LEARNING VISUAL PREDICTIVE MODELS OF PHYSICS FOR PLAYING BILLIARDS

Katerina Fragkiadaki* **Pulkit Agrawal*** **Sergey Levine** **Jitendra Malik**
Electrical Engineering and Computer Science
University of California, Berkeley
(katef,pulkitag,svlevine,malik)@berkeley.edu

ABSTRACT

The ability to plan and execute goal specific actions in varied, unexpected settings is a central requirement of intelligent agents. In this paper, we explore how an agent can be equipped with an internal model of the dynamics of the external world, and how it can use this model to plan novel actions by running multiple internal simulations ("visual imagination"). Our models directly process raw visual input, and use a novel object-centric prediction formulation based on visual glimpses centered on objects (fixations) to enforce translational invariance of the learned physical laws. The agent gathers training data through random interaction with a collection of different environments, and the resulting model can then be used to plan goal-directed actions in novel environments that the agent has not seen before. We demonstrate that our agent can accurately plan actions for playing a simulated billiards game, which requires pushing a ball into a target position or into collision with another ball.

1 INTRODUCTION

Imagine a hypothetical person who has never encountered the game of billiards. While this person may not be very adept at playing the game, he would still be capable of inferring the direction in which the cue ball needs to be hit to displace the target ball to a desired location. How can this person make such an inference without any prior billiards-specific experience? One explanation is that humans are aware of the laws of physics, and a strategy for playing billiards can be inferred from knowledge about dynamics of bouncing objects. However, humans do not appear to consciously solve Newton's equations of motion, but rather have an intuitive understanding of how their actions affect the world. In the specific example of billiards, humans can imagine the trajectory that the ball would follow when a force is applied, and how the trajectory of ball would change when it hits the side of the billiards table or another ball. We term models that can enable the agents to visually anticipate the future states of the world as visual predictive models of physics.

A visual predictive model of physics equips an agent with the ability to generate potential future states of the world in response to an action without actually performing that action ("visual imagination"). Such visual imagination can be thought of as running an internal simulation of the external world. By running multiple internal simulations to imagine the effects of different actions, the agent can perform planning, choosing the action with the best outcome and executing it in the real world. The idea of using internal models for planning actions is well known in the control literature (Mayne, 2014). However, the question of how such models can be learned from raw visual input has received comparatively little attention, particularly in situations where the external world can change significantly, requiring generalization to a variety of environments and situations.

Previous methods have addressed the question of learning models, including visual models, of the agent's own body (Watter et al., 2015; Lillicrap et al., 2015). However, when performing actions in complex environments, models of both the agent and the external world are required. The external world can exhibit considerably more variation than the agent itself, and therefore such models must generalize more broadly. However, it is precisely the external world that must be modeled in detail to determine the actions that an agent must perform to achieve a goal state in novel conditions.

*equal contribution

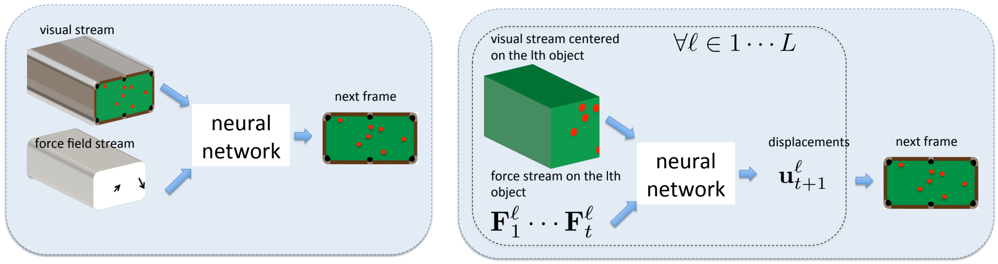

Figure 1: **Frame-centric versus object-centric prediction.** *Left:* Frame-centric model predicts takes as input the the image of the the entire billiards and forces applied by the agent to make predictions about the future. *Right:* Object-centric model predicts the future states of the world by individually modeling the temporal evolution of each the L objects in the world. In the billiards world, predicting the future velocities $(u_{t+h}^l, h \in [1, 20], l \in [1, L])$ of each of the L balls is sufficient for generating the future world states. Please see section 3 for more details.

The complexities associated with modeling the external world may be elucidated through an example. Consider the family of worlds composed of moving balls on a 2D table (i.e. *moving-ball* world). This family contains diverse worlds that can be generated by varying factors such as the number of balls, the table geometries, ball sizes, the colors of balls and walls, and the forces applied to push the balls. Because the number of objects can change across worlds, it is not possible to explicitly define a single state space for these worlds. For the purpose of modeling, an implicit state space must be learnt directly from visual inputs. In addition to this combinatorial structure, differences in geometry and nonlinear phenomena such as collisions result in considerable complexity.

Similar to the real world, in the *moving-ball* world, an agent must perform actions in novel conditions it has never encountered before. Although *moving-ball* worlds are relatively constrained and synthetic, the diversity of such worlds can be manipulated using a small number of factors. This makes them a good case study for systematically evaluating and comparing the performance of different model-based action selection methods under variation in external conditions (i.e. generalization).

Both the real world and the synthetic *moving-ball* worlds also contain regularities that allow learning generalizable models in the face of extensive variation, such as the translational invariance of physical laws. The main contribution of this work is a first step towards learning dynamical model of the external world directly from visual inputs that can handle combinatorial structure and exploits translation invariance in dynamics. We propose an object-centric (OC) prediction approach, illustrated in Figure 1), that predicts the future states of the world by individually modeling the temporal evolution of each object from object-centric glimpses. The object-centric (OC) approach naturally incorporates translation invariance and model sharing across different worlds and object instances.

We use a simulated billiards-playing domain as a working example to evaluate our approach, where the agent can push balls on a 2D billiard table with varying geometry. We show that our proposed method can learn a model of the billiards world that can effectively perform simulations of billiards games. We demonstrate this by using our model to predict the forces that are required to displace the ball to a desired location on the billiards table and to hit another moving ball. We also experimentally show that the object-centric (OC) prediction method generalizes better than the more standard frame-centric (FC) approach.

2 Previous Work

The idea that the basic physical and geometric constraints of our world play a crucial role in visual perception goes back at least to Helmholtz and his argument for "unconscious inference." This idea has been explored as a model in both control and computer vision.

Vision-based control Due to the recent success of deep neural networks for learning feature representations that can handle the complexity of visual input (Krizhevsky et al., 2012), there has been considerable interest in utilizing this capability for learning to control dynamical systems directly from visual input. Methods that directly learn policies that map from visual input to actions have been successfully used to learn to play Atari games (Mnih et al., 2013) and control a real robot for a variety of manipulation tasks (Levine et al., 2015). However, such methods do not attempt to model how the visual observations will evolve in response to the actions, making it difficult to repurpose the learned policies for new tasks. More recently works such as (Wahlström et al., 2015; Watter et al., 2015) have shown successful results on relatively simple domains of manipulating a synthetic two degree of freedom robotic arm or controlling an inverted pendulum in simulation. However, training and testing environments in these works were exactly the same. In contrast, our work shows that vision based model predictive control can be used in scenarios where the test environments are substantially different from training environments.

Another body of work has sought to learn useful features from raw sensory observations using methods such as deep autoencoders, in the hopes that these unsupervised learning techniques would produce a low-dimensional space that is better suited for reinforcement learning and control (Kietzmann & Riedmiller, 2009; Lange et al., 2012).

Models of physics and model based control Works such as (Jordan & Rumelhart, 1992; Wolpert et al., 1995; Haruno et al., 2001; Todorov & Ghahramani, 2003) proposed using internal models of the external world for planning actions. However these works have either been theoretical in nature or have striven to explain sensorimotor learning in humans. The work of (Hamrick et al., 2011) provided evidence that human judgement of how dynamical systems evolve in future can be explained by the hypothesis that humans use internal models of physics. To the best of our knowledge we are the first work that strives to build an internal model of the external world purely from visual data and use it for planning novel actions. (Oh et al., 2015) successfully predict future frames in Atari game videos and train a Q-controller for playing Atari games using the predicted frames. Training a Q-controller requires supervision for training. Our goal in this work is to show that models learnt directly from visual inputs can be used for planning actions without requiring any task specific supervision. Secondly, although Atari games represent more complex environments than billiards, the complexity of Atari game environments remain fixed. On the other hand, in our setup the complexity of environments is variable which is more representative of real world tasks. Finally, we train our predictive models using partial observation of the world, whereas in (Oh et al., 2015) the world is assumed to be fully observable.

Learning Physics from Images and Videos Works of Wu et al. (2015); Bhat et al. (2002); Brubaker et al. (2009); Mottaghi et al. (2015) propose methods for estimating the parameters of Newtonian equations from images and videos. This work proposes learning Physics through learning predictive visual models of the world, without using priors of Newtonian Equations of motion. As laws of physics governing the dynamics of balls and walls on a billiards table are well understood, it is possible to use these laws instead of learning a predictive model for planning actions. However, there are different dynamic models that control ball-ball collisions, ball-wall collisions and the movement of ball in free space. Therefore, if these known dynamical model are to be used, then a system for detecting different event types would be required for selecting the appropriate dynamics model at different time points. We instead propose a model to precisely discover events that are relevant for motion prediction directly from the visual input, without using priors regarding Newtonian Equations, a solution much more general and tractable with increasing complexity of the environment.

Video prediction Works of (Michalski et al., 2014; Sutskever et al., 2008) generate images of bouncing balls using a hierarchy of learnt image transformations in (Michalski et al., 2014), and Recurrent Temporal Restricted Boltzmann Machines in (Sutskever et al., 2008). Both models are trained for pixel reconstruction of the next frame and regress to grayscale pixel values. While these methods demonstrate accurate predictions, they do not consider generalization to different environments. Our model predicts object transformations instead of video pixel values, exploiting object persistency in the world. Such object-centric predictive models can handle combinatorial structure in the scene and operate on a more abstract visual representation, which improves generalization. In our comparisons, we show that this approach significantly outperforms more standard "full frame" models.

In (Boots et al., 2014), the authors propose to learn a predictive model for the visual appearance of a robotic arm, but the applications are limited to instances of the same object in the same visual context, and the non-parametric approach is restricted in its ability to learn from large datasets. In contrast, parametric models such as those used in this work scale readily to very large training sets. Our experiments demonstrate generalization to variation in the environment and the ability of our model to handle large and variable numbers of objects.

Motion prediction for Visual Tracking In Computer Vision, object trackers use a wide variety of predictive models, from simple constant velocity models, to linear dynamical systems and their variants (Urtasun et al., 2006), HMMs (Brand et al., 1997; Ghahramani & Jordan, 1997), and other models. Standard smoothers or filters, such as Kalman filters (Weng et al., 2006), usually operate on Cartesian coordinates, rather than the visual content of the targets, and in this way discard useful information that may be present in the images. Finally, methods for 3D tracking of Kyriazis et al. (2011); Salzmann & Urtasun (2011) use Physics simulators to constrain the search space during data association.

3 Learning Predictive Visual Models

We consider an agent observing an interacting with dynamic *moving-ball* worlds consisting of multiple balls and multiple walls. We also refer to these worlds as billiard worlds. The agent interacts with the world by applying forces to change the velocities of the balls. In the real world, the environment of an agent is not fixed, and the agent can find itself in environments that it has not seen before. To explore this kind of generalization, we train our predictive model in a variety of billiards environments, which involve different numbers of balls and different wall geometries, and then test the learnt model in previously unseen settings.

In the case of *moving-ball* world, it is sufficient to predict the displacement of the ball during the next time step to generate the visual of the world in the future. Therefore, instead of directly predicting image pixels, we predict each object's current and future velocity given a sequence of visual glimpses centered at the object (visual fixation) and the forces applied to it.

We assume that during training the agent can perfectly track the objects. This assumption is a mild one because not only tracking is a well studied problem but also because there is evidence in the child development literature that very young infants can redirect their gaze to keep an object in focus by anticipating its motion (i.e. smooth pursuit) (Hofsten & Rosander, 1997). The early development of smooth pursuit suggests that it is important for further development of visual and motor faculties of a human infant.

3.1 Model Details

Our network architecture is illustrated in figure 2. The input to the model is a stack of 4 images comprised of the current and previous 3 glimpses of the fixated object and the exercised force on the object at the current time step. The model predicts the velocity of the object at each of the h time steps in the future. We use $h = 20$. The same model is applied to all the objects in the world.

Our network uses an AlexNet style architecture Krizhevsky et al. (2012) to extract visual features. The first layer (conv1) is adapted to process a stack of 4 frames. Layers 2 and 3 have the same architecture as that of AlexNet. Layer 4 (conv4) is composed of 128 convolution kernels of size 3×3. The output of conv4 is rescaled to match the value range of the applied forces, then is concatenated with the current force and is passed into a fully connected (encoder) layer. Two layers of LSTM units operate on the output of the encoder to model long range temporal dynamics. Then, the output is decoded to predicted velocities.

The model is trained by minimizing the Euclidean loss between ground-truth and predicted object velocities for h time steps in the future. The ground-truth velocities are known because we assume object tracking. The loss is mathematically expressed as:

$$\mathcal{L} = \sum_{k=1}^{h} w_k \|\tilde{\mathbf{u}}_{t+k} - \mathbf{u}_{t+k}\|_2^2 \tag{1}$$

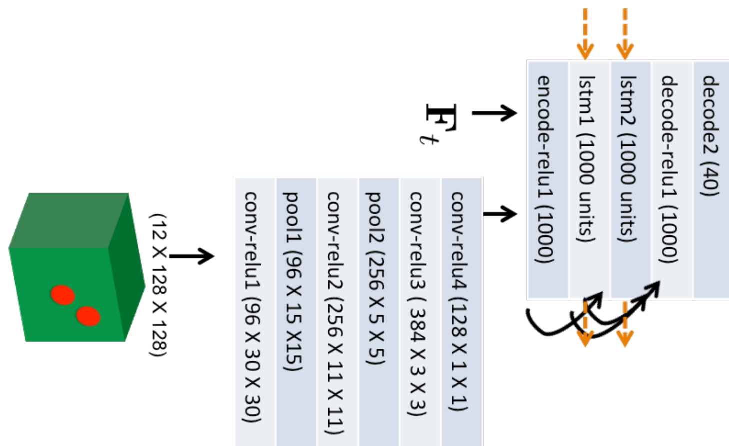

Figure 2: **Network architecture.** At each time step t, for each object, the network is provided with the previous four glimpses centered on the object's position, as well as the agent's applied forces $\mathbf{F}_t = (F_t^x, F_t^y)$ and the hidden states of the LSTM units from the previous time step. The output is ball displacements $\mathbf{u}_{t+k} = (\Delta x_{t+k}, \Delta y_{t+k})$ for $k = 1 \cdots h$ in the next h frames.

where $\mathbf{u}_{t+k}, \tilde{\mathbf{u}}_{t+k}$ represent ground-truth and predicted velocities at the k^{th} time step in the future respectively. Velocities are expressed in cartesian coordinates. We weigh the loss in a manner that errors in predictions at a shorter time horizon are penalized more than predictions at a longer time horizon. This weighting is achieved using penalty weights $w_k = \exp(-k^{\frac{1}{4}})$. We use the publicly available Caffe package for training our model.

For model learning, we generate sequences of ball motions in a randomly sampled world configuration. As shown in Figure 3, we experimented both with rectangular and non-rectangular wall geometries. For rectangular walls, a single sample of the world was generated by randomly choosing the size of the walls, location of the balls and the forces applied on the balls from a predefined range. The length of each sequence was sampled from the range [20, 200]. The length of the walls was sampled from a range of [300 pixels, 550 pixels]. Balls were of radius 25 pixels and uniform density. Force direction was uniformly sampled and the force magnitude was sampled from the range [30K Newtons, 80K Newtons]. Forces were only applied on the first frame. The size of visual glimpses is 600x600 pixels. The objects can move up to 10 pixels in each time step and therefore in 20 time steps they can cover distances up to 200 pixels.

For training, we pre-generated 10K such sequences. We constructed minibatches by choosing 50 random subsets of 20 consequent frames from this pre-generated dataset. Weights in layers conv2 and conv3 were initialized from the weights of Alexnet that was trained for performing image classification on Imagenet (Krizhevsky et al., 2012). Weights in other layers are randomly initialized.

4 Model Evaluation

We have found significant performance gains of the 4 frame LSTM model over a standard per frame LSTM model. Supplying 4 concatenated frames instead of one reduces the angular velocity error from 8.21 degrees to 5.65 degrees for single ball predictions. This suggests spatio-temporal convolutions are more suitable for making useful information available regarding estimating current and future velocities over per frame convolutions consolidated by a later recurrent layer. The benefit of the LSTM memory layer is discussed in figure 7: in very large environments where glimpses are not informative for more than two frames, long term memory is necessary to supply the correct velocity computed under the last ball-ball or ball-wall collision. When glimpses always anchor to wall surroundings, the recurrent model performs comparably to the non-recurrent one. Thus, our results presented next concern the non-recurrent 4 frame concatenation model.

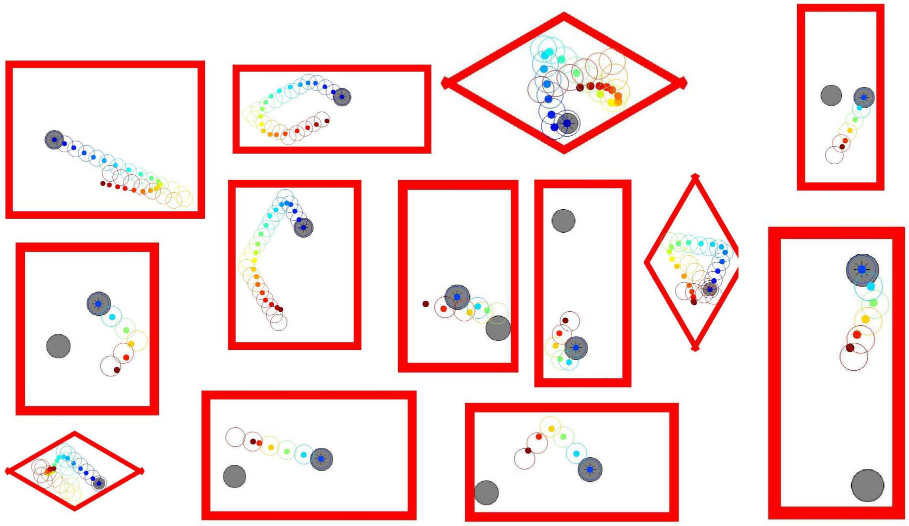

Figure 3: **Predicting ball motion under interactions and agent forces.** Dots denote predictions of our model and circles the ground-truth future positions for the fixated ball. Color indicates length of the prediction horizon, blue denotes close and red far in the future. Our model predicts collisions early and accurately under a wide variety of table configurations.

Time	Overall Error			Error Near Collisions		
	CV	FC	OC	CV	FC	OC
t+1	3.0^o/0.00	6.2^o/0.04	$5.1^o/0.03$	23.2^o/0.00	11.4^o/0.06	9.8^o/0.04
t+5	11.8^o/0.01	8.7^o/0.05	$7.2^o/0.04$	56.6^o/0.05	21.1^o/0.12	17.9^o/0.10
t+20	45.3^o/0.01	16.3^o/0.09	$14.8^o/0.09$	123.0^o/0.04	54.8^o/0.20	54.8^o/0.20

Table 1: Quantitative evaluation of the prediction model reported as error in the magnitude and angle of the velocity. We report the average error across all frames (i.e. Overall Error) and error averaged only in frames that were within [-4, 4] frames of a frame depicting collision (i.e. Error Near Collisions). Errors are reported for a constant velocity (CV) model, Frame Centric (FC) and Object Centric (OC) models. The errors are reported as a^o/b, where a is the mean of angular error in degrees and b is the relative error in the magnitude of the predicted velocity. The constant velocity model predicts the velocity of a ball for all the future frames to be same as the ground truth velocity at the previous time step.

First we report evaluations on random worlds sampled from the same distribution as the training data. Next, we report evaluations on worlds sampled from a different distribution of world configurations to study the generalization of the proposed approach. Error in the angle and magnitude of the predicted velocities were used as performance metrics. We compared the performance of the proposed object centric (OC) model with a constant velocity (CV) and frame centric (FC) model. The constant velocity model predicts the velocity of a ball for all the future frames to be same as the ground truth velocity at the previous time step. The ball changes the velocity only when it strikes another ball or hits a wall. As collisions are relatively infrequent, the constant velocity model is a good baseline.

We first trained a model on the family of rectangular worlds consisting of 1 ball only. The results of this evaluation are reported in Table 1. We used average error across all the frames and the error averaged across frames only near the collisions as the error metrics for measuring performance. As balls move in linear trajectories except for time of collision, accurately predicting the velocities after a collision event is of specific interest. Results in Table 1 show that the object centric (OC) model is better than frame centric model (FC) model and much better than the constant velocity model. These results show that object centric modelling leads to better learning.

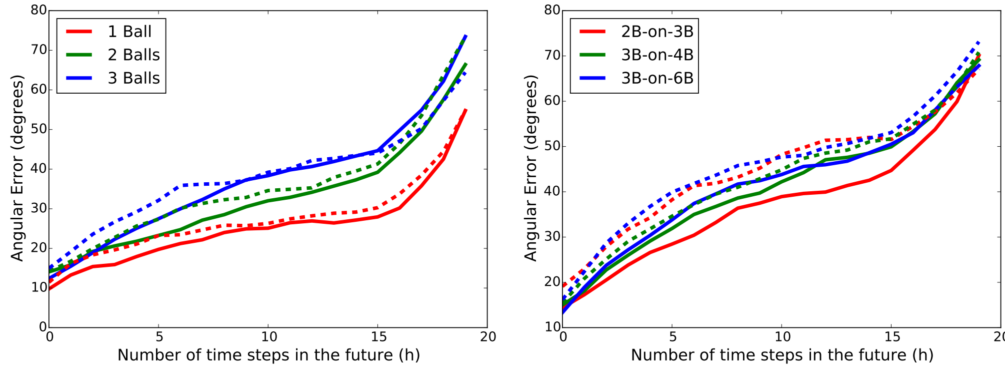

Figure 4: Performance comparison of frame centric (FC) and object centric (OC) models. Performance is measured as the angular error near collisions. Dashed and solid lines show the angular error over a horizon of 20 time steps ($h = 20$) for FC and OC models respectively. (*Left*) Angular errors for models trained and tested on 1, 2 and 3 ball worlds respectively. The performance of the models only degrades slightly with increasing number of balls. Generally, the OC model is more accurate than the FC model. (*Right*) Comparing the performance the FC model against the OC model when models trained on 2 and 3 balls are tested on worlds containing a larger number of balls. The naming convention nB-on-mB indicates models trained on n-ball worlds and tested on m-ball worlds. The generalization of the proposed OC model is significantly better than the FC model.

How well does our model scale with increasing number of balls? For studying this, we trained models on families of world consisting of 2 and 3 balls respectively. We used the learnt 1-ball model to initialize the training of a model with 2 balls. Then we used the 2-ball model to initialize the training of the 3-ball model. We found that this curriculum learning approach led to better results than training models from scratch. We then evaluated the 2 and 3 ball models on worlds separate from training set that consisted of 2 and 3 ball respectively. The angular errors measured near collisions (for $h = 1$ to 20) are shown Figure 4. The results show that the performance of our model degrades only by a small amounts as the number of balls increase. Also, in general the OC model performs better than the FC model.

We also trained and tested our models on non-rectangular walls. Qualitative visualizations of ground truth and predicted ball trajectories are show in figure 3. The figure shows that our model accurately predicts the velocities of balls after collisions in varied environments. This result indicates that our models are not constrained to any specific environment and have learnt something about the dynamics of balls and their collisions.

4.1 Evaluating Generalization

The results reported in the previous section show generalization to worlds sampled from the same distribution as the training set. In addition to this, we also tested our models on worlds substantially different from the worlds in the training set.

Figure 7 shows that our model can generalize to much larger wall configurations than those used in the training. The wall lengths in the training set were between 300-550 pixels, whereas the the wall lengths in the testing set were sampled from the range of 800-1200 pixels. This shows that our models can generalize to different wall geometries.

Figure 4 (*right*) shows that models trained on 2 and 3-ball worlds perform well when tested on 3, 4 and 6-ball worlds. This shows that our models can generalize to worlds with larger number of balls without requiring any additional training. The results in the figure also show that proposed OC model generalizes substantially better than the FC model.

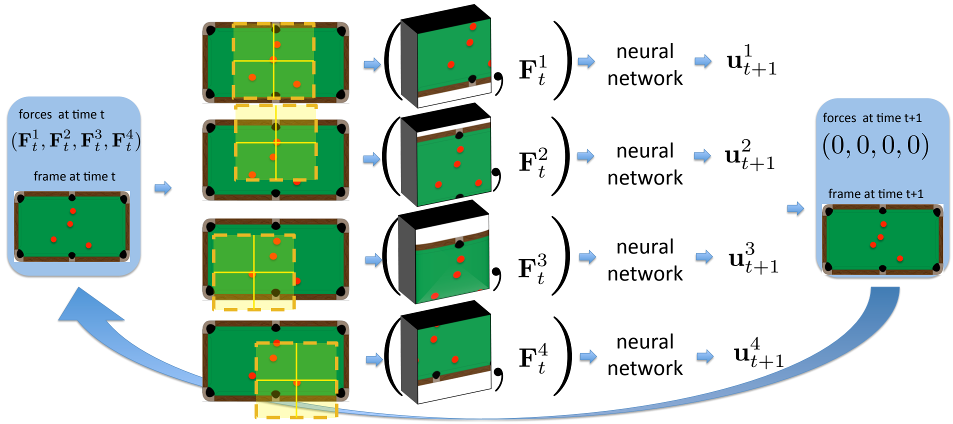

Figure 5: **Generating visual imaginations:** Given a configuration of the world and applied forces, the visual image of the world at the next time step is generated by first predicting the future velocities of every ball. The visual image of the world at the next time step is rendered by translating the balls by their predicted velocities. This process is repeated iteratively to generate a sequence of visuals of the future world states.

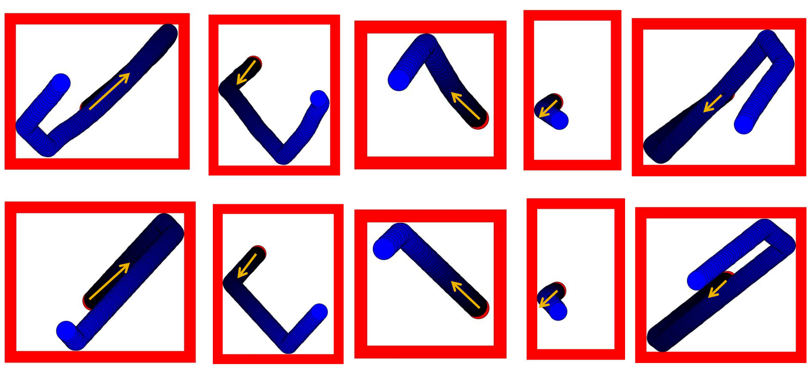

Figure 6: Visual Imaginations generated by our model. (Top) shows the trajectory of the ball generated by our model by iteratively rendering the next world image using the predicted ball velocities and feeding this rendered image as input to the model. The imagined ball trajectory is color coded to reflect the progression of time. Change in color from dark to light blue indicates moving forward in time. (Bottom) shows the respective ground truth trajectories. Force on the ball is applied at the first time step and the force vector is shown by the orange arrow. The figure shows that our model learns the dynamics of balls and collisions and is successfully able to image collision events.

5 Generating Visual Imaginations

As our models can accurately predict ball dynamics in the future, we can use these models to generate visual imaginations of the future. The procedure we use for generating visual imaginations is illustrated in figure 5. Given a configuration of the world and applied forces, the visual image of the world at the time step $t + 1$ is generated by translating each ball by amount to equal to its predicted velocity ($\tilde{u}_t$) at time t. This generated image forms the input to the model for generating the visual image of the world at time step $t+2$. This process is repeated iteratively to generate visual imaginations of the future world states.

Some examples of visual imaginations by our model are shown in figure 6. Our model learns to anticipate collisions early in time. Predicting trajectory reversals at collisions is not possible using methods Kalman filter based methods that are extensively used in object tracking(Welch & Bishop, 1995). Comparison with ground truth trajectories reveals that our models are not perfect and in some

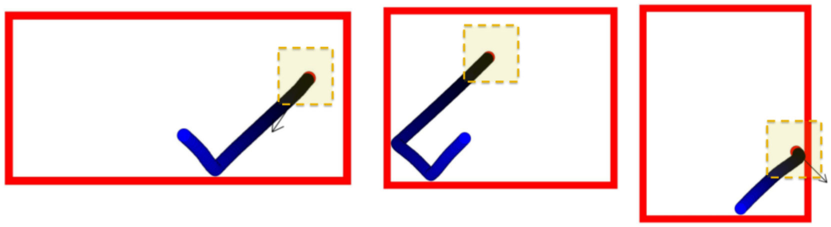

Figure 7: Visual imaginations of our model in very large environments. Accurate predictions of the ball trajectory are made despite the fact that visual glimpses (shown in yellow) are mostly uninformative (they only contain a ball in center of white background). This is made possible by the long term memory in LSTMs. Without LSTMs, the ball exhibits non natural motion and reverses its direction in the middle of the arena.

cases accumulation of errors can produce imagined trajectories that are different from the ground truth trajectories (for instance see the first column in figure 6). Even in the cases when predicted trajectories do not exactly match up with the ground truth trajectories, the visual imaginations are consistent with the dynamics of balls and collisions.

Figure 7 shows visual imaginations by our model in environments that are much larger than the environments used in the training set. Notice that the glimpse size is considerably smaller than the size of the environment. With glimpses of this size, visual inputs when the ball is not close to any of the walls are uninformative because such visual inputs merely comprise of a ball present in center of white background. In such scenarios, our model is able to make accurate predictions of velocity due to the long-range LSTM memory of the past. Without LSTM units, we noticed that imagined ball trajectory exhibited unexpected reversal in directions and other errors. For more examples, please see accompanying video for imaginations in two and three ball worlds.

6 Using Predictive Visual Models for Action Planning

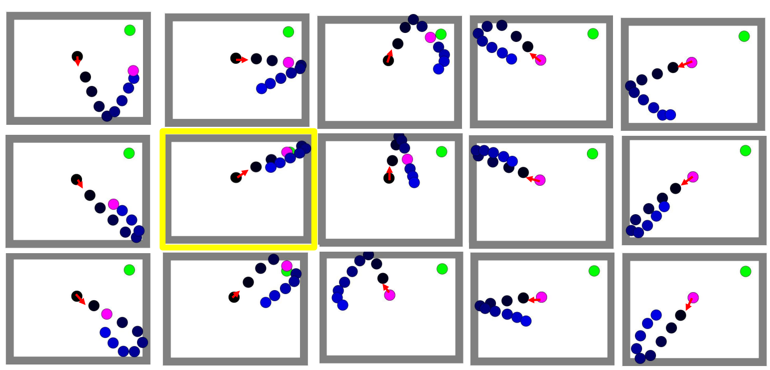

Figure 8: Illustration of the method for planning novel actions. In the task shown in the figure, the agent needs to decide the force with which it needs to hit the cue ball (the ball with the arrow) to a target location (shown in green). The agent runs multiple simulations of the world (i.e. visual imagination) by applying different forces (shown by red arrows). Each grey box shows the results of simulations performed by the agent. The position of the ball closest to the target location is shown in pink in each of the simulations. The agent chooses the force that leads to a ball location that is closest to the target location (simulation in yellow box).

Results in previous sections show that our agent is able to learn a model of the external world and can successfully predict the future states of the external world under the action of forces applied by the agent. These models can be used for planning actions for which the agent has never received any

Method	Hit Accuracy		
	$<$ 10 pixels	$<$ 25 pixels	$<$ 50 pixels
Oracle	95%	100%	100%
Random	3%	14%	23%
Ours (FC-Model)	15%	39%	60%
Ours (OC-Model)	30%	56%	85%

Table 2: Hit accuracy of our approach for pushing the ball to a desired target location. The hit accuracy was measured as the number of trials for which the closest point on the ball's trajectory was within p pixels of the target. Accuracy has been reported for p={10, 25, 50} pixels. The random baseline was constructed by randomly choosing a force and the oracle used the ground truth physics simulator for selecting the optimal actions.

direct supervision. We first show results on a relatively simple task of planning the force required to push the desired ball to a desired location. Next, we show results on a more challenging task of planning the force required to push the desired ball to hit a second moving ball.

Figure 8 illustrates the method of action planning. Given a target state, the optimal force is found by running multiple simulations (i.e. visual imaginations) of the world after applying different forces. The optimal force is the one that produces the world state that is closest to the target state. In order to verify the accuracy of this method, we use the predicted force from our model as input to our physics engine to generate its actual (rather than the imagined) outcome and compare the resulting states against the goal states. In practice, instead of exhaustively searching for all forces we use CMA-ES method (Hansen & Ostermeier, 2001) for determining the optimal force.

Table 2 reports the hit accuracy of our system in pushing the ball to a desired location. The hit accuracy was measured as the number of trials for which the closest point on the ball's trajectory was within p pixels of the target. With an accuracy of 56% our model is able to push the ball within 25 pixels (the size of the arena was between 300-550 pixels in size) of the target location as compared to the oracle which is successful 100% times. The oracle was constructed by using the ground truth physics simulator for making predictions and used the same mechanism for action selection as described above. Qualitative results of our methods are best seen in the accompanying video. We will include quantitative evaluation of more complex actions in the next revision of the paper.

7 Discussion and Conclusion

We have presented an object-centric prediction approach that exploits translation invariance in dynamics of physical systems to learn a dynamical model of the world directly from visual inputs. We show that the model generalizes to environments never encountered during training and can be used for planning actions in novel environments without the requirement of task-specific supervision.

Currently our prediction models are not perfect and we are pursuing multiple approaches to address this. Furthermore, scaling our method to real world settings would require a more complex mechanism for creating visual renderings. We are investigating directions that recurrently carry out imaginations in a latent abstract feature space and using visual exemplar images as proxies instead of synthesizing images by rendering individual pixels.

References

Bhat, Kiran S., Seitz, Steven M., and Popovic, Jovan. Computing the physical parameters of rigid-body motion from video. In Heyden, Anders, Sparr, Gunnar, Nielsen, Mads, and Johansen, Peter (eds.), *ECCV (1)*, volume 2350 of *Lecture Notes in Computer Science*, pp. 551–565. Springer, 2002. ISBN 3-540-43745-2. URL http://dblp.uni-trier.de/db/conf/eccv/eccv2002-1.html#BhatSP02.

Boots, Byron, Byravan, Arunkumar, and Fox, Dieter. Learning predictive models of a depth camera & manipulator from raw execution traces. In *2014 IEEE International Conference on Robotics* *and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, pp. 4021–4028, 2014. doi: 10.1109/ICRA.2014.6907443. URL http://dx.doi.org/10.1109/ICRA.2014.6907443.

Brand, M., Oliver, N., and Pentland, A. Coupled hidden markov models for complex action recognition. In *CVPR*, 1997.

Brubaker, Marcus A., Sigal, L., and Fleet, D.J. Estimating contact dynamics. In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2389–2396, Sept 2009. doi: 10.1109/ICCV.2009.5459407.

Ghahramani, Zoubin and Jordan, Michael I. Factorial hidden markov models. *Mach. Learn.*, 29, 1997.

Hamrick, Jessica, Battaglia, Peter, and Tenenbaum, Joshua B. Internal physics models guide probabilistic judgments about object dynamics. In *Proceedings of the 33rd annual conference of the cognitive science society*, pp. 1545–1550. Cognitive Science Society Austin, TX, 2011.

Hansen, Nikolaus and Ostermeier, Andreas. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

Haruno, Masahiko, Wolpert, David H, and Kawato, Mitsuo. Mosaic model for sensorimotor learning and control. *Neural computation*, 13(10):2201–2220, 2001.

Hofsten, Claes Von and Rosander, Kerstin. Development of smooth pursuit tracking in young infants. *Vision Research*, 37(13):1799 – 1810, 1997. ISSN 0042-6989. doi: http://dx.doi.org/10.1016/S0042-6989(96)00332-X. URL http://www.sciencedirect.com/science/article/pii/S004269899600332X.

Jordan, Michael I and Rumelhart, David E. Forward models: Supervised learning with a distal teacher. *Cognitive science*, 16(3):307–354, 1992.

Kietzmann, Tim C. and Riedmiller, Martin A. The neuro slot car racer: Reinforcement learning in a real world setting. In *International Conference on Machine Learning and Applications, ICMLA 2009, Miami Beach, Florida, USA, December 13-15, 2009*, pp. 311–316, 2009. doi: 10.1109/ICMLA.2009.15. URL http://dx.doi.org/10.1109/ICMLA.2009.15.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012.

Kyriazis, Nikolaos, Oikonomidis, Iason, and Argyros, Antonis A. Binding computer vision to physics based simulation: The case study of a bouncing ball. In *British Machine Vision Conference, BMVC 2011, Dundee, UK, August 29 - September 2, 2011. Proceedings*, pp. 1–11, 2011. doi: 10.5244/C.25.43. URL http://dx.doi.org/10.5244/C.25.43.

Lange, Sascha, Riedmiller, Martin A., and Voigtländer, Arne. Autonomous reinforcement learning on raw visual input data in a real world application. In *The 2012 International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, June 10-15, 2012*, pp. 1–8, 2012. doi: 10.1109/IJCNN.2012.6252823. URL http://dx.doi.org/10.1109/IJCNN.2012.6252823.

Levine, Sergey, Finn, Chelsea, Darrell, Trevor, and Abbeel, Pieter. End-to-end training of deep visuomotor policies. *CoRR*, abs/1504.00702, 2015. URL http://arxiv.org/abs/1504.00702.

Lillicrap, Timothy P., Hunt, Jonathan J., Pritzel, Alexander, Heess, Nicolas, Erez, Tom, Tassa, Yuval, Silver, David, and Wierstra, Daan. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015.

Mayne, David Q. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967 – 2986, 2014. ISSN 0005-1098. doi: http://dx.doi.org/10.1016/j.automatica.2014.10.128. URL http://www.sciencedirect.com/science/article/pii/S0005109814005160.

Michalski, Vincent, Memisevic, Roland, and Konda, Kishore Reddy. Modeling deep temporal dependencies with recurrent grammar cells"". In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 1925–1933, 2014. URL http://papers.nips.cc/paper/5549-modeling-deep-temporal-dependencies-with-recurrent-grammar-cells.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Graves, Alex, Antonoglou, Ioannis, Wierstra, Daan, and Riedmiller, Martin A. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL http://arxiv.org/abs/1312.5602.

Mottaghi, Roozbeh, Bagherinezhad, Hessam, Rastegari, Mohammad, and Farhadi, Ali. Newtonian image understanding: Unfolding the dynamics of objects in static images. *CoRR*, abs/1511.04048, 2015. URL http://arxiv.org/abs/1511.04048.

Oh, Junhyuk, Guo, Xiaoxiao, Lee, Honglak, Lewis, Richard, and Singh, Satinder. Action-conditional video prediction using deep networks in atari games. *arXiv preprint arXiv:1507.08750*, 2015.

Salzmann, Mathieu and Urtasun, Raquel. Physically-based motion models for 3d tracking: A convex formulation. In Metaxas, Dimitris N., Quan, Long, Sanfeliu, Alberto, and Gool, Luc J. Van (eds.), *ICCV*, pp. 2064–2071. IEEE, 2011. ISBN 978-1-4577-1101-5. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2011.html#SalzmannU11.

Sutskever, Ilya, Hinton, Geoffrey E., and Taylor, Graham W. The recurrent temporal restricted boltzmann machine. In *NIPS*, 2008.

Todorov, Emanuel and Ghahramani, Zoubin. Unsupervised learning of sensory-motor primitives. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, volume 2, pp. 1750–1753. IEEE, 2003.

Urtasun, Raquel, Fleet, David J., and Fua, Pascal. 3d people tracking with gaussian process dynamical models. In *CVPR*, 2006.

Wahlström, Niklas, Schön, Thomas B., and Deisenroth, Marc Peter. From pixels to torques: Policy learning with deep dynamical models. *CoRR*, abs/1502.02251, 2015. URL http://arxiv.org/abs/1502.02251.

Watter, Manuel, Springenberg, Jost Tobias, Boedecker, Joschka, and Riedmiller, Martin A. Embed to control: A locally linear latent dynamics model for control from raw images. *CoRR*, abs/1506.07365, 2015. URL http://arxiv.org/abs/1506.07365.

Welch, Greg and Bishop, Gary. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

Weng, Shiuh-Ku, Kuo, Chung-Ming, and Tu, Shu-Kang. Video object tracking using adaptive kalman filter. *J. Vis. Comun. Image Represent.*, 17(6):1190–1208, 2006. ISSN 1047-3203. doi: 10.1016/j.jvcir.2006.03.004. URL http://dx.doi.org/10.1016/j.jvcir.2006.03.004.

Wolpert, Daniel M, Ghahramani, Zoubin, and Jordan, Michael I. An internal model for sensorimotor integration. *Science-AAAS-Weekly Paper Edition*, 269(5232):1880–1882, 1995.

Wu, Jiajun, Yildirim, Ilker, Lim, Joseph J, Freeman, Bill, and Tenenbaum, Josh. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 127–135. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5780-galileo-perceiving-physical-object-properties-by-integrating-a-physics-engine pdf.