LEARNING UNSUPERVISED FORWARD MODELS FROM OBJECT KEYPOINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Object-centric representation is an essential abstraction for forward prediction. Most existing forward models learn this representation through extensive supervision (e.g., object class and bounding box) although such ground-truth information is not readily accessible in reality. To address this, we introduce KINet (Keypoint Interaction Network)—an end-to-end unsupervised framework to reason about object interactions based on a keypoint representation. Using visual observations, our model learns to associate objects with keypoint coordinates and discovers a graph representation of the system as a set of keypoint embeddings and their relations. It then learns an action-conditioned forward model using contrastive estimation to predict future keypoint states. By learning to perform physical reasoning in the keypoint space, our model automatically generalizes to scenarios with a different number of objects, novel backgrounds, and unseen object geometries. Experiments demonstrate the effectiveness of our model in accurately performing forward prediction and learning plannable object-centric representations which can also be used in downstream robotic manipulation tasks.

1 INTRODUCTION

Discovering a structured representation of the world allows humans to perform a wide repertoire of motor tasks such as interacting with objects. At the core of this process is the ability to predict the response of the environment to applying an action (Miall & Wolpert, 1996; Wolpert & Kawato, 1998). Such an internal model, often referred to as the forward model, aims to come up with an estimation of the future states of the world given its current state and an action. By cascading the predictions of a forward model it is also possible to plan a sequence of actions that would bring the world from an initial state to a desired goal state (Wolpert et al., 1998; 1995).

Recently, various deep learning architectures have been proposed to perform forward modeling using an object-centric representation of the system (Ye et al., 2020; Chen et al., 2021b; Li et al., 2020; Qi et al., 2020). This representation is learned from the visual observation by factorizing the scene into the underlying object instances using ground-truth object states (e.g., object class, position, and bounding box). We identified two major limitations in the recent work: First, they either assume access to the ground-truth object states (Battaglia et al., 2016; Li et al., 2020) or predict them using idealized techniques such as pre-trained object detection or instance segmentation models (Ye et al., 2020; Qi et al., 2020). However, obtaining ground truth object states is not always feasible in practice. Relying on object detection and segmentation tools, on the other hand, makes the forward model fragile and dependent on the flawless performance of these tools which is often infeasible in real-world settings. Second, factorizing the scene into a fixed number of object instances limits the generalization of forward models to scenarios with a different number of objects.

In this paper, we address both of these limitations by proposing to learn forward models using a keypoint-based object-centric representation. Keypoints represent a set of salient locations of moving entities. Our model KINet (Keypoint Interaction Network) learns an unsupervised forward model in three major steps: (1) A keypoint extractor factorizes the scene into keypoint coordinates with no supervision other than raw visual observations. (2) A probabilistic graph representation of the system is learned where each node corresponds to a keypoint and edges are keypoints relations. Node features carry implicit object-centric representation as well as explicit keypoint state information. (3) With probabilistic message passing, our model learns an action-conditional forward model

to predict future locations of the keypoints. Using this prediction, it also reconstructs the future appearances of the scene. We evaluate KINet's forward prediction accuracy and demonstrate that, by learning the physical reasoning in a keypoint coordinate, our model can effectively re-purpose this knowledge and generalize it to complex unseen circumstances.

Our key contributions are as follows: (1) We introduce KINet, an end-to-end method for learning unsupervised action-conditional forward models from visual observations using keypoint coordinates. (2) We introduce Probabilistic Interaction Networks for efficient message-passing to aggregate relevant information. (3) We introduce the GraphMPC algorithm for accurate action planning based on graph similarity. (4) We demonstrate that learning forward models in keypoint coordinates enables zero-shot generalization to complex unseen scenarios.

2 Related Work

Unsupervised keypoint extraction methods have been used successfully in computer vision tasks such as pose tracking (Zhang et al., 2018; Yao et al., 2019) and video prediction (Minderer et al., 2019; Zhang et al., 2018; Xue et al., 2016; Manuelli et al., 2020). Recent work explored keypoint extraction for control tasks in reinforcement learning environments to project the visual observation space to a lower-dimensional keypoint space (Kulkarni et al., 2019; Chen et al., 2021a; Jakab et al., 2018; Suwajanakorn et al., 2018). We extend these methods for the object manipulation setting.

Forward prediction models estimate the future state of a system given its current state. Neural networks can be used to estimate such models for various applications. Example of earlier approaches are Nguyen & Widrow (1990) and Grzeszczuk et al. (1998) for modeling state transition in dynamic systems. The most fundamentally relevant work to ours is the Interaction Network (IN) (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2018) and follow-up work on using graph neural network in forward modeling (Pfaff et al., 2020; Li et al., 2019; Kipf et al., 2018; Mrowca et al., 2018). Although these approaches demonstrated powerful forward modeling, they rely on building explicit object representations based on the ground-truth state information of objects. In an attempt to bridge this gap several methods have been proposed to use IN-based models based on visual features extracted from image observations (Watters et al., 2017; Qi et al., 2020; Ye et al., 2020). However, two main concerns remain unaddressed in these methods. First, the visual features are often extracted from object bounding boxes using object detection or segmentation model (Janner et al., 2018; Qi et al., 2020; Kipf et al., 2018) that is either pretrained on the environment (Qi et al., 2020) or has the ground-truth object position (Ye et al., 2020). Second, these approaches lack generalization as they are formulated on a fixed number of objects.

Learning action-conditional forward models is an active challenge. The original IN augments the action vector as an external effect to the node embeddings. For probabilistic forward models Henaff et al. (2019) interestingly suggests using a latent variable dropout mechanism to properly condition the forward model on the action (Gal & Ghahramani, 2016). In a more relevant application to ours, Yan et al. (2020) highlighted the effectiveness of contrastive estimation (Oord et al., 2018) to learn proper actionable object-centric representations.

Graph representation inference methods have been introduced in variational graph autoencoders (Kipf & Welling, 2016). To formulate the graph structure inference we were also inspired by Simonovsky & Komodakis (2018) where they use a variational graph autoencoder to generate small graphs of molecular structures. More relevant to our work is Kipf et al. (2019) where an object-level contrastive loss has been used to learn object-centric abstractions in multibody systems. However, this work is only based on fixed environments with oversimplified objects such as two-dimensional geometries (Watters et al., 2019; Stanić et al., 2020). In our work, we experiment with more complex three-dimensional objects with randomized properties to replicate real-world settings.

3 Keypoint Interaction Networks (KINet)

We assume access to observational data that consists of RGB image, action vector, and the resulting image after applying the action: $\mathcal{D} = \{(I_t, u_t, I_{t+1})\}$. Our goal is to learn a forward model that predicts future states of the system with no supervision above the observational data. We describe

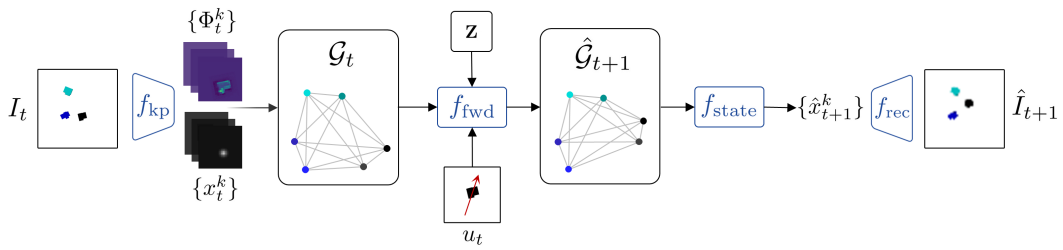

Figure 1: KINet performs forward modeling in three major steps: extracting keypoints, inferring a probabilistic graph representation, and estimating the future appearance conditioned on the action. Some arrows are simplified for clarity.

our approach in two main steps (see Figure 1): learning to encode visual observations into keypoints and then learning an action-conditioned forward model in the keypoint space.

3.1 Unsupervised Keypoint Detection

The keypoint detector (f_{kp}, Fig.1) is a mapping from visual observations to a lower-dimensional set of K keypoint coordinates $\{x_t^k\}_{k=1\ldots K} = f_{\text{kp}}(I_t)$. The keypoint coordinates are learned through capturing the spatial appearance of the objects in an unsupervised manner. Specifically, the keypoint detector receives a pair of initial and current images (I_0, I_t) and uses a convolutional encoder to compute a K-dimensional feature map for each image $\Phi(I_0), \Phi(I_t) \in \mathbb{R}^{H' \times W' \times K}$. The expected number of keypoints is set by the dimension K. Next, the feature maps are marginalized into a 2D keypoint coordinate $\{x_0^k, x_t^k\}_{k=1\ldots K} \in \mathbb{R}^2$. We use a convolutional image reconstruction model (f_{rec}, Fig.1) with skip connections to inpaint the current image frame using the initial image and the predicted keypoint coordinates $\hat{I}_t = f_{\text{rec}}(I_0, \{x_0^k, x_t^k\}_{k=1\ldots K})$. With this formulation, f_{kp} and f_{rec} create a bottleneck to encode the visual observation in a temporally consistent lower-dimensional set of keypoint coordinates (Kulkarni et al., 2019).

3.2 Graph Representation of System

After factorizing the system into K keypoints, we build a graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{Z})$ (undirected, no self-loop) where keypoints and their pairwise relations become the graph nodes and edges. Keypoint positional and visual information are encoded into the embedding of nodes $\{\mathbf{n}_t^k\}_{k=1\ldots K} \in \mathcal{V}_t$ and edges $\{\mathbf{e}_t^{ij}\} \in \mathcal{E}_t$. We also define an adjacency matrix to specify the graph connectivity as $\mathbf{Z} \in \mathbb{R}^{K \times K}$ where $[\mathbf{Z}]_{ij} \in [0, 1]$ specifies the probability of the edge $\{\mathbf{e}^{ij}\} \in \mathcal{E}$. At each timestep t, node embeddings encode keypoint visual features and positional information $\{\mathbf{n}_t^k\} = \left[\Phi_t^k, x_t^k\right]$. Edge embeddings also contain relative positional information of each node pair $\{\mathbf{e}_t^{ij}\} = \left[x_t^i - x_t^j, \|x_t^i - x_t^j\|_2^2\right]$.

3.3 Probabilistic Interaction Networks

To build a forward model, we extend the recent approaches and propose a probabilistic variation of the Interaction Networks (IN) (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2018). The core of probabilistic IN is generating edge-level latent variables $z_{ij} \in \mathbb{R}^d$ that represents the edge probability $[\mathbf{Z}]_{ij} = z_{ij}$. A posterior network p_θ infers the elements of the adjacency matrix given the graph representation of the scene. In particular, we model the posterior as $p_\theta(z_{ij}|\mathcal{G}_t) = \sigma(f_{\text{enc}}([\mathbf{n}_t^i, \mathbf{n}_t^j]))$ where $\sigma(.)$ is the sigmoid function.

The probabilistic IN forward model $\hat{\mathcal{G}}_{t+1} = f_{\text{fwd}}(\mathcal{G}_t, u_t, \mathbf{Z})$ predicts the graph representation at the next timestep by taking as input the current graph representation and action (f_{fwd}, Fig.1). The message-passing operation in the forward model can be described as, $\{\hat{\mathbf{e}}^{ij}\} \leftarrow f_{\text{e}}(\mathbf{n}^i, \mathbf{n}^j, \mathbf{e}^{ij})$, $\{\hat{\mathbf{n}}^k\} \leftarrow f_{\text{n}}(\mathbf{n}^k, \sum_{i \in N(k)} \mathbf{Z}_{ik} \hat{\mathbf{e}}^{ik}, u_t)$ where the edge-specific function f_{e} first updates edge embeddings, then the node-specific function f_{n} updates each node embedding $\hat{\mathbf{n}}^k$ by probabilistically aggregating its neighboring nodes $N(k)$ information (i.e., the edge probabilities from the inferred

adjacency matrix $\mathbf{Z}$ is used as weights in the neighbor aggregation). The action vector u_t is also an input to the neighbor aggregation step. Note that the functions f_{enc}, f_{e}, and f_{n} are MLPs.

Recent models for forward prediction rely on fully connected graphs for message passing (Qi et al., 2020; Ye et al., 2020; Li et al., 2018). Our model, however, learns to probabilistically sample the neighbor information. Intuitively, this adaptive sampling allows the network to efficiently aggregate the most relevant neighboring information. This is specifically essential in our model since keypoints could provide redundant information if they are in very close proximity.

3.4 Forward Prediction

The state decoder (f_{state}, Fig.1) transforms the predicted node embeddings of the updated graph $\hat{\mathcal{G}}_{t+1}$ to a first-order difference which is integrated once to predict the position of the keypoints in the next timestep $\{\hat{x}_{t+1}^k\} = \{x_t^k\} + f_{\text{state}}(\{\hat{\mathbf{n}}_{t+1}^k\})$. To reconstruct the image at the next timestep, we borrow the reconstruction model f_{rec} from the keypoint detector $\hat{I}_{t+1} = f_{\text{rec}}(I_0, \{x_0^k, \hat{x}_{t+1}^k\})$.

3.5 Learning KINet

Reconstruction loss. The keypoint detector is trained using the distance between the ground truth image and the reconstructed image at each timestep $\mathcal{L}_{\text{rec}} = \|\hat{I}_t - I_t\|_2^2$. As suggested by Minderer et al. (2019), errors from the keypoint detector were not backpropagated to other modules of the model. This is a necessary step to ensure the model does not conflate errors from image modules and reasoning modules.

Forward loss. The model is also optimized to predict the next state of the keypoints. A forward loss penalizes the distance between the estimated future keypoint locations using first-order state decoder and the keypoint extractor predictions: $\mathcal{L}_{\text{fwd}} = \sum_K \|\hat{x}_{t+1}^k - f_{\text{kp}}(I_{t+1})^k\|_2^2$.

Inference loss. Our model is also trained to minimize the KL-divergence between the posterior and prior distributions: $\mathcal{L}_{\text{infer}} = D_{\text{KL}}\big(p_\phi(\mathbf{Z}|\mathcal{G})\big\|p(\mathbf{Z})\big)$. We use independent Gaussian prior $p(\mathbf{Z}) = \prod_i \mathcal{N}(\mathbf{z}_i)$ and use reparameterization trick for training (Kingma & Welling, 2013).

Contrastive loss. To further encourage the model to learn actionable representations, we make use of the contrastive estimation method. We add a contrastive loss (Oord et al., 2018; Yan et al., 2020) and reframe it for graph embeddings as: $\mathcal{L}_{\text{ctr}} = -\mathbb{E}_{\mathcal{D}}[\log(\frac{\mathcal{S}(\hat{\mathcal{G}}_{t+1}, \mathcal{G}_{t+1}^+)}{\sum \mathcal{S}(\hat{\mathcal{G}}_{t+1}, \mathcal{G}_{t+1}^-)})]$. The predicted graph representations $\hat{\mathcal{G}}_{t+1}$ are maximally similar to their corresponding positive sample pair $\mathcal{G}_{t+1}^+ := \mathcal{G}_{t+1}$ and maximally distant from the negative sample pairs $\mathcal{G}_{t+1}^- := G_\tau \;\; \forall \tau \neq t+1$. We use a simple node embedding similarity as the graph matching algorithm $\mathcal{S}(\mathcal{G}_1, \mathcal{G}_2) = \sum_K \{n_1^k\}.\{n_2^k\}$.

Finally, the combined loss is: $\mathcal{L} = \lambda_{\text{rec}}\, \mathcal{L}_{\text{rec}} + \lambda_{\text{fwd}}\, \mathcal{L}_{\text{fwd}} + \lambda_{\text{infer}}\, \mathcal{L}_{\text{infer}} + \; \lambda_{\text{ctr}}\, \mathcal{L}_{\text{ctr}}$.

3.6 GraphMPC Planning with KINet

We use a learned KINet model and select actions based on Model Predictive Control (MPC) algorithm (Finn & Levine, 2017) in the graph embedding space (GraphMPC). With KINet forward prediction, we compute the predicted graph representation of the next timestep for multiple sampled actions. We then select the optimal action that produces the most similar graph representation to the goal graph representation $\mathcal{G}^{\text{goal}}$. We describe our GraphMPC algorithm with a time horizon of T as: $u_t^* = \arg\max\{\mathcal{S}\big(\mathcal{G}^{\text{goal}}, f_{\text{fwd}}(\mathcal{G}^t, \{u_{t:T}\})\big)\};\; t \in [0, T]$. Unlike performing conventional MPC only with respect to positional states, GraphMPC allows for accurately bringing the system to a goal state both explicitly (i.e, position) and implicitly (i.e, pose, orientation, and visual appearance).

4 Experimental Setups

Our experiments seek to address the following: (1) How accurate is our forward model? (2) Can we use the model for action planning? and (3) Does the model generalize to unseen circumstances?

4.1 DATASET

We apply our approach to learn a forward model for multi-object manipulation tasks. The task involves rearranging multiple objects in the scene and bringing them to the desired goal state using point-to-point pushing actions. We use MuJoCo 2.0 (Todorov et al., 2012) to generate simulated training and testing scenarios. We also use a real Sawyer robot pushing dataset from Ye et al. (2020).

We simulate 10K episodes of random pushing on multiple objects (1-5 objects) where a simple robot end-effector applies randomized pushing for 90 timesteps per episode. Each object has randomized geometry and color. We sample the object geometry from a predefined continuous range (geom_{train} for training and geom_{gen} for generalization to unseen geometries). Unseen geometries are designed to have elongated shapes to create complex out-of-distribution cases (see Figure 5). We only collect the 4-dimensional action vector (pushing start and end location) and RGB images of the scene before and after the action is applied. Images are obtained using an overhead (*Top View*) and an angled camera (*Angled View*) (see Appendix A.1 for more details).

For the real-robot data, we use the Sawyer robot pushing dataset in Ye et al. (2020) to exemplify how our proposed framework applies to the real settings. The real-robot data includes 5K random point-to-point pushing actions on 7 simple blocks. For each pushing action, an RGB image pair is provided along with the action vector in the image coordinates.

4.2 BASELINES

We compare our approach with existing methods for learning object-centric forward models:

Forward Model *(Forw)*: We train a convolutional encoder to extract visual features of the scene image (*Img*) and learn a forward model in the feature space.

Forward-Inverse Model *(ForwInv)*: We train a convolutional encoder to extract visual features of the scene image (*Img*) and jointly learn forward and inverse models in the feature space as in Agrawal et al. (2016). Each object has a separate feature channel.

Interaction Network *(IN)*: We follow Battaglia et al. (2016); Sanchez-Gonzalez et al. (2018) to build an Interaction Network based on the ground truth location of the objects. Each object representation contains the ground-truth position and velocity of the objects (*GT state*). This approach is only applicable to scenarios where the number of objects in the scene is known and fixed.

Visual Interaction Network *(VisIN)*: To replicate Ye et al. (2020); Watters et al. (2017), we train a convolutional encoder to extract visual features of fixed-size bounding boxes centered on ground-truth object locations (*GT state* + *Img*). We use the extracted visual features as object representations in the Interaction Network. This approach also requires prior knowledge of the number of objects in the scene.

4.3 TRAINING AND EVALUATION SETTING.

All models are trained on a subset of the simulated data with 3 objects (8K episodes: 80% training, 10% validation, and 10% testing sets). To evaluate generalization to a different number of objects, we use other subsets of data with 1, 2, 4, and 5 objects (~ 400 episodes for each case). We also measure generalization for unseen geometries (geom_{gen}). We train our model separately on images obtained from the overhead camera (*Top View*) and the angled camera (*Angled View*). For the real-robot data, we only provide qualitative results as the ground-truth location of the objects is unknown. We set the expected number of keypoints $K = 6$ for the simulation and $K = 14$ for the real robot.

5 RESULTS

This section is organized to answer a series of questions to thoroughly evaluate our model and justify the choices we made in formulating our approach.

5.1 DOES THE MODEL ACCURATELY LEARN A FORWARD MODEL?

First, we evaluate the forward prediction accuracy. Figure 2 showcases qualitative results of forward predictions based on real-robot data. Our model factorizes the observation into a keypoint representation and accurately estimates the future appearance of the scene conditioned on external action. These qualitative results also highlight that our model learns to predict the effect of the action on objects as well as object-to-object interactions (see Appendix A.3 for more examples).

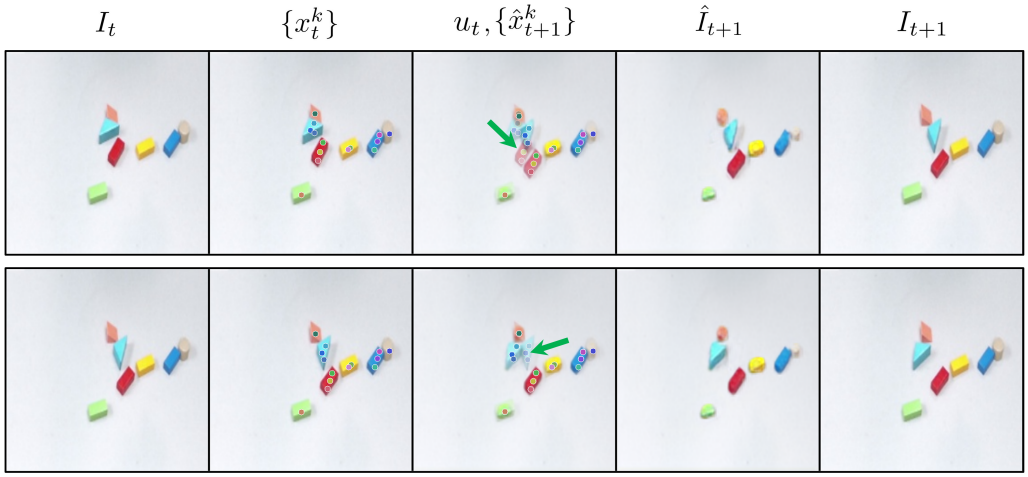

Figure 2: Qualitative results of real robot dataset Ye et al. (2020). Given an image (I_t), our model factorizes the scene into keypoints (x_t^k) and conditioned on the action (u_t, green arrows) estimates the next keypoint coordinates ($\hat{x}_{t+1}^k$) and appearance ($\hat{I}_{t+1}$) of the scene.

Table 1: Forward Prediction performance measured in position error for a single-step prediction. The prediction errors are computed for models separately trained on *Top* and *Angled View* images.

		Mean Position Error $\times 10^{-3}$	
Model	Supervision	*Top View*	*Angled View*
Forw	Img	0.309 ± 0.12	0.317 ± 0.09
ForwInv	Img	0.293 ± 0.08	0.266 ± 0.02
IN	GT state	0.112 ± 0.003	0.109 ± 0.008
VisIN	GT state + Img	0.107 ± 0.006	0.121 ± 0.03
KINet (Ours)	Img	0.122 ± 0.01	0.129 ± 0.02
KINet - deterministic	Img	0.127 ± 0.03	0.133 ± 0.01
KINet - no ctr loss	Img	0.173 ± 0.02	0.169 ± 0.05

We quantify the effectiveness of our model in comparison with Forw, ForwInv, IN, and VisIN baseline performances (Table 1) on the simulated data. We separately train and examine each model on *Top View* and *Angled View* images. The prediction error is computed as the average distance between the predicted and ground-truth positional states. VisIN performs best among baseline models as it builds object representations with explicit ground-truth object positions and their visual features. Our model, on the other hand, achieves a comparable performance to VisIN while it does not rely on any supervision beyond the scene images. Forw and ForwInv baselines have similar supervision to ours but are significantly less accurate. This emphasizes the capability of our approach to learn a rich graph representation of the scene and an accurate forward model while relaxing prevailing assumptions of the prior work on the structure of the environment and availability of ground-truth state information (see Appendix A.2 for more insight into the learned graph representations).

5.2 CAN WE USE THE MODEL IN CONTROL TASKS?

To further compare our approach with baseline methods, we examine a robotic manipulation task of bringing a set of objects to the desired goal state using MPC. For all models, we run 1K trials where at each trial object geometries and initial pose are randomized and a random goal configuration of objects is also generated. The planning horizon is set to $T = 40$ timesteps. For our model,

we perform GraphMPC based on graph embedding similarity as described in Section 3.6. For all baseline models, we perform MPC directly based on the distance to the goal. Figure 3 shows MPC results over the planning horizon based on *Top View* observations. Our approach is consistently faster than the baseline models in reaching the goal configuration. Additionally, we compute the pixel-wise distance to the goal image to evaluate whether the planned actions reach the goal configuration in terms of object appearances (i.e., orientation).

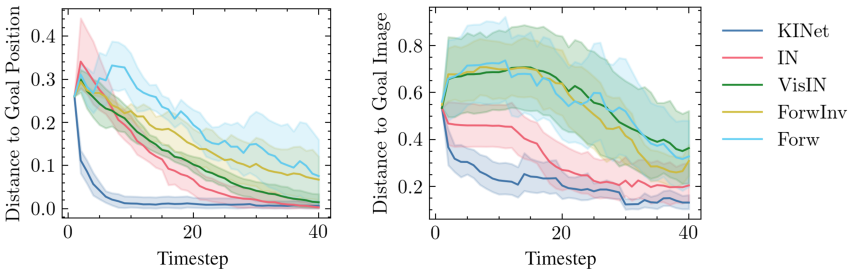

Figure 3: MPC comparison with baseline models for *Top View* images. Distance to goal measured based on (a) position and (b) pixel errors.

5.3 Does the model generalize to unseen circumstances?

One of our main motivations to learn a forward model in keypoint space is to eliminate the dependency of the model formulation on the number of objects in the system. We test for zero-shot generalization to a different number of objects (1, 2, 4, and 5 objects) and unseen object geometries (geom_{gen}). Figure 5 shows qualitative generalization results (see Appendix A.4 for more details). We separately train and examine for generalization on *Top View* and *Angled View* observations. Note that baseline models, with the exception of *Forw*, are not able to generalize to a different number of objects because this number is hard-coded in their formulation (e.g., in the number of feature channels or the number of graph nodes). We set the planning horizon to $T = 80$. Since our model learns to perform forward modeling in the keypoint space, with zero-shot generalization, it reassigns the keypoints to unseen objects and then makes forward predictions. We also quantitatively evaluate the quality of generalization scenarios (Table 2). As expected, by increasing the number of objects the average distance to the goal position increases. Also, objects with geometries that were sampled out of the training dimension range have more distance to the goal position. Our model's generalization performance is significantly superior to Forw baseline which is a simple action-conditioned video prediction model that uses the same supervision as our model.

To further assess the generalizability, we test for the performance of the model on unseen background textures (i.e., the table texture). Our model is trained on 3 objects placed on a white-colored table (domain_{train}). We test for zero-shot generalization to randomized table textures (domain_{gen}). Figure 6) shows qualitative results of the control task performed in the unseen background textures. Since the keypoint extraction step relies on visual features of salient objects, our model successfully performs the control tasks by ignoring the background and appropriately assigning keypoints to the objects. In extreme out-of-distribution cases such as the dirt texture (Fig 6, last row), a few keypoints are assigned to specific parts of the background. We speculate that this is because the unseen texture, unlike the training set, has rich visual features (see Fig. A.5.3 for failed cases). Figure 4 compares the MPC results for the unseen randomized texture with the simple background. As expected, although the MPC converges, the final distance to the goal configuration is larger for generalization to unseen textures.

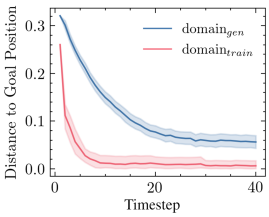

Figure 4: MPC results for generalization to unseen background texture.

5.4 Analysis and Ablation

We justify the major choices we made to formulate the model through ablation studies. Specifically, we examine two elements in our approach: the probabilistic graph representation, and the contrastive

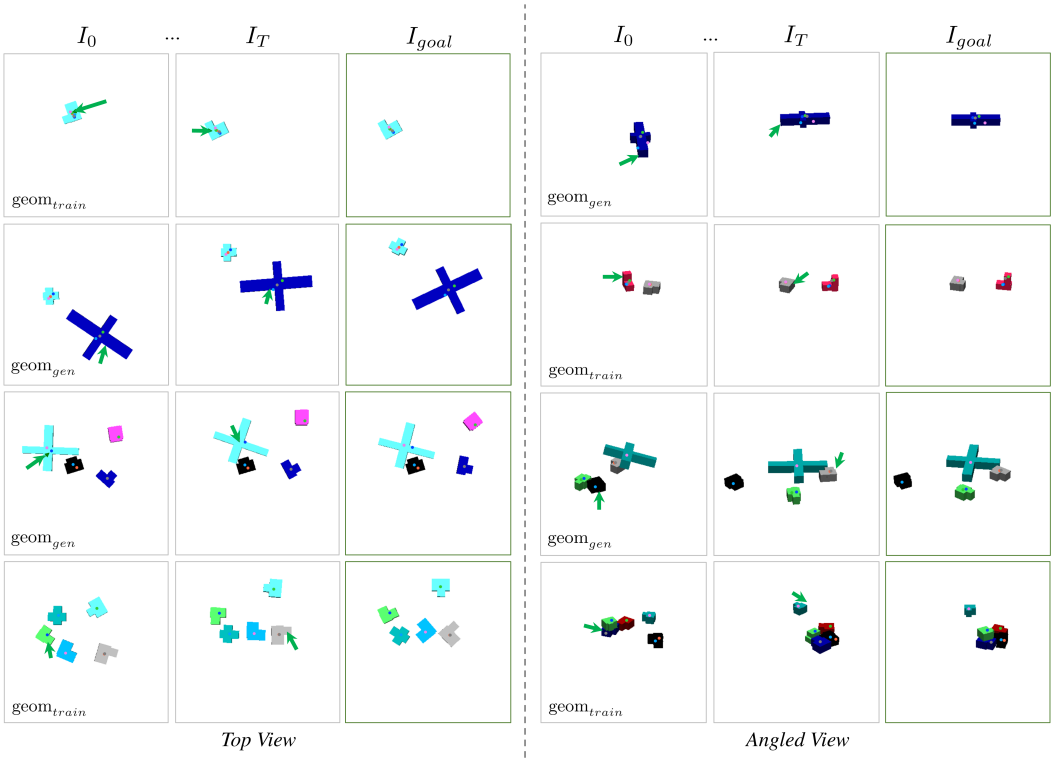

Figure 5: Qualitative results of generalization to different number of objects and unseen geometries for *Top* and *Angled View* observations. The green arrows are the optimal actions.

Table 2: Generalization results measured by the average distance to the goal position for a different number of objects and unseen object geometries in *Top* and *Angled View* images. Note that all models have been trained on 3 objects with geom_{train}.

		Distance to Goal Position $\times 10^{-3}$					
		KINet (Ours)		KINet - deterministic		Forw	
	Objects	geom_{train}	geom_{gen}	geom_{train}	geom_{gen}	geom_{train}	geom_{gen}
Top View	1	0.24±0.02	0.31±0.01	0.25±0.02	0.34±0.05	1.18±0.12	1.13±0.07
	2	0.22±0.01	0.58±0.02	0.26±0.02	0.65±0.03	1.07±0.08	1.02±0.05
	3	0.18±0.03	0.21±0.01	0.19±0.06	0.28±0.08	0.65±0.13	0.89±0.11
	4	0.54±0.01	0.63±0.13	0.68±0.09	0.89±0.11	1.87±0.15	1.96±0.13
	5	0.86±0.08	1.73±0.16	0.94±0.06	2.01±0.14	1.95±0.12	3.12±0.10
Angled View	1	0.21±0.04	0.35±0.04	0.28±0.01	0.36±0.08	1.09±0.08	1.34±0.07
	2	0.21±0.03	0.53±0.06	0.22±0.05	0.59±0.07	1.12±0.12	1.27±0.09
	3	0.19±0.02	0.20±0.05	0.19±0.03	0.31±0.08	0.98±0.11	1.16±0.09
	4	0.51±0.02	0.65±0.07	0.57±0.12	0.96±0.09	1.33±0.15	2.25±0.13
	5	0.89±0.13	1.64±0.11	1.05±0.16	2.17±0.10	1.48±0.12	3.20±0.19

loss. We train two variants of KINet: (1) *KINet - deterministic* in which the graph representation is fully connected and not probabilistic and there's no inference on the graph adjacency matrix. (2) *KINet - no ctr loss* where we train the model without the contrastive loss.

The best forward prediction performance for both *Top View* and *Angled View* observations is achieved when the model is probabilistic and trained with a contrastive loss (Table 1). The contrastive loss is an essential element in our approach to ensure the learned forward model is accurately action conditional. Also, a probabilistic graph representation allows for more efficient information aggregation of neighboring nodes. With a probabilistic graph representation, our model achieves higher generalization performance compared to its deterministic variant. This performance gap is more evident when generalizing to unseen geometries (Table 2).

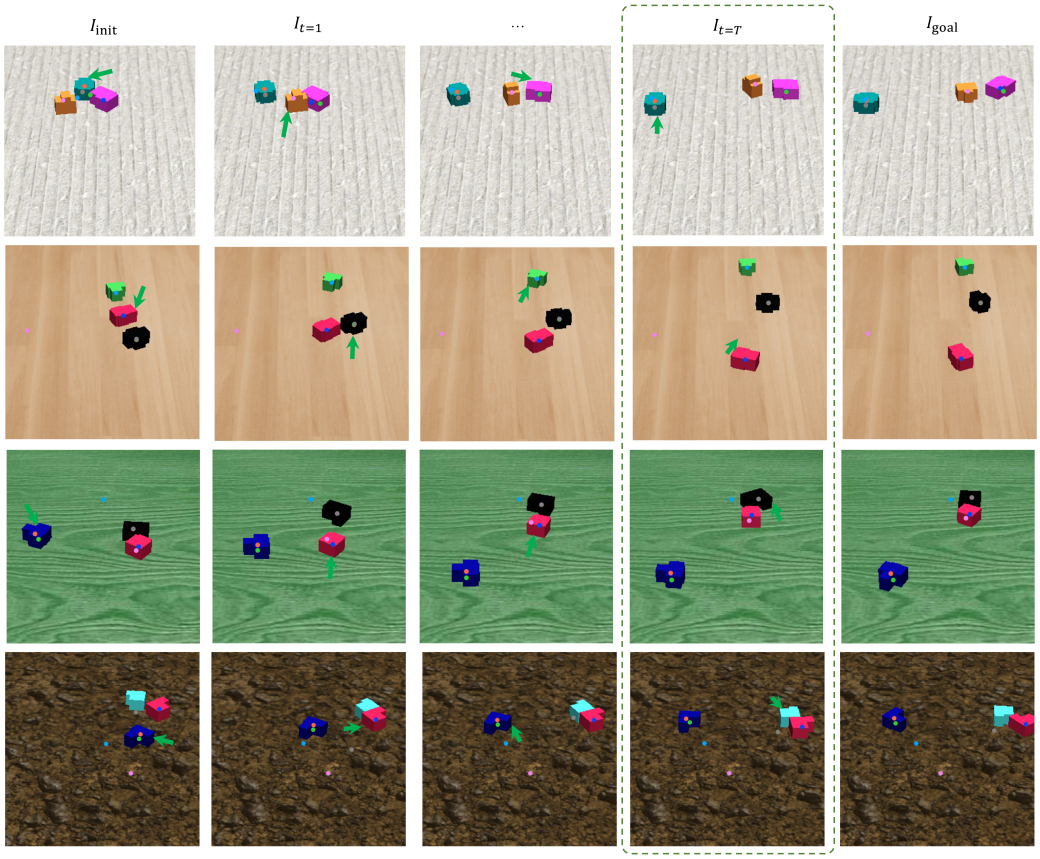

Figure 6: Qualitative results of generalization to unseen background. The green arrows are the optimal actions.

6 Conclusion

Summary. In this paper, we proposed a method for learning action-conditioned forward models based only on image observations. We showed that our approach effectively makes forward predictions through keypoint factorization. Also, we demonstrated that a keypoint-based forward model, unlike prior work, does not make assumptions about the number of objects which allows for automatic generalization to a variety of unseen circumstances. Importantly, our model learns a forward model without explicit supervision on ground-truth object state information.

Limitations and future work. One limitation in our formulation is fixing the number of expected keypoints; however, we showed this gives more generalizability compared to fixing the number of objects. We also observed inconsistency in the predicted keypoints for real-robot data scenarios where most of the objects were pushed out of the image frame (see Appendix A.5). An interesting future direction is to focus on the keypoint extraction module to further enhance forward models for real settings. Finally, we hope our general approach inspires future research on physical reasoning in settings where ground-truth information is hard to obtain.

REFERENCES

Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *arXiv preprint arXiv:1606.07419*, 2016.

Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *arXiv preprint arXiv:1612.00222*, 2016.

Boyuan Chen, Pieter Abbeel, and Deepak Pathak. Unsupervised learning of visual 3d keypoints for control. *arXiv preprint arXiv:2106.07643*, 2021a.

Zhenfang Chen, Jiayuan Mao, Jiajun Wu, Kwan-Yee Kenneth Wong, Joshua B Tenenbaum, and Chuang Gan. Grounding physical concepts of objects and events through dynamic visual reasoning. *arXiv preprint arXiv:2103.16564*, 2021b.

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

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Radek Grzeszczuk, Demetri Terzopoulos, and Geoffrey Hinton. Neuroanimator: Fast neural network emulation and control of physics-based models. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 9–20, 1998.

Mikael Henaff, Alfredo Canziani, and Yann LeCun. Model-predictive policy learning with uncertainty regularization for driving in dense traffic. *arXiv preprint arXiv:1901.02705*, 2019.

Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object landmarks through conditional image generation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 4020–4031, 2018.

Michael Janner, Sergey Levine, William T Freeman, Joshua B Tenenbaum, Chelsea Finn, and Jiajun Wu. Reasoning about physical interactions with object-oriented prediction and planning. *arXiv preprint arXiv:1812.10972*, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pp. 2688–2697. PMLR, 2018.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Tejas Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *arXiv preprint arXiv:1906.11883*, 2019.

Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B. Tenenbaum, Antonio Torralba, and Russ Tedrake. Propagation networks for model-based control under partial observation. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:1205–1211, 9 2018. URL `http://arxiv.org/abs/1809.11169`.

Yunzhu Li, Jiajun Wu, Jun-Yan Zhu, Joshua B Tenenbaum, Antonio Torralba, and Russ Tedrake. Propagation networks for model-based control under partial observation. In *ICRA*, 2019.

Yunzhu Li, Toru Lin, Kexin Yi, Daniel M. Bear, Daniel L. K. Yamins, Jiajun Wu, Joshua B. Tenenbaum, and Antonio Torralba. Visual grounding of learned physical models. *37th International Conference on Machine Learning, ICML 2020*, PartF168147-8:5883–5892, 4 2020. URL http://arxiv.org/abs/2004.13664.

Lucas Manuelli, Yunzhu Li, Pete Florence, and Russ Tedrake. Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning. *arXiv preprint arXiv:2009.05085*, 2020.

R Chris Miall and Daniel M Wolpert. Forward models for physiological motor control. *Neural networks*, 9(8):1265–1279, 1996.

Matthias Minderer, Chen Sun, Ruben Villegas, Forrester Cole, Kevin Murphy, and Honglak Lee. Unsupervised learning of object structure and dynamics from videos. *arXiv preprint arXiv:1906.07889*, 2019.

Damian Mrowca, Chengxu Zhuang, Elias Wang, Nick Haber, Li Fei-Fei, Joshua B Tenenbaum, and Daniel LK Yamins. Flexible neural representation for physics prediction. *arXiv preprint arXiv:1806.08047*, 2018.

Derrick Nguyen and Bernard Widrow. The truck backer-upper: An example of self-learning in neural networks. In *Advanced neural computers*, pp. 11–19. Elsevier, 1990.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.

Haozhi Qi, Xiaolong Wang, Deepak Pathak, Yi Ma, and Jitendra Malik. Learning long-term visual dynamics with region proposal interaction networks. *arXiv preprint arXiv:2008.02265*, 2020.

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 *International Conference on Machine Learning*, pp. 4470–4479. PMLR, 2018.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International conference on artificial neural networks*, pp. 412–422. Springer, 2018.

Aleksandar Stanić, Sjoerd van Steenkiste, and Jürgen Schmidhuber. Hierarchical relational inference. *arXiv preprint arXiv:2010.03635*, 2020.

Supasorn Suwajanakorn, Noah Snavely, Jonathan J Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *Advances in neural information processing systems*, 31, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. *Advances in neural information processing systems*, 30:4539–4547, 2017.

Nicholas Watters, Loic Matthey, Matko Bosnjak, Christopher P Burgess, and Alexander Lerchner. Cobra: Data-efficient model-based rl through unsupervised object discovery and curiosity-driven exploration. *arXiv preprint arXiv:1905.09275*, 2019.

Daniel M Wolpert and Mitsuo Kawato. Multiple paired forward and inverse models for motor control. *Neural networks*, 11(7-8):1317–1329, 1998.

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

Daniel M Wolpert, R Chris Miall, and Mitsuo Kawato. Internal models in the cerebellum. *Trends in cognitive sciences*, 2(9):338–347, 1998.

Tianfan Xue, Jiajun Wu, Katherine L Bouman, and William T Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. *arXiv preprint arXiv:1607.02586*, 2016.

Wilson Yan, Ashwin Vangipuram, Pieter Abbeel, and Lerrel Pinto. Learning predictive representations for deformable objects using contrastive estimation. *arXiv preprint arXiv:2003.05436*, 2020.

Yuan Yao, Yasamin Jafarian, and Hyun Soo Park. Monet: Multiview semi-supervised keypoint detection via epipolar divergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 753–762, 2019.

Yufei Ye, Dhiraj Gandhi, Abhinav Gupta, and Shubham Tulsiani. Object-centric forward modeling for model predictive control. In *Conference on Robot Learning*, pp. 100–109. PMLR, 2020.

Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2694–2703, 2018.