GEOMETRY-AWARE VISUAL PREDICTIVE MODELS OF INTUITIVE PHYSICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning object dynamics for model-based control usually involves choosing among two alternatives: i) engineered 3D state representations comprised of 3D object locations and poses, or, ii) learnt 2D image representations trained end-to-end for the dynamics prediction task. The former requires laborious human annotations to extract the 3D information from 2D images, and does not permit end-to-end learning. The latter has not shown until today to generalize across camera viewpoints or to handle camera motion and cross-object occlusions. We propose neural architectures that learn to disentangle an RGB-D video steam into camera motion and 3D scene appearance, and capture the latter into 3D feature representations that can be trained end-to-end with 3D object detection and object motion forecasting. We feed object-centric 3D feature maps and actions of the agent into differentiable neural modules and learn to forecast object 3D motion. We empirically demonstrate the proposed 3D representations learn object dynamics that generalize across camera viewpoints and can handle object occlusions. They do not suffer from error accumulation when unrolled over time thanks to the permanence of object appearance in 3D. They outperform by a margin both 2D learned image representations as well as engineered 3D ones in forecasting object dynamics.

1 INTRODUCTION

For agents to control their environment. they need models that describe how the environment reacts to their actions and interactions (Miall & Wolpert, 1996; Haruno et al., 1999). Dynamic models have a vital role for the development of motor control (Miall & Wolpert, 1996) as they permit self-training by mentally simulating the results of the agent's actions. Such mental planning is especially important in unfamiliar environments, where expert reactive policies have not yet been acquired (M. Lake et al., 2016). Dynamics models encode cause-and-effect relationships during agent-object and object-object interactions, and describe the dynamics of world states given actions of the agent. World states are extracted from the sensory observation streams of the agent, e.g., visual RGB, or RGB-D streams. We call such learnt dynamic models *intuitive physics* as they operate on state representations and abstractions learnt by the agent, as opposed to mass, inertia, accurate 3D object shape, coefficient of friction and so on, required by Newtonian physics. Though Newtonian physics also describes dynamics of the world that can be used for robot control, yet, it suffers from under-modelling: the Newtonian system of equations would need to be terribly complex to describe sufficiently well picking up a mug in a cluttered environment. Instead, intuitive physics and learnt dynamics models extract from the sensory streams state information sufficient for solving the task, and find shortcuts to achieve this, bypassing Newtonian physics requirements. Indeed, we, humans, excel in manipulation despite that less than 1% of us know what inertia is. We can all effectively drink coffee from our coffee mug, despite that we do not have an accurate perception of the coffee particles and their displacements.

Many recent works have tried learning dynamics of visual state representations using deep neural networks. A central question then is: how do we represent the world state? We identify two main research threads:

i) Methods that **predict the future in a 2D projection space**, such as future visual frames (Mathieu et al., 2015; Oh et al., 2015a; Finn et al., 2016), CNN encodings of future frames (Ha &

Schmidhuber, 2018; Chiappa et al., 2017), or object 2D motion trajectories (Fragkiadaki et al., 2015; Battaglia et al., 2016; Chang et al., 2016). Such dynamics models have not been shown to generalize across environment variations—e.g. background clutter—or camera viewpoints. High capacity neural networks are used to learn dynamics of training environments of very limited variability, e.g., of a constant camera viewpoint. Moreover, works that use object factorization biases (Fragkiadaki et al., 2015; Battaglia et al., 2016; Chang et al., 2016) and share weights of the predictive model across objects present in the scene, either assume the objects' masks are given (Fragkiadaki et al., 2015; Battaglia et al., 2016; Chang et al., 2016) or trivial to obtain from color segmentation.

ii) Methods that **predict the future in a hand-designed 3D space** of object or particle locations and poses extracted from the RGB images using extensive human annotations (OpenAI et al., 2018) or instrumentation (OpenAI et al., 2018; Wu et al., 2017; Li et al., 2019; Mrowca et al., 2018). In 3D, objects move and deform under force applications during robot-object and object-object interactions, but rarely do they drastically change appearance. In contrast, 2D images— and their CNN activations as a result— change drastically over time due to occlusions and dis-occlusions, camera motion and changes of the camera field-of-view. As a result, a temporal change predictor has a lot more to account for in 2D than in 3D. Moreover, if we could detect the objects in 3D, it is trivial to reason about free space, predict object collisions and plan obstacle-avoiding trajectories in truly novel environments. Similar capabilities and reasoning would require many examples to learn directly from 2D RGB images and would have questionable generalization (OpenAI et al., 2018). Yet, explicitly engineered 3D representations such as 3D boxes and their arrangements cannot be learnt end-to-end for the end prediction task or under auxiliary objectives, since the engineer decides what to retain from the RGB input. Also, such models cannot be used to learn physics in the real world since these 3D state representations are hard in general to obtain from raw RGB input in-the-wild (OpenAI et al., 2018), outside multiview environments (Li et al., 2019).

We embrace the attractiveness of end-to-end dynamics learning from RGB or RGB-D streams of the approaches in the first research thread, and conjecture that their limited generalizability is due to the fact that **2D videos in fact violate basic rules of Newtonian Physics**: objects "shake" under camera motion, they change size as a result of zoom-in or zoom-out, and they appear and disappear as a result of camera or object motion. This is in turn due to the fact that the camera motion is entangled with objects' motion and appearance in video streams. Human brain performs effortlessly such disentanglement: we perceive a stable computer screen while moving our gaze around our desks. Thus, when we imagine how the screen will move when pushed, such prediction is carried out in the egomotion-stable model of the world we infer. This allows us to effortless generalize dynamics across camera viewpoints.

We propose learning models of intuitive physics in a latent egomotion-disentangled 3-dimensional feature space that our agent learns to infer from 2.5D video streams by predicting results of its (ego)motion and end-effectors' motion. We present neural architectures that estimate camera motion and disentangle it from the 3D feature appearance of the scene using explicit 3D feature transformations in their latent feature space (Figure 1 top). As a result, objects in the 3D latent feature space are *permanent*: they persist despite occlusions and dis-occlusions in the corresponding 2.5D video input. It is thus easy to detect and segment such objects in 3D using the egomotion-disentangled 3D feature map as input to a per-frame 3D object detector. Then, object-centric 3D feature maps are used as the input to a forecasting neural module that learns to predict an object's future 3D motion conditioned on a robot's action, as shown in Figure 1 bottom. We unroll the learned models forward in time and show we can effectively push objects to desired 3D locations. We show such forward unrolling benefits from the permanence of 3D feature representations, where object appearance does not vary much over time, rather, objects simply change locations and orientations according to the robot's actions.

We test the proposed architectures in perceiving objects in 3D and predicting their 3D motion resulting from pushing actions of a robot agent. We empirically show our model outperforms models that either do not take into account object appearance by using 3D object box centroids, ground-truth or estimated, or models that use 2D visual features obtained from 2D image views. Given completely new objects and arrangements, our model can detect the objects in 3D, find optimal paths, and choose the right actions to push the object to desired target locations given a single view as input. The proposed representations are end-to-end differentiable under both (supervised) motion forecasting and 3D object detection, as well as (self-supervised) view prediction, that aids their generalizability. In summary, our contributions are:

- Re-conciling 3D representations and end-to-end learning for model learning and control with object-centric predictive models supervised by motion and view prediction, and 3D object detection.
- Model-based control by unrolling our model's predictions forward in time in the 3D latent feature space.
- Strong generalization to environments of novel objects, novel number of objects, novel objects spatial arrangements and novel camera viewpoints.
- Ablations and comparisons against both 3D engineered baselines and end-to-end learnt 2D feature baselines.

2 Related Work

Model learning and model-based control Many researchers have realized the importance of model learning, both for model-based control (Ha & Schmidhuber, 2018; Finn & Levine, 2016; Fragkiadaki et al., 2015) as well as a premise towards unsupervised learning of visuomotor representations (Agrawal et al., 2016; Pinto et al., 2016). Several computational models for learning intuitive physics have been proposed, under various names, such as, world models (Ha & Schmidhuber, 2018), action-conditioned prediction (Oh et al., 2015b), forward models (Miall & Wolpert, 1996), neural physics (Chang et al., 2016), neural simulators, etc. A central question is the representation space in which such predictive learning is carried out. One line of work proposes to learn such representations by predicting future visual observations directly (Oh et al., 2015b). To facilitate prediction of future frames, many works predict 2D pixel motion fields, which they use to warp frames forward in time in a differentiable manner (Finn & Levine, 2016; Ebert et al., 2017). This relies on the fact that pixel appearance persists while pixels themselves are rearranged spatially over time, yet such assumption can easily break during occlusions and camera motion where some pixels might disappear and some new pixels might be introduced to the frame. Some works constrain the flow fields to be piece-wise affine (Finn et al., 2016; Byravan & Fox, 2016), but cross-object occlusions still remain a problem. An alternative is to predict the future in some abstract feature representation, e.g., CNN feature encodings of future visual frames. One line of work learn such abstract frame encoding using autoencoding objectives (Stadie et al., 2015). Works of (Agrawal et al., 2016; Pathak et al., 2017) combine such (forward) model learning, with inverse model learning (given current and future observations or their encoding, they predict the action that caused the transition), in order to learn to ignore part of the visual scene not directly controllable by the agent. However, such models have not been shown to generalize across either camera viewpoints or scene changes (Kansky et al., 2017).

Obect-centric implicit representation To generalize across different scenes, there is a number of works that follow object-centric biases, detect objects in the visual frames and predict their 2D or 3D trajectories, and share weights of such object-centric predictive model across objects present in the scene (Fragkiadaki et al., 2015; Battaglia et al., 2016). Cross-object interactions are captured either implicitly by considering large image crops around an object (Fragkiadaki et al., 2015), or explicitly using edges in a graph neural network architecture (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2018). While these works propose object-centric predictive models that operate in either a 1D or 2D space, in this paper, we consider to extend the idea in a 3D feature space and propose to encode object feature by applying a wide enough 3D crop around the object.

Geometry-aware learnable 3D representations Some recent works have attempted various forms of geometrically-consistent temporal integration of visual information (Gupta et al., 2017; Henriques & Vedaldi, 2018; Tung et al., 2018), in place of geometry-unaware vanilla LSTM (Hochreiter & Schmidhuber, 1997) or GRU (Chung et al., 2014) models. Our work builds upon geometry-aware RNNs (GRNNs) of Tung et al. (2018). GRNNs are neural networks that integrate multi-view visual observations into geometrically-consistent egomotion-stabilized 3D deep feature maps by predicting image views from queried viewpoints. Our work extends those architectures for object-centric action-conditioned dynamics learning and model-based control.

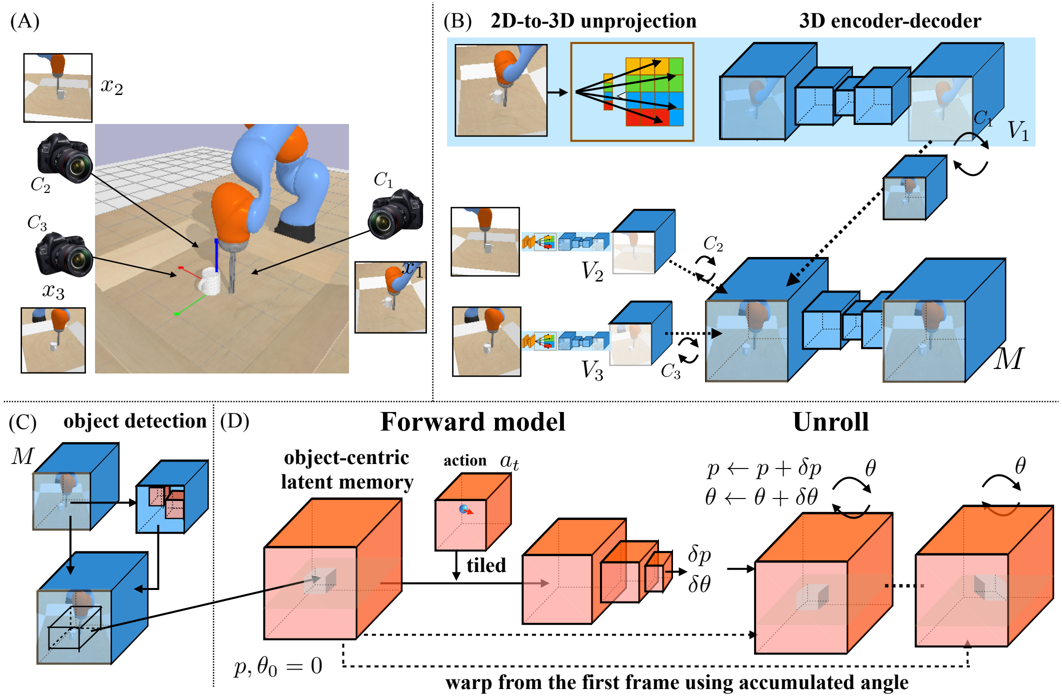

Figure 1: **Neural architectures for dynamics learning in an implicit 3D feature space (A):** Inputs to our model are one or more images of the scene captured from various camera viewpoints. **(B):** Each RGB image is unprojected into a set of 3D feature maps through ray shooting and 3D convolutions, oriented to match a common coordinate system, and integrated through plain averaging into a single 3D feature map **M**, following Tung et al. (2018) **(C):** An object detector detects 3D object boxes and 3D object voxel occupancies using the scene 3D feature **M** as input. **(D):** The detected boxes are used to crop the scene 3D feature maps around the object of interest and feed it to a 3D encoder-decoder network to predicts the object's future 3D relative rotation and translation for the object. The action of the agent is represented as a segmentation and 3D flow in the object crop.

3 Learning object-centric 3D implicit dynamic models

We consider agents equipped with cameras and end-effectors that can change camera viewpoint and push objects present in the scene (Figure 1). Our dynamics model is a recurrent neural network that given one of more RGB-D views $x_i, i = 1 \cdots K$ of a (static) scene from corresponding camera viewpoints $C_i, i = 1 \cdots K$ whose relative pairwise poses we assume known, and an optional action of the end-effector, it predicts the future state for the object of interest.

Our model disentangles 3D appearance of the scene from the motion of the camera—changes of viewpoint—and motion of the objects. It represents the scene appearance in terms of world-centric 3D feature maps, as opposed to image-centric 2D feature maps of vanilla CNNs. In comparison to engineered 3D representations that also disentangle appearance from camera motion, our 3D object feature appearance is learned end-to-end from any number of 2.5D camera views by optimizing for the end dynamics prediction objective. Thus, the proposed model combines the best of 3D object permanence and end-to-end learning. We assume that the 3D appearance of an object does not change over time. Under this assumption, the future state of the object is entirely represented by the 3D rotation and translation it undergoes as a result of the (optional) pushing action. We note that appearance constancy holds exactly in 3D for most rigid objects, but only approximately in 2D due to occlusions and dis-occlusions.

Our model's architecture builds upon geometry-aware recurrent neural networks (GRNNs) of Tung et al. (2018), that given RGB or RGB-D image sequences as input, learn a set of geometrically-consistent 3D deep feature maps of the scene $\mathbf{M} \in \mathbb{R}^{w \times h \times d \times c}$, by optimizing a downstream objective. In contrast to popular convolutional LSTM or GRU models used in the literature (Hochreiter

& Schmidhuber, 1997; Shi et al., 2015; Song et al., 2018; Zhang et al., 2018), whose hidden state is image-centric, GRNNs' hidden state is world-centric: it is a multi-dimensional tensor with 3 spatial dimensions (X, Y, Z) and multiple feature dimensions, akin to a 3D map of the scene, which for every (x, y, z) grid location holds an 1-dimensional feature vector F, as we show in Figure 1 right. That feature vector describes the semantic and geometric content of a corresponding 3D physical point in the 3D world scene. The map is updated with each new video frame in an **egomotion-stabilized manner**: deep features are transformed to match the coordinate system of the map before fusing features, so that information from 2D pixels that correspond to the same 3D physical point end up nearby in the map. GRNNs are equipped with trainable neural modules in order to map between 2D pixel space and 3D feature space in a differentiable manner, as well as to estimate and transform image features to cancel egomotion. They are inspired by Simultaneous Localization and Mapping (SLAM) methods (Sturm et al., 2012), but instead of 3D pointcloud maps they build 3D feature maps. These feature maps can represent a wide variety of information that is related to the downstream task, as opposed to merely 3D occupancy. The tasks they considered are 3D object detection and RGB view prediction. Our work extends those architectures to dynamic scenes, where object displacements are caused by an agent pushing objects around on a table surface. We focus on learning object-centric motion dynamics and unroll those forward in time for model-based control.

Our 3D object detector module follows the architecture of Mask R-CNN (He et al., 2017), a state-of-the-art 2D visual detector, re-purposed to operate with 3D input and output: the input is the motion-disentangled 3D feature map $\mathbf{M}$ and the output is object 3D bounding boxes and 3D object binary voxel occupancies. We use supervision from groundtruth 3D object boxes and masks supplied by our physics simulator. For each detected object, we compute a fixed-size axis-aligned 3D box around the predicted 3D centroid and crop accordingly to obtain an object-centered 3D feature map,fed into our object-centric dynamics module. We use the predicted object 3D segmentation mask m to sample nearby gripper locations for initiating the pushing motion. Specifically, the action of the agent is represented in terms of a segmentation mask for the robotic arm and a corresponding 3D flow field of the gripper's motion, resulting in an action representation in the Cartesian space.

Our object-centric dynamics module is a single-step 3D motion predictor. Given an object-centric 3D feature crop and the action representation, it predicts the object 3D motion for the next time step, in terms of relative rotation $\delta\theta$ and translation δp, as shown in Figure 1. Specifically, we predict the quaternion representation of the relative 3D object rotation. For every object in the scene and for each time step our loss reads:

$$\mathcal{L}(\delta\hat{p}, \delta\hat{\theta}, \delta p, \delta\theta) = ||\delta p - \delta\hat{p}||_1 + (1 - \langle\delta\theta \cdot \delta\hat{\theta}\rangle^2)), \tag{1}$$

where $\delta\hat{p}$ and $\delta\hat{\theta}$ denote groundtruth translation and rotation, respectively.

Model unrolling using temporal skip connections Our dynamics module predicts the single step 3D motion of an object given an agent's actions. To predict long term results of actions, or results of long action sequences, the model is unrolled over time by feeding its predictions back as input. Such forward model unrolling is notorious for causing error accumulation over time (Ross et al., 2010). Permanence of 3D representations allows us to use a novel model unrolling mechanism that iteratively warps the *initial* 3D object appearance, as opposed to the one computed in the last time step, minimizing error accumulation. Specifically, given estimated 3D object motion, δp, $\delta\theta$, and predicted from the 3D detector object mask m, we rotate and translate the object differentiably using 3D spatial transformers. We iteratively update the 3D rotation and translation with respect to the first time step, and each time warp the 3D object feature tensor using the cumulative predicted motion. After we rotate and translate the object features and the corresponding mask, we use them to compose a new scene tensor and crop again at the predicted object location. The predicted appearance for both the scene and the objects can be updated using the following equations:

$$m^o_{t+1} = R(m^o_0, \theta^o_t), \forall o \in O \tag{2}$$

$$\mathbf{M}_{t+1} = \sum_{o \in O} \mathrm{DRAW}(m^o_{t+1} \odot R(\mathbf{M}^o_0, \theta^o_t), p^o_{t+1}), \forall o \in O \tag{3}$$

$$\mathbf{M}^{o_i}_{t+1} = m^{o_i}_{t+1} \odot R(\mathbf{M}^{o_i}_0, \theta) + (1 - m^{o_i}_{t+1}) \odot \sum_{o_j \in O \wedge o_j \neq o_i} \mathrm{CROP}(\mathbf{M}_{t+1}, p^o_{t+1}), \forall o_i \in O, \tag{4}$$

where $\odot$ denotes element-wise multiplication, $R(\cdot, \theta)$ denotes the rotation operation given angle θ, $\mathrm{DRAW}(\cdot, p)$ denotes the operation of putting an object-centric tensor back to the scene tensor at location p, and $\mathrm{CROP}(\cdot, p)$ denotes cropping an object-centric tensor from the scene tensor, and m denotes 3D object mask in terms of a binary 3D voxel occupancy. Such temporal skip connections is a benefit of the what-where decomposition of the proposed 3D feature representations.

Model-based control To control an object towards desired locations on the table surface we follow a receding horizon control (Tassa et al., 2008) scheme. We search over randomly sampled actions around the objects. For each sampled action sequence, we unroll the model forward in time as detailed in the previous paragraph, evaluate how close its predicted final location is to the desired goal 3D location, and select the action sequence with the minimum error. We then execute the first action of the selected sequence and re-plan, till either we reach the goal or a maximum number of planning look-ahead steps. Specifically, a receding horizon window of 1 step proves sufficient in our experiments.

Auxiliary view-prediction objective The proposed architectures are end-to-end differentiable. We train them under the joint objectives of 3D object detection, 3D object motion forecasting, and RGB prediction, an auxiliary self-supervised objective, similar to Tung et al. (2018). This helps the 3D object detector generalize to unknown objects with smaller number of groundtruth annotations, as independently empirically verified by authors of Harley et al. (2019).

4 Experiments

We train and test our dynamics model in the Bullet Physics Simulator. We create scenes using 31 different 3D object meshes, including 11 objects from the MIT Push dataset (Yu et al., 2016) and 20 objects selected from 4 categories (*camera*, *mug*, *bowl* and *bed*) in the Shapenet Dataset (Chang et al., 2015). Each scene is observed from multiple viewpoints. We consider a Kuka robotic arm equipped with a single rod (as shown in Figure 1) pushing objects on the table. Each pushing video contains a pushing trajectory of 5 steps. The objects move freely on a planar workspace of size $0.6m \times 0.6m$. We show experimental results below on pushing tasks in both 1-obj scenes and 2-obj scenes. Specifically, 2-obj experiments contain data where the object pushed by the robot in turn pushes the other object. With this setting we show our model's capability to model dynamics for inter-object interactions.

We train our model using three randomly selected views as input, and we test it using one or three randomly selected views as input. All images are 128×128. Further details of the dataset collection are included in the appendix.

Our experiments aim to evaluate the accuracy of our model for single step and multi-step motion prediction, its ability to generalize across camera viewpoints, its performance in pushing objects to desired locations, and its ability to detect and segment objects in 3D from RGB streams in diverse environments, a necessary step for applying the learned object-centric dynamics.

We compare our model against the following baselines: i) a model that uses groundtruth 3D object centroid position and orientation as object state (XYZ), similar to (OpenAI et al., 2018; Wu et al., 2017), ii) a model that uses object-centered 2D image patches, computes 2D feature embeddings, tiles the embeddings with the action representation and camera pose of the image, and concatenates those across views (2D-multiview), extending 2D object-centric models to take multiple views as input, iii) our model with access only to multiview depth with no RGB information (ours-depth).

4.1 Single step motion prediction

We evaluate performance of our model and baselines in single step motion prediction in Tables 1 and 2. We evaluate predicted translation error in Euclidean distance and predicted rotation error in degrees. In Table 1, we show the performance of our model and baselines using different number of views for pushing 1 object. The XYZ baseline does not use image views and has the same error in both scenarios. Our model outperforms the baselines both in position and orientation prediction. 2D-multiview performs on par with XYZ, which means it does not gain from having access to additional appearance information. The prediction error of our single view model is only slightly higher than the model using three random views as input. Our model is flexible enough to accept a variable number of views as input, and improve the more views are available, yet, it can accurately predict future motion even from a single view. As we see, the 2D-multiview baseline does not improve with more views available. We believe this is due to the geometry-unaware way of combining multiview information by concatenation, though the model does have access to camera poses of the input images.

In Table 2 row 1, we show prediction performance in the 2-obj scenes. Our proposed model shows a clear advantage over all the baselines. All the experiments so far are done with the aforementioned single rod as the Kuka robot's end-effector. Additionally, we conducted another experiment where we test our model's performance over varing end-effector geometries. We train our model in the 2-obj scene, using only the single rod Kuka robot as the pushing agent for data collection, and then test the model on pushing tasks using robot with 6 different end-effectors with varying geometries (see the appendix for the detailed designs). Interestingly, as shown in Table 2 row 2, our model is able to generalize well to novel end-effector design although it's only trained on a simple rod-shape end-effector, since it captures such dynamics by learning inter-object interactions, where the objects provide geometry diversities.

	Experiment Setting	XYZ	2D-multiview	ours-depth	ours
object position (mm)	3views + gt-bbox	17.68	17.72	12.38	**9.02**
object orientation (degree)		6.13	7.68	5.79	**5.62**
object position (mm)	1view + gt-bbox	17.68	17.60	12.26	**9.80**
object orientation (degree)		6.13	7.85	5.96	**5.84**

Table 1: Single step prediction error with a single object. We show prediction error using different number of views as inputs.

	Experiment Setting	XYZ	2D-multiview	ours-depth	ours
object position (mm)	3views + gt-bbox	13.9	13.9	8.30	**7.12**
object orientation (degree)		10.3	11.7	3.90	**3.94**
object position (mm)	3views + gt-bbox + varfinger	19.4	23.2	13.6	**12.5**
object orientation (degree)		11.4	12.8	**4.06**	4.07

Table 2: Single step prediction error with **multiple objects**. We evaluate the models with fingertips of various shape.

Inferring 3D object detections for object-centric dynamics Most previous works on learning object-centric dynamic models assume object segmentations given, i.e., they assume the state of the simulator known, and they focus on approximating its next-step function. However, accurately detecting objects in 3D is an unsolved problem, and given the projection artifacts and occlusions in the input RGB stream, 3D object detection is as difficult as, if not more difficult, than applying learned dynamics over ground-truth 3D boxes. The architectures we propose invert the input RGB stream to obtain a stable 3D feature model of the scene. Such architectures are ideal for extracting 3D object detections that persist over time despite occlusions (Tung et al., 2018). In Table 3, we show motion prediction performance using the 3D object detector built on top of the 3D latent representation. Our model outperforms all the baselines, and reaches comparable results as the one using groudtruth object poses.

	Experiment Setting	XYZ	2D-multiview	ours-depth	ours
object position (mm)	3views + est-bbox	18.24	17.88	17.60	**14.56**
object orientation (degree)		10.03	7.62	6.94	**6.76**

Table 3: Single step prediction error with a single object with **detected objects**. We show prediction error using different number of views as inputs, using object pose estimated by the proposed 3D object detector.

4.2 Multi-steps motion prediction

We show unrolling results using the dynamic models trained on single-step motion. XYZ is trivial to unroll forward in time without much error accumulation since it does not use any appearance features. Still, our model outperforms it. Though our model uses appearance features in the form of 3D feature maps, it uses temporal skip connections to avoid error accumulation. XYZ is blind to the object's appearance and thus cannot learn fine-grained motion forecasting for different tailored to different object shapes. We show qualitative results for multi-step motion prediction in Figure 2.

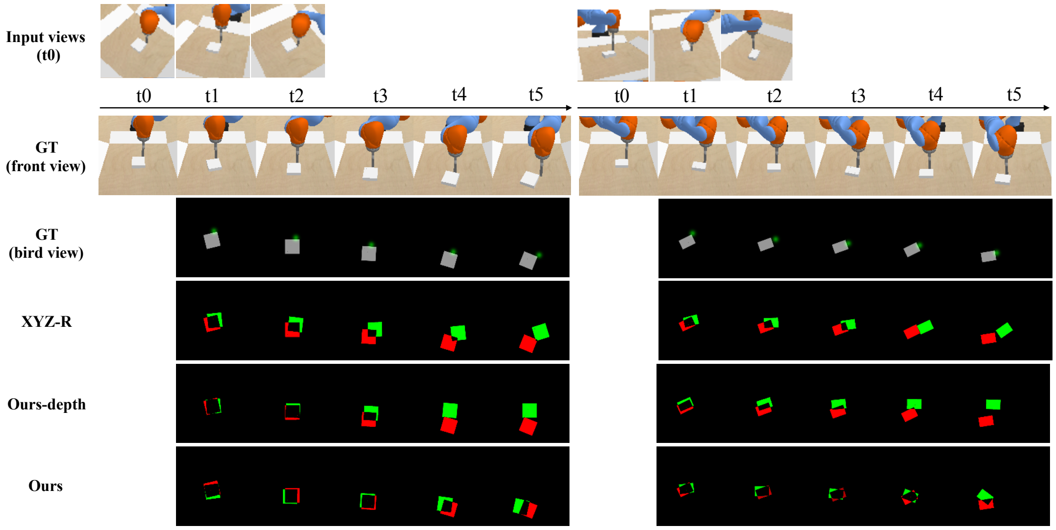

Figure 2: **Forward unrolling of our dynamics model and baselines** In the top row, we show randomly sampled input image views for our model and the ours-depth baseline. The second row shows the ground-truth motion of the object from the front view. Rows 3-6 show the object motion from an overhead camera. The ground truth object poses are colored in red while the predicted object poses are colored in green. Their intersected regions stay in black. Our model shows a clear performance margin over the baselines.

4.3 Model-based control

We test the performance of our model in single-step motion prediction by applying it with model based control. We run 50 examples by randomizing initial and desired object 3D location, as well as randomizing the camera viewpoint. It is considered a success pushing sequence if all objects end up within 5cm (about half of the average object size) from the target positions on average. We compare our model against the XYZ baseline in Table 4.

	Setting	XYZ	ours
success rate	1 object	0.76	**0.86**
success rate	2 objects	0.64	**0.74**

Table 4: Success rate for pushing objects to target 3D locations.

5 Conclusion

We have presented models of object dynamics that invert 2D video streams into a learnt 3D feature space, detect 3D objects, and predict their future motion therein, given actions of the agent. The proposed models enjoys the benefits of permanence of 3D representations and end-to-end feature learning, can generalize across camera viewpoints, can effectively detect objects in 3D to support object-centric dynamics forecasting, and benefit from view prediction auxiliary objectives. Our empirical findings suggest that learning models of intuitive physics benefits from **state representations that themselves obey physics**: states that do not arbitrarily "shake" under camera motion, do not change size, or appear and disappear, but rather persist over time and are viewpoint invariant, similar to the egomotion-stabilized perception we, humans, are capable of. Extending this model to non-rigid object dynamics and multi-object interactions is a clear avenue for future work.

REFERENCES

Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *CoRR*, abs/1606.07419, 2016. URL http://arxiv.org/abs/1606.07419.

Peter W. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *CoRR*, abs/1612.00222, 2016. URL http://arxiv.org/abs/1612.00222.

A. Byravan and D. Fox. SE3-Nets: Learning rigid body motion using deep neural networks. *CoRR*, abs/1606.02378, 2016.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

Michael B. Chang, Tomer Ullman, Antonio Torralba, and Joshua B. Tenenbaum. A compositional object-based approach to learning physical dynamics. *CoRR*, abs/1612.00341, 2016. URL http://arxiv.org/abs/1612.00341.

Silvia Chiappa, Sébastien Racanière, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. *CoRR*, abs/1704.02254, 2017. URL http://arxiv.org/abs/1704.02254.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL http://arxiv.org/abs/1412.3555.

Frederik Ebert, Chelsea Finn, Alex X. Lee, and Sergey Levine. Self-supervised visual planning with temporal skip connections. *CoRR*, abs/1710.05268, 2017. URL http://arxiv.org/abs/1710.05268.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. *CoRR*, abs/1610.00696, 2016. URL http://arxiv.org/abs/1610.00696.

Chelsea Finn, Ian J. Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *CoRR*, abs/1605.07157, 2016. URL http://arxiv.org/abs/1605.07157.

Katerina Fragkiadaki, Pulkit Agrawal, Sergey Levine, and Jitendra Malik. Learning visual predictive models of physics for playing billiards. *CoRR*, abs/1511.07404, 2015. URL http://arxiv.org/abs/1511.07404.

Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2616–2625, 2017.

David Ha and Jürgen Schmidhuber. World models. *CoRR*, abs/1803.10122, 2018. URL http://arxiv.org/abs/1803.10122.

Adam W Harley, Fangyu Li, Shrinidhi K Lakshmikanth, Xian Zhou, Hsiao-Yu Fish Tung, and Katerina Fragkiadaki. Embodied view-contrastive 3d feature learning. *arXiv*, 2019.

Masahiko Haruno, Daniel M Wolpert, and Mitsuo Kawato. Multiple paired forward-inverse models for human motor learning and control. In M. J. Kearns, S. A. Solla, and D. A. Cohn (eds.), *Advances in Neural Information Processing Systems 11*, pp. 31–37. MIT Press, 1999. URL http://papers.nips.cc/paper/1585-multiple-paired-forward-inverse-models-for-human-motor-learning-and-control.pdf.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870.

J. F. Henriques and A. Vedaldi. MapNet: An allocentric spatial memory for mapping environments. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Ken Kansky, Tom Silver, David A. Mély, Mohamed Eldawy, Miguel Lázaro-Gredilla, Xinghua Lou, Nimrod Dorfman, Szymon Sidor, D. Scott Phoenix, and Dileep George. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. *CoRR*, abs/1706.04317, 2017. URL http://arxiv.org/abs/1706.04317.

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B. Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJgbSn09Ym.

Brenden M. Lake, Tomer Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Center for Brains, Minds Machines (CBMM) Memo No. 046*, arXiv, 04 2016. doi: 10.1017/S0140525X16001837.

Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015. URL http://arxiv.org/abs/1511.05440.

R. C. Miall and D. M. Wolpert. Forward models for physiological motor control. *Neural Netw.*, 9 (8):1265–1279, November 1996. ISSN 0893-6080. doi: 10.1016/S0893-6080(96)00035-4. URL http://dx.doi.org/10.1016/S0893-6080(96)00035-4.

Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li Fei-Fei, Joshua B Tenenbaum, and Daniel LK Yamins. Flexible neural representation for physics prediction. In *Advances in Neural Information Processing Systems*, 2018.

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

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 28*, pp. 2863–2871. Curran Associates, Inc., 2015b. URL http://papers.nips.cc/paper/5859-action-conditional-video-prediction-using-deep-networks-in-atari-games.pdf.

OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub W. Pachocki, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *CoRR*, abs/1808.00177, 2018. URL http://arxiv.org/abs/1808.00177.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. *CoRR*, abs/1705.05363, 2017. URL http://arxiv.org/abs/1705.05363.

Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta. The curious robot: Learning visual representations via physical interactions. *CoRR*, abs/1604.01360, 2016. URL http://arxiv.org/abs/1604.01360.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. No-regret reductions for imitation learning and structured prediction. *CoRR*, abs/1011.0686, 2010. URL http://arxiv.org/abs/1011.0686.

Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4470–4479, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/sanchez-gonzalez18a.html.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 802–810, Cambridge, MA, USA, 2015. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969239.2969329.

Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Bradly C. Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *CoRR*, abs/1507.00814, 2015. URL http://arxiv.org/abs/1507.00814.

J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, 2012.

Yuval Tassa, Tom Erez, and William D Smart. Receding horizon differential dynamic programming. In *Advances in neural information processing systems*, pp. 1465–1472, 2008.

Hsiao-Yu Fish Tung, Ricson Cheng, and Katerina Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. *arXiv:1901.00003*, 2018.

Jiajun Wu, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum. Learning to see physics via visual de-animation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 153–164. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/6620-learning-to-see-physics-via-visual-de-animation.pdf.

Kuan-Ting Yu, Maria Bauzá, Nima Fazeli, and Alberto Rodriguez. More than a million ways to be pushed: A high-fidelity experimental data set of planar pushing. *CoRR*, abs/1604.04038, 2016. URL http://arxiv.org/abs/1604.04038.

Dongqing Zhang, Ilknur Icke, Belma Dogdas, Sarayu Parimal, Smita Sampath, Joseph Forbes, Ansuman Bagchi, Chih-Liang Chin, and Antong Chen. A multi-level convolutional LSTM model for the segmentation of left ventricle myocardium in infarcted porcine cine MR images. *CoRR*, abs/1811.06051, 2018. URL http://arxiv.org/abs/1811.06051.

A Appendix

A.1 Data collection details

For each sequence, we render images from 27 different views including 9 different azimuth angels ranging from the left side of the agent to the right side of the agent combining with 3 different elevation angles from 20, 40, 60 degrees. All cameras are looking at the 0.1m above the center of the table, and are 1 meter away from the look-at point.

A.2 End-effector designs

We designed 6 different types of end-effectors for evaluating our model's generalizability towards novel end-effector geometries. They are *rod*, *hexagon*, *circle*, *ellipse*, *square* and *triangle*, as shown in Figure 3.

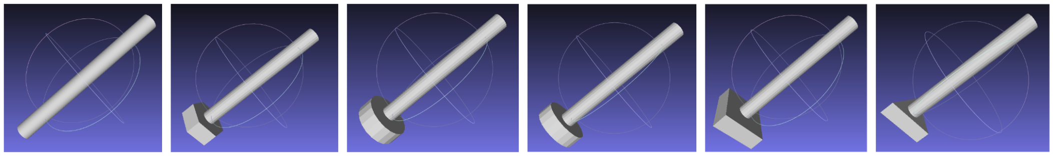

Figure 3: Various end-effector geometries for generalizability evaluation

A.3 Implementation details

We train our model and baselines for single step prediction. Both after unprojection and after aggregation, we have a 3D encoder-decoder tower with 4 3D convolutional layers and 4 deconvolutional layers. The channel size is set to 8, 16, 32, 64. We apply relu activation and batch normalization after each layer. The size of the object-centric latent memory is set to 16. To convert the object memory to the final motion prediction, we use 4 convolutional layers without batch normalization. The channel size is set to 32, 32, 64, respectively. We then flatten it as a vector and pass it through two more fully-connected layers. For the XYZ-orn baseline, we use four fully-connected layers with the size of 32 for each. We also experiment with deeper and wider network, but the performance seems similar. For the appearance-based baseline, we use seven convolutional layers with channel size 16, 32, 32, 64, 64, 128 and filter size 3,5,3,5,3,5,3. Each layers has batch normalization and relu activation. We again flatten the output from the convolutional layers and pass it to two fully connected layers. For the SLAM baseline, we remove the 2D encoder-decoder tower from our model and we double the channel size in the following 3D convolutional layers. The learning is set to $1e-3$ for all the experiments with Adam optimizer.