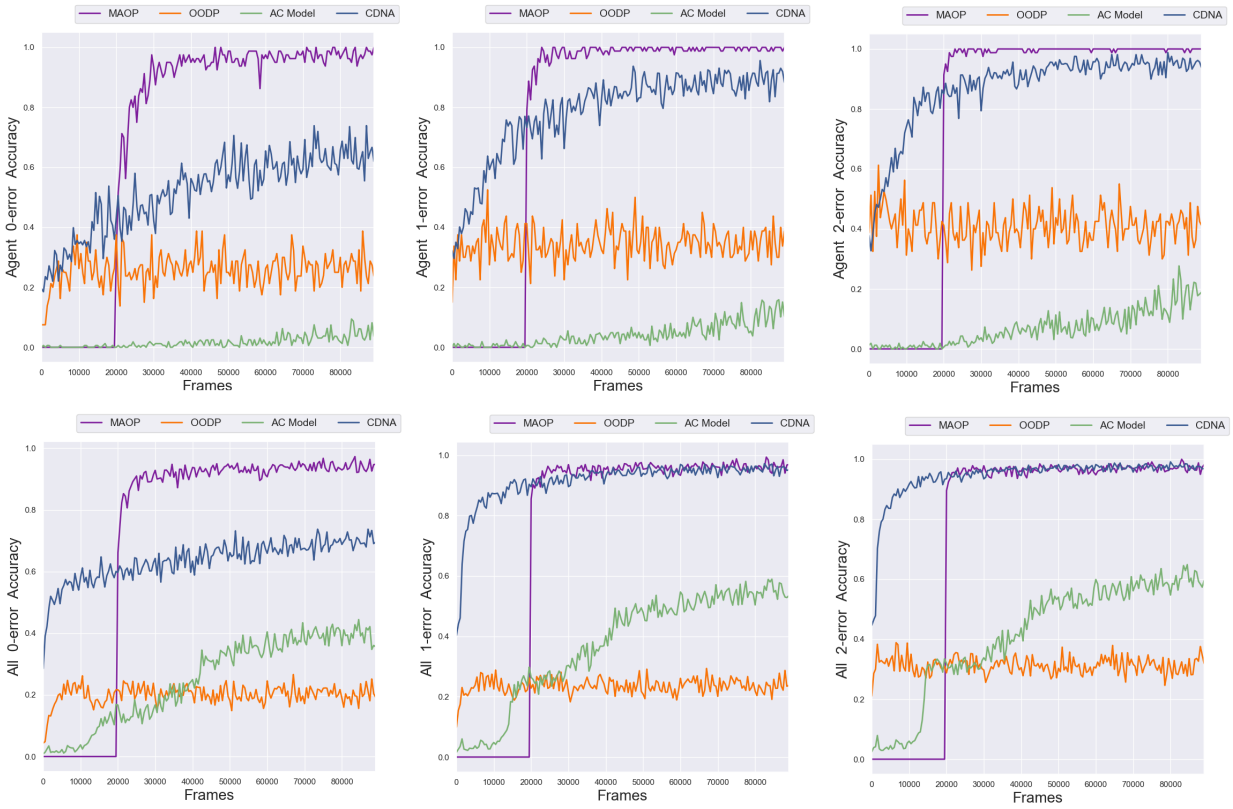


Figure A3. The performance of object dynamics prediction in unseen environments as training progresses on 3-to-5 generalization problem of *Monster Kong*. "Agent" represents the dynamics of the agent, while "all" represents the dynamics of all dynamic objects. Since we use the first 20k samples to train the level of dynamic instance segmentation, the curve of MAOP starts at iteration 20001.

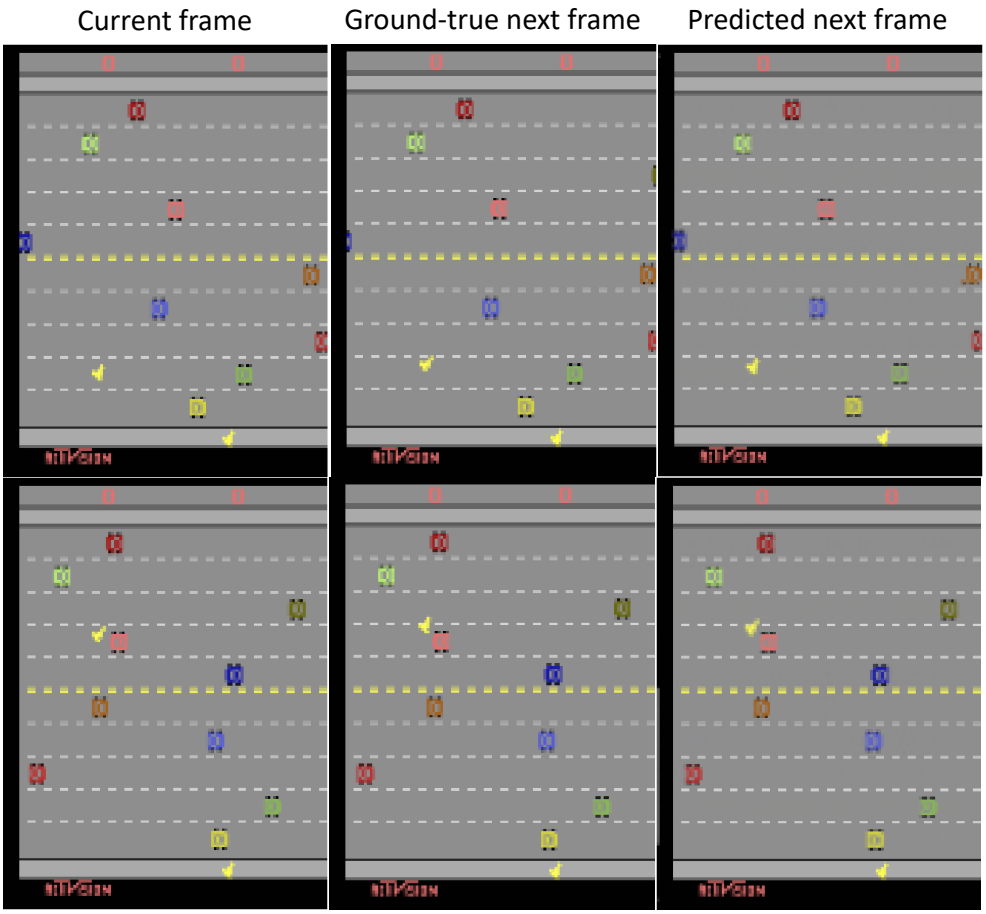


Figure A4. Image predictions in testing environments on *Freeway*. Since just the agent's ground-true location is accessible in Arcade Learning Environment, we can only examine the predictions of other dynamic objects by comparing the predicted with ground-true images. These two samples are not cherry-picked. From the figure, we can observe that the errors between the predicted and ground-true images are visually indistinguishable, which suggests that our prediction of all dynamic objects are accurate.

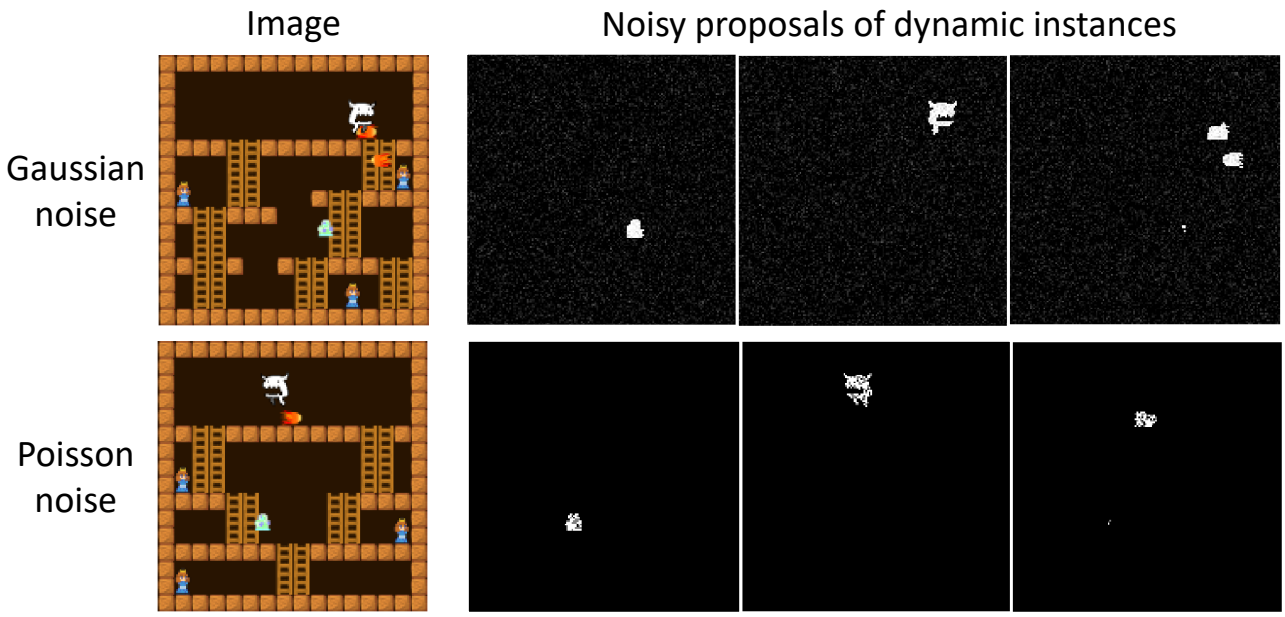


Figure A5. Noisy region proposals of dynamic instances. Zoom in to see the details.

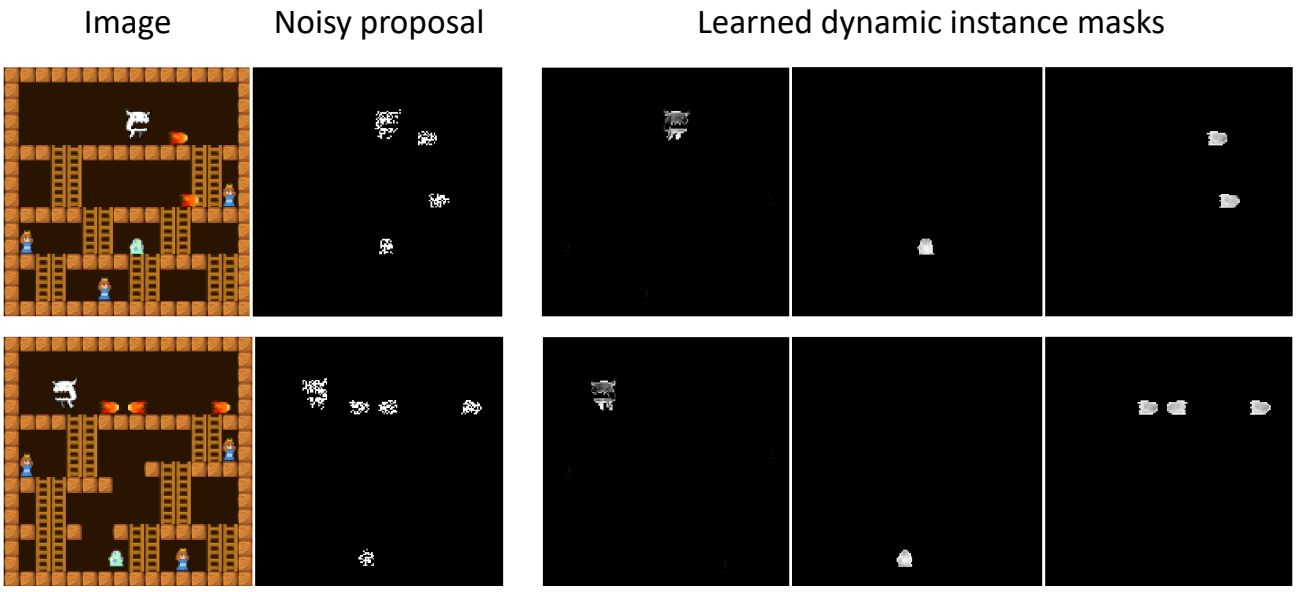


Figure A6. The learned dynamic instance masks in the level of dynamic instance segmentation with noisy foreground proposals.

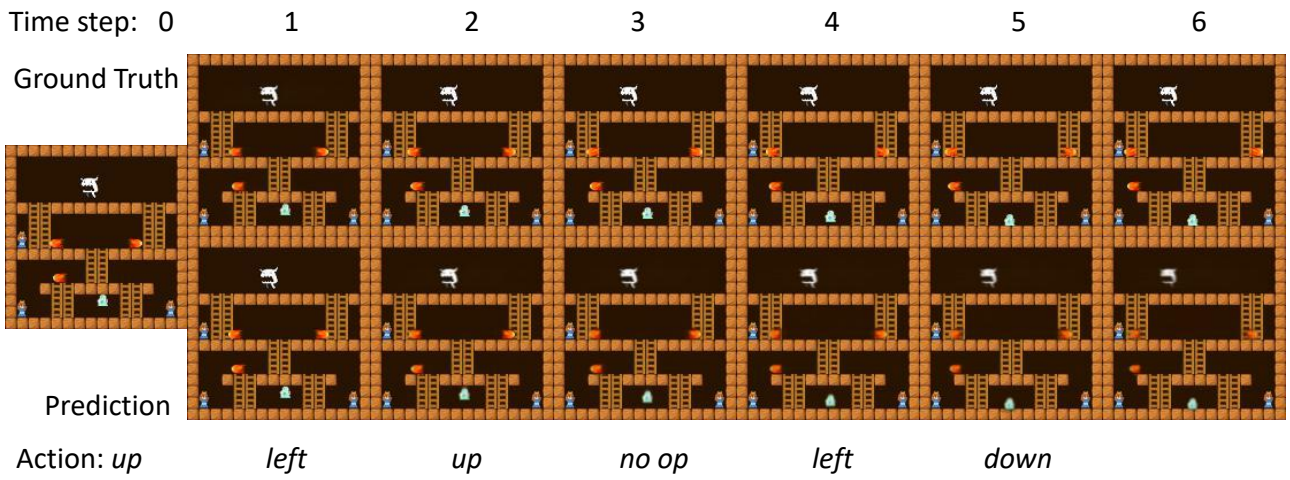


Figure A7. A case shows the 6-step prediction of our model in unseen environments on 3-to-5 generalization problem of *Monster Kong*.