Learning Long-term Visual Dynamics with Region Proposal Interaction Networks

Haozhi Qi[1] **Xiaolong Wang**[2] **Deepak Pathak**[3] **Yi Ma**[1] **Jitendra Malik**[1]
[1]UC Berkeley [2]UC San Diego [3]CMU

Abstract

Learning long-term dynamics models is the key to understanding physical common sense. Most existing approaches on learning dynamics from visual input sidestep long-term predictions by resorting to rapid re-planning with short-term models. This not only requires such models to be super accurate but also limits them only to tasks where an agent can continuously obtain feedback and take action *at each step* until completion. In this paper, we aim to leverage the ideas from success stories in visual recognition tasks to build object representations that can capture inter-object and object-environment interactions over a long range. To this end, we propose *Region Proposal Interaction Networks (RPIN)*, which reason about each object's trajectory in a latent region-proposal feature space. Our approach outperforms prior methods by a significant margin both in terms of prediction quality and their ability to plan for downstream tasks, and also generalize well to novel environments. Results are available at https://sites.google.com/view/orlr-workshop-rpin.

1 Introduction

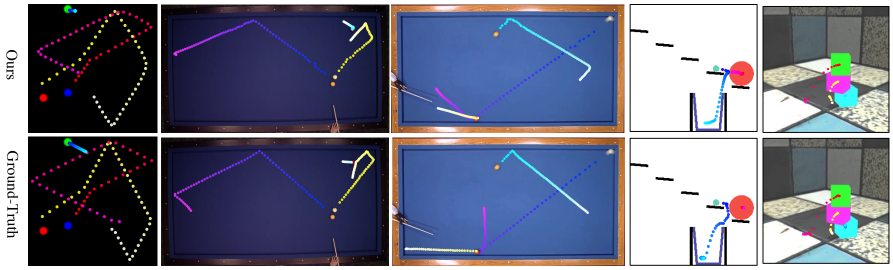
As argued by Kenneth Craik, *if an organism carries a model of external reality and its own possible actions within its head, it is able to react in much fuller, safer and more competent manner to emergencies which face it* [4]. Indeed, building prediction models has been long studied in computer vision and intuitive physics. In vision, most approaches make predictions in pixel-space [5, 6, 10, 13, 18], which ends up capturing the optical flow [18] and is difficult to generalize to long-horizon. In intuitive physics, a common approach is to learn the dynamics directly in an abstracted state space of objects to capture Newtonian physics [2, 3, 16]. However, the states end up being detached from raw sensory perception. Unfortunately, these two extremes have barely been connected. In this paper, we argue for a middle-ground to treat images as a window into the world, i.e., objects exist but can be accessed only via images. Images are neither to be used for predicting pixels nor to be isolated from dynamics. We operationalize it by learning to extract a rich state representation directly from images and build dynamics models using the extracted state representations.

It is difficult to make predictions, especially about the future — Niels Bohr

Contrary to Niels Bohr, predictions are, in fact, easy if made only for the short-term. Predictions that are indeed difficult to make and actually matter are the ones made over the long-term. Consider the example of "Three-cushion Billiards" in Figure 1 (b) and (c). The goal is to hit the cue in such a way that it touches the other two balls and contacts the wall thrice before hitting the last ball. This task is extremely challenging even for human experts because the number of successful trajectories is very sparse. Do players perform classical Newtonian physics calculations to obtain the best action before each shot, or do they just memorize the solution by practicing through exponentially many configurations? Both extremes are not impossible, but often impractical. Players rather build a physical understanding by experience [12, 14, 15] and plan by making intuitive, yet accurate predictions in the long-term. Hence, in this work, we focus primarily on the long-term aspect of prediction by just considering environments, such as the three-cushion billiards example or the PHYRE [1] in Figure 1 (d), where an agent is allowed to take *only one* action in the beginning so as to preclude any scope of re-planning.

How to learn an accurate dynamics model has been a popular research topic for years. Recently, there are a series of work trying to represent video frames using object-centric representations [2, 3, 9, 11,

NeurIPS 2020 Workshop on Object Representations for Learning and Reasoning, Vancouver, Canada.



(a) Sim-Billiard (b) Real-Billiard (c) Real-Billiard (d) PHYRE (e) ShapeStacks

Figure 1: Visualization on all of the four datasets. The first row is our prediction results and the second row is the ground-truth trajectories. Our method accurately predicts long-term future even after complex interactions.

19, 21]. However, those methods are either operates in the state space, or ignore the environment information, both of which are not practical in real-world scenarios. In contrast, our objective is to build a data-driven prediction model that can both: (a) model long-term interactions over time to plan successfully for new instances, and (b) work from raw visual input in complex real-world environments. To this end, we propose Region Proposal Interaction Network (RPIN) which contains two key components. Firstly, we leverage the region of interests pooling (RoIPooling) operator [7] to extract object features maps from the frame-level feature. By using RoIPooling, each object feature not only contains its own information but also the context of the environment. Secondly, we extend the Interaction Network and propose Convolutional Interaction Networks that perform interaction reasoning on the extracted RoI feature maps. Interaction Networks is originally proposed in [2], where the interaction reasoning is conducted via MLPs. By changing MLPs to convolutions, we can effectively utilize the spatial information of an object and make accurate future prediction of object location and shapes changes. Notably, our approach is simple, yet outperforms the state-of-the-art object feature extraction methods in both simulation and real datasets. Our method reduces the prediction error by 75% in the complex PHYRE environment and achieves state-of-the-art on the PHYRE cross-task generalization setting.

2 Region Proposal Interaction Networks

Our model takes N video frames and the corresponding object bounding boxes at each frame as inputs, and outputs the objects' bounding boxes and masks for the future T timesteps. The overall model structure is illustrated in Figure 2. For each frame, we first extract the image features using a ConvNet. Then we apply RoIPooling [7, 8] to obtain the object-centric visual features. These features are then forwarded to our Convolutional Interaction Networks (CIN) to perform objects' interaction reasoning and used to predict future object bounding boxes and masks.

Convolutional Interaction Networks The original interaction network is a general-purpose model to learn and predict future physical dynamics. It takes the feature representation of m objects at timestep t: $X = \{x_1^t, x_2^t, ..., x_m^t \,|\, x_i^t \in \mathbb{R}^d\}$ and performs object reasoning f_O as well as relational reasoning f_R on these features. Specifically, the updated rule of object features can be described as:

$$e_i^t = f_A\big(f_O(x_i^t) + \textstyle\sum_{j \neq i} f_R(x_i^t, x_j^t)\big), \quad z_i^t = f_Z(x_i^t, e_i^t), \quad x_i^{t+1} = f_P\big(z_i^t, z_i^{t-1}, \dots, z_i^{t-k}\big). \tag{1}$$

In the above equation, f_A is the function to calculate the effect of both of object reasoning and relational reasoning results. And f_Z is used to combine the original object state and the reasoning effect. Finally, f_P is used to do future state predictions based on one or more previous object states. In IN, $f_{O,R,A,Z,P}$ are instantiated by a fully-connected layer with learnable weight.

The input of Convolutional Interaction Network (CIN) is m object features at timestep t: $X = \{x_1^t, x_2^t, ..., x_m^t | x_i^t \in \mathbb{R}^{d \times h \times w}\}$. The high-level update rule is the same as IN, but the key difference is that we use convolution to instantiate $f_{O,R,A,Z,P}$. Such instantiation is crucial to utilize the spatial information encoded in our object feature map and to effectively reason future object states. Specifically, we have

$$f_R(x_i^t, x_j^t) = W_R * [x_i^t, x_j^t] \quad f_A(x_i^t) = W_A^T * x_i^t \quad f_O(x_i^t) = W_O * x_i^t \tag{2}$$

$$f_Z(x_i^t, e_i^t) = W_Z * [x_i^t, e_i^t] \quad f_P(z_i^t, z_i^{t-1}, ..., z_i^{t-k}) = W_P * [z_i^t, z_i^{t-1}, ..., z_i^{t-k}] \tag{3}$$

One can plug the functions in Equation 1 for better understanding the operations. In the above equations, $*$ denotes the convolution operator, and $[\cdot, \cdot]$ denotes concatenation along the channel dimension. $W_{R,Z,O,A,P}$ are learnable weights of the convolution kernels with kernel size 3×3.

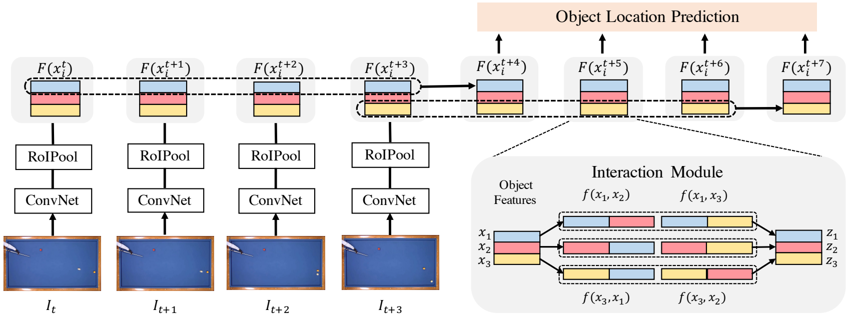


Figure 2: Our Region Proposal Interaction Network. Given N frames as inputs, we forward them to an encoder network, and then extract the foreground object features with RoIPooling (different colors represent different instances). We perform interaction reasoning on top of the region proposal features (gray box, bottom right).

Learning Region Proposal Interaction Networks (RPIN) Our model predicts the future bounding box and (optionally) masks of each object. Given the predicted feature x_i^{t+1}, we use a simple two layer MLP decoder to estimate its bounding boxes coordinates and masks. The bounding box decoder takes a flattened object feature map as input, and outputs 4-d vector, representing the center and size of the box. The mask decoder is of the same architecture but has 21×21 output channels, representing a 21×21 binary masks inside the corresponding bounding boxes. We use the ℓ^2 loss for bounding box predictions. For mask prediction, we use spatial cross-entropy loss which sums the cross-entropy values of a 21×21 predicted positions.

3 Evaluation Results: Prediction, Generalization, and Planning

Datasets. We evaluate our method's prediction performance on four different datasets: 1) PHYRE [1]; 2) ShapeStacks (SS) [21]; 3) Real World Billiards (RealB), which is the "Three-cushion Billiards" game video downloaded from YouTube; 4) Simulation Billiards (SimB), which is a simple simulated environments. The full dataset details are in the appendix. We predict object masks for PHYRE and ShapeStacks, and only predict bounding boxes for Billiard datasets.

Baselines. 1) *VIN* [11, 19]: Instead of using object-centric spatial pooling to extract object features, it use a ConvNet to globally encode an image to a fixed $d \times m$ dimensional vector. 2) *Object Masking (OM)* [9, 17, 20]: This approach takes one image and m object bounding boxes or masks as input. For each proposal, only the pixels inside object proposals are kept while others are set to 0, leading to m masked images. 3) *CVP:* The object feature is extracted by cropping the object image patch and forwarding it to an encoder [21, 22]. Since the object features are directly extracted from the raw image patches, the context information is also ignored. We re-implement CVP's feature extraction method within our framework. We show we can reproduce their results in the appendix.

3.1 How accurate is the predicted dynamics?

To evaluate how well the world dynamics is modeled, we first report the average prediction errors on the test split, over *the same time-horizon* as which model is trained on, i.e., $t \in [0, T_{\text{train}}]$. The prediction error is calculated by the squared ℓ_2 distance between predicted object center location and the ground-truth object centers. The results are shown in Table 1 (left half).

Firstly, we show the effectiveness of our proposed RoI Feature by comparing Table 1 VIN, OM, CVP, and Ours (IN). These four entries use the same backbone network and interaction network modules. Our results are significantly better than all of the baselines. In the very challenging PHYRE dataset, the prediction error is only $1/4$ of the best baseline. In the other three easier datasets, the gap is not as large but our method still achieves more than 10% improvements. This demonstrates the advantage of using rich state representations. Secondly, we show that the effectiveness of our proposed Convolutional Interaction Network by comparing Table 1 Ours (IN) and Ours (CIN). With every other components the same, changing the vector-form representation to spatial feature maps and use convolution to model the interactions can further improve the performance by 10%∼20%. This result shows our convolutional interaction network could better utilize the spatial information encoded in the object feature map.

In Table 1 (right half), we report the prediction error for $t \in [T_{\text{train}}, 2 \times T_{\text{train}}]$. The results in this setting are consistent with what we found in Section 3.1. Our method still achieves the best performance against all baselines. Specifically, for all datasets except SimB, we reduce the error

method	$t \in [0, T_{\text{train}}]$				$t \in [T_{\text{train}}, 2 \times T_{\text{train}}]$			
	PHYRE	SS	RealB	SimB	PHYRE	SS	RealB	SimB
VIN	N.A.	2.47	1.02	3.89	N.A.	7.77	5.11	29.51
OM	8.15	3.01	0.59	3.48	25.25	9.51	3.23	28.87
CVP	62.71	2.84	3.57	80.01	79.26	7.72	6.63	108.56
Ours (IN)	2.10	1.85	0.37	3.01	13.56	4.89	2.72	27.88
Ours (CIN)	**1.70**	**1.73**	**0.32**	**2.55**	**11.91**	**4.33**	**2.44**	**27.04**

Table 1: The left part shows the prediction error when rollout timesteps is the same as training time. The right part shows the generalization ability to longer horizon unseen during training. The error is scaled by 1,000. Our method has significantly improvements on all of the datasets.

method	PHYRE-C	SS-4	SimB-5
VIN	N.A.	N.A.	N.A.
OM	53.33	17.02	59.70
CVP	101.03	16.88	113.39
Ours (IN)	11.17	15.02	24.42
Ours (CIN)	**9.85**	**14.41**	**22.38**

Table 2: The ability to generalize to novel environments. We show the average prediction error for $t \in [0, 2 \times T_{\text{train}}]$. The error is scaled by 1,000.

	Within	Cross
RAND	13.7$\pm$0.5	13.0$\pm$5.0
DQN	77.6$\pm$1.1	36.8$\pm$9.7
Ours	**82.5**$\pm$1.1	**40.9**$\pm$11.6

Table 3: PHYRE Planning results. RAND stands for a score function with random policy. We show that our method achieves state-of-the-art on both within-task generalization as well as cross-task generalization.

	Target State Error	Hitting Accuracy
RAND	36.91	9.50%
CVP	29.84	20.3%
VIN	9.11	51.2%
OM	8.75	54.5%
Ours	**7.62**	**57.2%**

Table 4: Simulation Billiards planning results. To make a fair comparison, all of baselines and our methods are using the original interaction network. RAND stands for a policy taking random actions.

by more than 30% percent. In SimB, the improvement is not as significant because there is no environment information needed to infer future dynamics (see Figure 1).

3.2 Does learned model generalize to novel environments?

As one of the benefits, our method can generalize to novel environments configurations without any modifications or online training, thanks to the effective object-centric representations. We test such a claim by testing on several novel environments unseen during training. Specifically, we construct 1) simulation billiard dataset contains 5 balls with radius 2 (SimB-5); 2) PHYRE-C where the test tasks are not seen during training; 3) ShapeStacks with 4 stacked blocks (SS-4). The results are shown in Table 2. In the SimB-5 and PHYRE-C setting, where generalization ability to different numbers and appearances is required, our method reduce the prediction error by 75%.

3.3 How well can the learned model be used for planning actions?

The advantage of using a general purpose task-independent prediction model is that it can be used to do downstream planning tasks without any adaptation. We evaluate our prediction model in PHYRE benchmark and simulation billiards planning tasks. The full planning algorithm and implementation details are included in the appendix.

PHYRE. In this task, we need to place a red ball to solve a specific goal for each environment. The action space contains the position and size of the red ball. We use the first $10,000$ actions from pre-computed actions as our candidate sets and predict future object locations and masks for each action. After that, we score each action based on the predicted trajectory based on a trained classifier. We report the AUCCESS metric on the official 10 folds of train/test splits in Table 3. Our method achieves 1 points improvement over the strong DQN baseline [1] in within task generalization. On the other hand, our method is 4.7 points higher than DQN.

SimB Planning. We consider two tasks in the SimB environment: 1) *Billiard Target State.* Given an initial and final configuration after 40 timesteps, the goal is to find one action that will lead to the target configuration. We report the smallest distances between the trajectory between timestep 35-45 and the final position. 2) *Billiard Hitting.* Given the initial configurations, the goal is to find an action that can hit the other two balls within 50 timesteps.

4 Conclusions

In this paper, we leverage the modern computer vision techniques to propose *Region Proposal Interaction Networks* for physical interaction reasoning with visual inputs. We show that our general, yet simple method achieves a significant improvement and can generalize across both simulation and real-world environments for long-range prediction and planning. We believe this method may serve as a good benchmark for developing future methods in the field of learning intuitive physics, as well as their application to real-world robotics.

References

[1] A. Bakhtin, L. van der Maaten, J. Johnson, L. Gustafson, and R. Girshick. Phyre: A new benchmark for physical reasoning. *arXiv*, 2019. 1, 3, 4

[2] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. Interaction networks for learning about objects, relations and physics. In *NeurIPS*, 2016. 1, 2

[3] M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum. A compositional object-based approach to learning physical dynamics. *ICLR*, 2016. 1

[4] K. J. W. Craik. *The nature of explanation*. CUP Archive, 1952. 1

[5] E. Denton and R. Fergus. Stochastic video generation with a learned prior. In *ICML*, 2018. 1

[6] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine. Visual foresight: Model-based deep reinforcement learning for vision-based robotic control. *arXiv*, 2018. 1

[7] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 2

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. 2

[9] M. Janner, S. Levine, W. T. Freeman, J. B. Tenenbaum, C. Finn, and J. Wu. Reasoning about physical interactions with object-oriented prediction and planning. *ICLR*, 2019. 1, 3

[10] D. Jayaraman, F. Ebert, A. A. Efros, and S. Levine. Time-agnostic prediction: Predicting predictable video frames. *arXiv*, 2018. 1

[11] T. Kipf, E. van der Pol, and M. Welling. Contrastive learning of structured world models. *ICLR*, 2020. 1, 3

[12] J. R. Kubricht, K. J. Holyoak, and H. Lu. Intuitive physics: Current research and controversies. *Trends in cognitive sciences*, 2017. 1

[13] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv*, 2018. 1

[14] M. McCloskey. Intuitive physics. *Scientific american*, 1983. 1

[15] M. McCloskey, A. Washburn, and L. Felch. Intuitive physics: the straight-down belief and its origin. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 1983. 1

[16] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia. Learning to simulate complex physics with graph networks. *arXiv*, 2020. 1

[17] R. Veerapaneni, J. D. Co-Reyes, M. Chang, M. Janner, C. Finn, J. Wu, J. B. Tenenbaum, and S. Levine. Entity abstraction in visual model-based reinforcement learning. In *CoRL*, 2019. 3

[18] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016. 1

[19] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti. Visual interaction networks: Learning a physics simulator from video. In *NeurIPS*, 2017. 2, 3

[20] J. Wu, E. Lu, P. Kohli, B. Freeman, and J. Tenenbaum. Learning to see physics via visual de-animation. In *NeurIPS*, 2017. 3

[21] Y. Ye, M. Singh, A. Gupta, and S. Tulsiani. Compositional video prediction. *ICCV*, 2019. 2, 3

[22] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba, and J. B. Tenenbaum. Clevrer: Collision events for video representation and reasoning. In *ICLR*, 2020. 3