Dynamical Scene Representation and Control with Keypoint-Conditioned Neural Radiance Field

Weiyao Wang[1], Andrew S. Morgan[2], Aaron M. Dollar[2], and Gregory D. Hager[1]

***Abstract*— In this work, we present a method that can learn to model dynamic and arbitrary 3D scenes, purely from 2D visual observations. Our approach uses a keypoint-conditioned Neural Radiance Field (KP-NeRF) to capture and model these scenes with the overarching goal of supporting image-based robot manipulation. Differentiating this from previous methods, which typically condition the model on generic embedding vectors for representation, our implicit neural radiance function is conditioned on a set of keypoints that are inferred from a learned encoder given imagery observations. This implicitly separates the visual modeling components into object appearances and object pose configurations. Such inductive bias built into the architecture encourages discovered keypoints to capture state transitions in the robot's environment across time and space. We then learn a forward prediction model of the encoded keypoints, constructed over the keypoint representation space, and perform MPC control for challenging manipulation tasks including block pushing and door closing. We evaluate the performance of our method through various tasks: novel scene view synthesis, action-conditioned forward prediction, and robot manipulation tasks.**

I. Introduction

Visual-based control of a robot for manipulation has been a question studied in the literature for decades, but remains largely unsolved and is still open today. From a general manipulation standpoint, a vast majority of previous work has assumed there exists an underlying model for the dynamics associated with the robot and its environment [1]. However, these models are increasingly difficult or even impossible to derive analytically in many cases, as tasks get more complicated and environments become less structured [2], [3]. A way to alleviate such restraint is to learn and continuously infer the dynamics associated with contact between a robot and its environment, e.g. in Model-Based Reinforcement Learning approaches [4], and one promising approach is through vision [5]. In this work, we are interested in developing methods that are able to learn a representation and its dynamics purely through 2D visual images of a task — this opens the door to being able to work with systems where the dynamics are unknown and cannot be analytically derived.

One common approach is to use the 2D image itself as such representation. One can learn dynamics models and make future predictions directly for the image's pixels [6], [7], [8]. Planning will then be carried out using the cost of distances measured in image space. This approach, however, suffers from high modeling error given the difficulty in 2D image prediction itself. Another approach is to learn a reduced dimensional representation, either as an embedding or as keypoints, and then perform control tasks [9], [10], [11], [12], [13]. This approach usually also performs 2D image reconstruction and prediction as auxiliary tasks to obtain meaningful representations. These two types of approaches, although mostly promising, suffer from an inherent limitation which is the lack of 3D-awareness in the scene. When projected from 3D space to a 2D image, even simple scene element movement can create complex appearance changes due to relative viewing angle shift and occlusion. This difficulty in visual modeling leads to blurry imagery predictions and less effective representations, which are especially required for manipulation.

There have been recent advances in learning implicit 3D representation of scenes [14], [15], [16]. Neural Radiance Field (NeRF) [17] and its extensions [18], [19] can learn 3D structures and appearances of particular scenes trained solely on a set of 2D images from different views. Followup work, PixelNeRF [20], extends NeRF to generalize across multiple scenes by conditioning on image features from one or multiple observed images. [5] combines NeRF and contrastive learning to learn an image embedding function which is then applied to control. A-NeRF [21] uses articulated skeleton to model 3D human body and refine 3D pose estimation. Contrary to these previous works, our work proposes to combine Neural Radiance Field with keypoint representation for control. The intuition behind this method is that in most robotic manipulation scenarios, scenes are dynamic since the state of the world is always changing, but the objects being manipulated typically remain the same. It would thus be desirable to have the appearance modeling and configuration estimation be separated. Such disentanglement would improve the visual modeling quality (e.g., lower view synthesis error) because the locally conditioned appearance model will be easier to learn, and also provide for a more straightforward distance metric when performing Model Predictive Control (MPC) [22], [23].

To this end, we devise a keypoint-conditioned Neural Radiance Field (KP-NeRF). Given an input image from a fixed camera view, we first predict keypoint locations and depth in the image space and then unproject to 3D space. Thereafter, for each point sampled in 3D space as in [24], we additionally compute the position with respect to each

This work was supported by the U.S. National Science Foundation grants IIS-1900952 to Johns Hopkins University and grants IIS-1752134, IIS-1900681 to Yale University

[1]W. Wang and G. D. Hager are with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA {wwang121,hager} @ cs.jhu.edu

[2]A. Morgan and A. Dollar are with the Department of Mechanical Engineering and Materials Science, Yale University, New Haven, CT 06520, USA {andrew.morgan, aaron.dollar} @ yale.edu

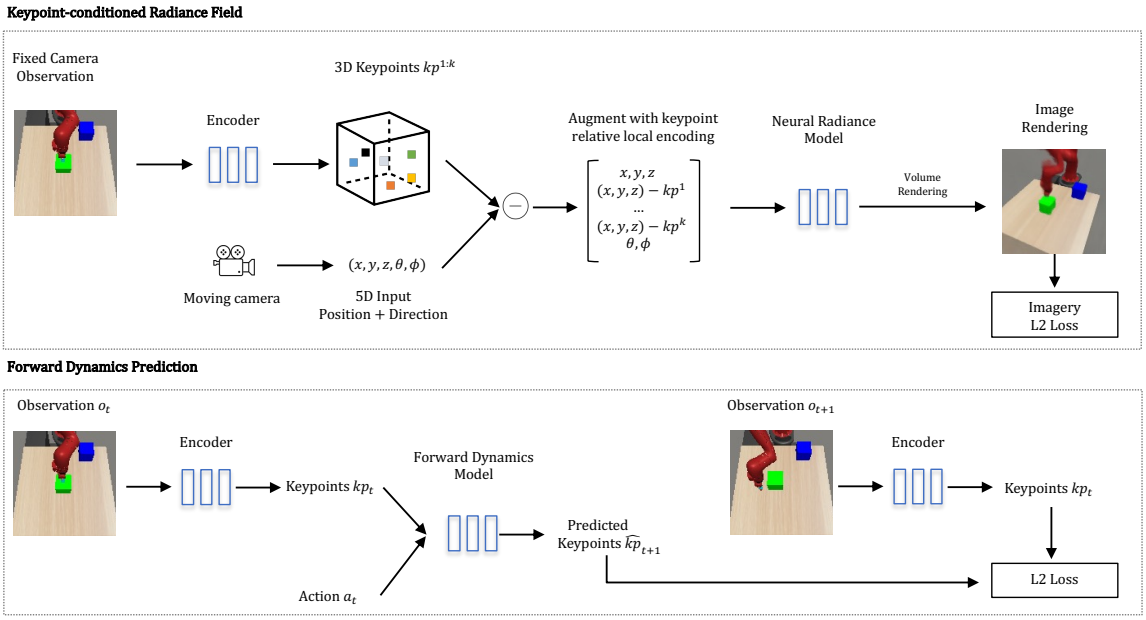


Fig. 1. Our method first learn a keypoint-conditioned neural radiance field and then train a forward dynamics prediction based on the discovered keypoints. In the first step, we augment the original 5D input with the keypoint relative local encoding as context for the model to generalized along a dynamical scene. In the second step, we learn an action-conditioned forward prediction model of the transition dynamics of keypoints.

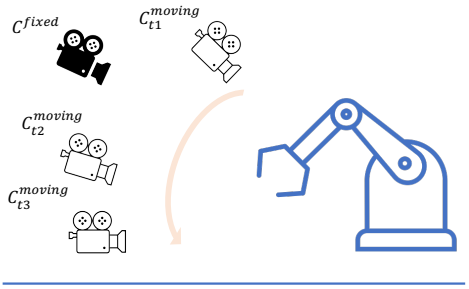


Fig. 2. Static camera only provides a 2D view of a robotic manipulation scene. Additional data from moving cameras provide invaluable 3D context information for training purposes. During testing, though, only fixed cameras are used as we can now rely on the detected 3D aware keypoints.

keypoint. These local positions are then concatenated to the original 5D input (3D position + 2D view direction) to feed into a MLP to output radiance and opacity for that point. The neural radiance function is now conditioned on the point's relative location to all the keypoints so that the local appearance and structure is invariant when the keypoints move within a dynamical scene. By imposing this inductive bias, the network is encouraged to predict keypoints that attach to dynamical elements of the scene so as to exploit the local invariance of the object's appearance. As shown in Figure 3, our method discovers keypoints that are relatively "attached" to objects in the scene, even when no explicit label information is provided.

Given our learned KP-NeRF, we subsequently train a dynamics model that predicts keypoint movements conditioned on input action. This dynamics model can be used to forward propagate keypoint poses, which enables action conditioned scene predictions from arbitrary views and model-based control schemas, such as MPC. Through comparison with baseline methods, we find our method, KP-NeRF, produces higher quality scene modeling in both, same timestep view synthesis and forward predictive view synthesis settings. The learned keypoint representation provides a more well-constructed space, which in turn produces less control error and more accurate manipulation procedures.

II. Methodology

A. Preliminary

Neural Radience Field: Proposed initially in the context of novel view synthesis, NeRF [17] represents the scene as a continuous function F_{nerf} parameterized as multi-layer perception:

$$F_{nerf} : \mathbf{x}, \mathbf{d} \rightarrow \sigma, \mathbf{c} \tag{1}$$

which maps any given 3D scene coordinate $\mathbf{x} \in \mathcal{R}^3$ and view direction $\mathbf{d} \in \mathcal{S}^2$ to volumetric density $\sigma \in \mathcal{R}$ and radiance $\mathbf{c} \in \mathcal{R}^3$. The points in space are obtained by sampling depth s on the ray from camera origin $\mathbf{o}$ as $\mathbf{x} = \mathbf{r}(s) = \mathbf{o} + s\mathbf{d}$. Both $\mathbf{o}$ and d are derived from calibrated camera pose. The color $C(\mathbf{r})$ of a pixel that corresponds to ray $\mathbf{r}$ is then numerically estimated via an integral of accumulated radiance along the corresponding ray $\mathbf{r}$:

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \mathbf{c}_i, \tag{2}$$

$$\text{where } T_i = \exp\left(- \sum_{j=1}^{i-1} \sigma_j (s_{j+1} - s_j) \right) \tag{3}$$

$$\text{and } \alpha_i = 1 - \exp(-\sigma_i (s_{i+1} - s_i)). \tag{4}$$

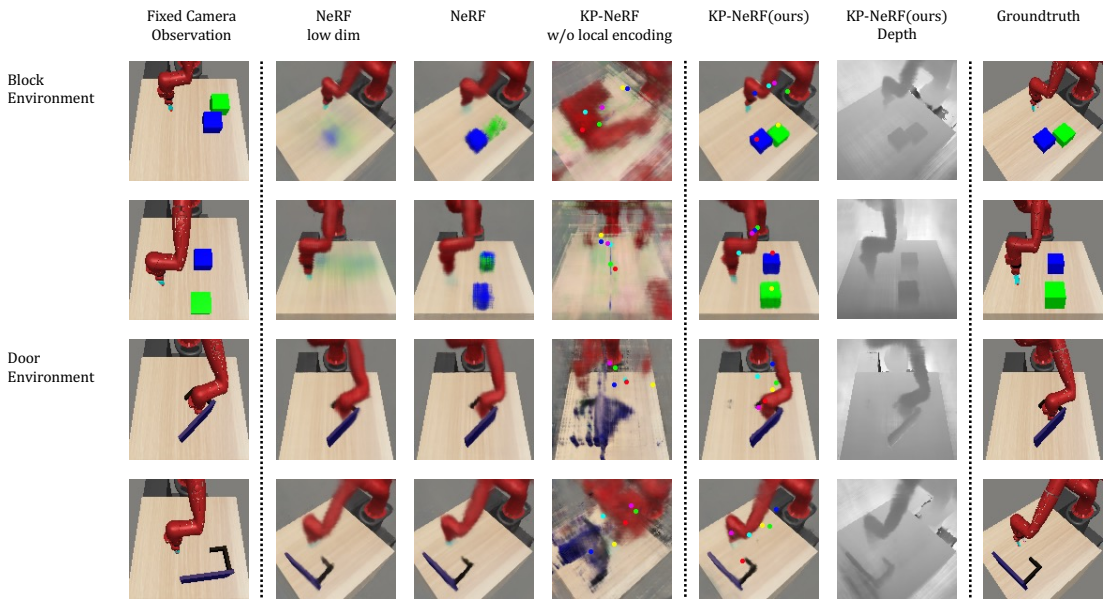


Fig. 3. Novel view synthesis in block and door environments. The models receive observation from the fixed camera and try to render images of another moving camera given its pose. Our method KP-NeRF produces accurate view synthesis with better details compare to baselines. Keypoints are overlayed on the rendered image for KP-NeRF and its variant for visualization.

Here, $s_{i=1}^{N}$ is a set of samples from near bound s_n to far bound s_f and $\sigma_i = \sigma(\mathbf{r}(s_i))$, $\mathbf{c}_i = \mathbf{c}(\mathbf{r}(s_i))$ are evaluations of volume density and radiance at sample points $\mathbf{r}(s_i)$ along the ray.

In training, F_{nerf} will be optimized to render a scene from multiple views given a set of images and camera pose pairs. During testing, the model can then render that given scene from a novel view corresponding to an arbitrary camera pose. A pixel-wise L2 distance norm is used as loss function:

$$\mathcal{L} = \sum_{\mathbf{r}} \left\| C(\mathbf{r}) - \hat{C}(\mathbf{r}) \right\|^2 \tag{5}$$

B. Dynamical Scene Representation

Keypoint-conditioned Neural Radiance Field: The original NeRF, while impressive in generating high quality view synthesis, lacks the ability to generalize across different scenes. The followup work [20] in image conditioning 3D scene modeling uses either local or global features from CNN to provide context, which does not fully leverage object invariance usually appearing in robot manipulation settings. We extend NeRF to generalize across a dynamical scene by conditioning on a set of recognized keypoints given an imagery observation. We use encoder F_{enc} to extract k keypoints $kp_{[1:k]} = F_{enc}(O)$ from observational image O.

In order to encourage the network to learn 3D aware representation that is invariant to keypoint movements, we compute the relative location of given point $\mathbf{x}$ with respect to each keypoint and provide that as context information for the neural radiance model. Different from generic embedding produced by CNN features, our representation as keypoints provides structure that supports object structure and appearance invariance with respect to keypoints. This keypoint relative local encoding is defined as the following:

$$F_{kp\_nerf} : \mathbf{x}, \mathbf{d}, \{\mathbf{x} - kp_i \mid i \in [1:k]\} \rightarrow \sigma, \mathbf{c} \tag{6}$$

It is also possible to simply provide keypoint locations as additional inputs without local encoding as defined as follow:

$$F_{kp\_nerf\_no\_local} : \mathbf{x}, \mathbf{d}, kp_{[1:k]} \rightarrow \sigma, \mathbf{c} \tag{7}$$

This variant however does not establish structural relationship between points in 3D and keypoint locations. The performances of above two variants will be then compared by experiments in Section III. A-NeRF [21] also uses relative encoding that is in spirit similar to ours. However their relative encoding is with respect to bones in prespecified skeleton, and they focus on the human body modeling which is different from our settings.

Keypoint Encoder: Inspired by previous works in unsupervised keypoint detection [25], [12], we choose to parameterize keypoint locations by heatmap and depth map. The keypoint encoder F_{enc} is implemented with a fully convolutional neural network backbone. We pass observational image O into the backbone network to obtain k feature maps $F_{[1,\ldots,k]}$. For each of the feature maps F_i we then apply spatial softmax to produce a probability heatmap H_i that represents the probability that a corresponding keypoint appears in certain location in that given image. Additionally, we also have network output a depth map $D_{[1,\ldots,k]}$ representing the depth from that camera origin to the keypoints. The final location of a keypoint is then computed as the expected position in the heatmap with expected depth from the depth map using the heatmap as probabilities.

C. Model Predictive Control with Discovered Keypoints

Given a dynamics model, either learned or analytical, Model Predictive Control (MPC) solves for the minimum

cost, i.e. optimal, trajectory over a receding time horizon. Unfortunately, this optimization at each timestep can be prohibitively expensive for most systems to do online, especially in high dimensional spaces. Various fast optimization procedures have been introduced in the literature: from simple hill climbing approaches for straightforward systems [26] to high dimensional importance sampling methods for complex, highly dynamic systems [27], [28]. Using methods from this literature, we decide to leverage a cross entropy optimization approach to solve our MPC problem and control keypoints according to a learned dynamics model.

Once we have obtained the keypoint encoder F_{enc}, we can use supervised learning to estimate the forward dynamics model, $\hat{kp}_{t+1} = F_{dyn}(kp_t, a_t)$. We predict H steps in the future by iteratively feeding actions into the one-step forward model. We implement F_{dyn} as an MLP network which is trained by optimizing the following loss function:

$$\mathcal{L}_{dyn} = \sum_{h=1}^{H} \|\hat{kp}_{t+h} - kp_{t+h}\|^2, \tag{8}$$

where $\hat{kp}_{t+h} = F_{dyn}(kp_{t+h-1}, a_{t+h-1})$, $\hat{kp}_t = kp_t$.

Once F_{dyn} is learned, we use an importance sampling version of MPC to find the optimal next-state action as to minimize the differences between current keypoint positions and goal keypoint positions kp_g. This is done iteratively after each actuation step and relies on the cross entropy method to optimize the following cost function:

$$C(\hat{kp}_t, kp_g) = \left\|kp_g - \hat{kp}_t\right\|_2^2 \tag{9}$$

III. Experiments

Our primary experimental domain is with simulated tabletop manipulation task built off of the Meta-World suite of environments [29]. Specifically, it consists of a simulated Sawyer robot, and two blocks or a door on a tabletop. The setup of environment is adapted from [30]. As illustrated in Figure. 2, the agent receives pixel image from a fixed posed camera and an additional pixel image from a camera moving in space. In this section, we are going to compare our approach to baseline methods on the following three tasks: novel view synthesis, action-conditioned video prediction, and image-based robot manipulation.

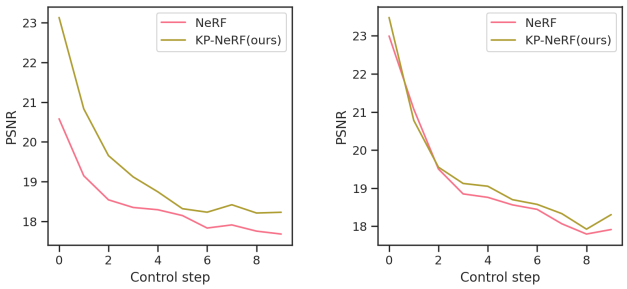
TABLE I

Quantitative comparison in novel view synthesis by PSNR (the higher the better) and MSE (the lower the better).

	Block		Door	
	PSNR↑	MSE↓	PSNR↑	MSE↓
NeRF low dim	18.30	0.0154	23.00	0.0050
NeRF	20.42	0.0095	23.19	0.0050
KP-NeRF w/o local encoding	12.86	0.0524	12.99	0.0512
KP-NeRF (ours)	**22.85**	**0.0054**	**23.65**	**0.0045**

A. Novel View synthesis

We first evaluate the model's ability to encode 3D-aware information by measuring the novel view synthesis quality.



(a) Block (b) Door

Fig. 4. Accuracy of the forward predicted frames measured in PSNR. Left: block environment; Right: door environment. X-axis is the number of control steps used to predict. Our method produces higher PSNR in predicted frames, especially in the block environment.

The agent executes a random policy for 200 episodes to collecting 10,000 pair of image frames from a fixed camera and a moving camera of another pose. Aside from our method, KP-NeRF, and KP-NeRF with no local encoding variant, we also have NeRF as baseline, in which we directly adapt the original NeRF by using a global embedding vector as context information. The global embedding vector is extracted from the fixed camera view by using a CNN encoder. For our method and its variant, we choose the total number of keypoints k to be 6 for both the blocks and environment. We have NeRF with a 128 dimensional latent embedding and also compare to a NeRF low dimensional version having an 18 dimensional embedding to align the degrees of freedom as in KP-NeRF. All models are trained to convergence for 100,000 iterations. We choose to utilize two commonly evaluated image reconstruction metrics, the peak signal to noise ratio (PSNR) and the mean squared error (MSE) to measure the performance.

Quantitative results comparing the methods are presented in Table I. We note that our method, KP-NeRF, outperforms the others in both, the block and the door environments. This can be further qualitatively validated by the graphical samples in Figure 3. From these results, both keypoints and local encodings are important to produce a high quality novel view synthesis in the tested dynamical scene. When conditioning on generic vector embedding, the model produces more blurry images compared to ours. In particular, when an embedding's degrees of freedom matches ours, it cannot capture the blocks due to the lack of expressiveness. The poor performance of the KP-NeRF variant without a local encoding, on the other hand, indicates that the structure of computing local locations to each keypoint brings crucial inductive bias for learning meaningful keypoints and rendering a model. As we can see from the visualization, the keypoints from our methods can approximately following dynamical elements in the scene. Note that such a pattern emerges in an unsupervised manner during training which is a result of the inductive bias we mentioned earlier.

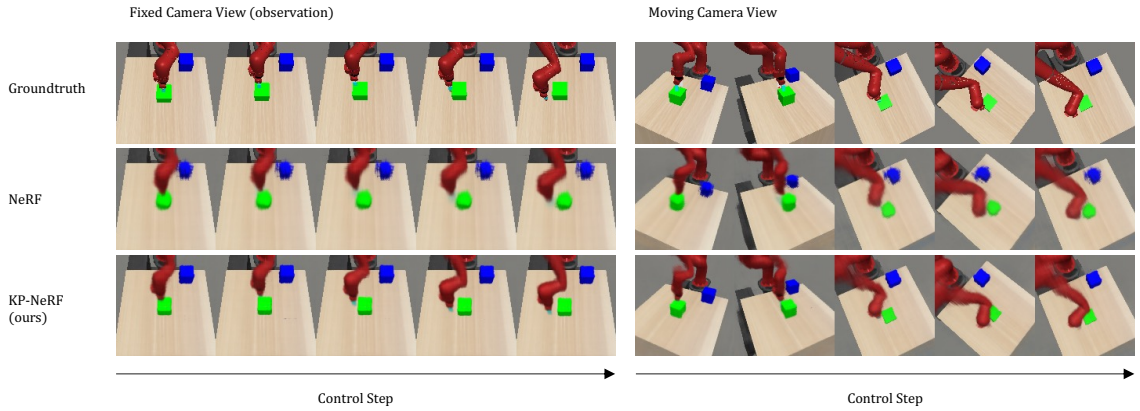


Fig. 5. Forward prediction visualization across control steps. Left: from the observational fixed camera view. Right: from the additional moving camera view. Our method generates more clear and less blurry prediction compared to the baseline, especially around the edge of objects.

B. Action-conditioned video prediction

Going beyond novel view synthesis, where the model receives an observation from the same timestep, we further test the model's performance when doing forward video prediction. Under this setting, the model trains an action conditioned forward prediction on the encoded vector embedding or keypoints, and then uses the original rendering function to synthesize images from predicted values. We use the learned encoding function from the previous subsection to extract keypoints or embeddings for a 50,000 frame dataset collected by executing a random policy with only one fixed observational camera view. We have more single-view data to train the forward dynamics model than multi-view data to train the keypoint-conditioned rendering model, where single-view data is easier to obtain than multi-view data. Therefore, we can train the forward dynamics model using a larger but easier to acquire dataset once the keypoint encoder is learned. The forward dynamics prediction model is implemented as a MLP with 4 hidden layers of 512 width. Since the previous subsection illustrates that neither a low dimensional vector embedding nor keypoints without local encodings work well, we only compare our method against NeRF with a 128 dimensional embedding vector.

In Figure 4, we can see that our method produces higher PSNR when performing forward predictions, and is particularly more evident in the block environment. The block environment has one more object moving across the space under which our keypoint-based representation is easier to model and generalize due the inductive bias we introduce through our keypoint relative local encoding. The forward prediction quality can be visualized in Figure 5. Our method generates sharp object appearances across steps, while the baseline is blurry–especially around the blocks.

C. Image-based robot manipulation

After showing our performance in same timestep view synthesis and forward prediction, we apply the learned dynamics model in image-based robot manipulation tasks: block pushing and door closing. The block pushing task is

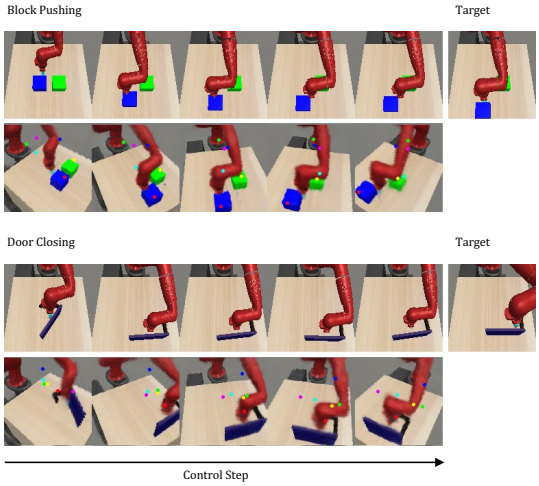


Fig. 6. A sample trial for each of the manipulation tasks. For each task, the first row is the rollout observation and the second row is the detected keypoints overlaid on a rendered viewed from a different camera pose. The last column is the goal image to reach.

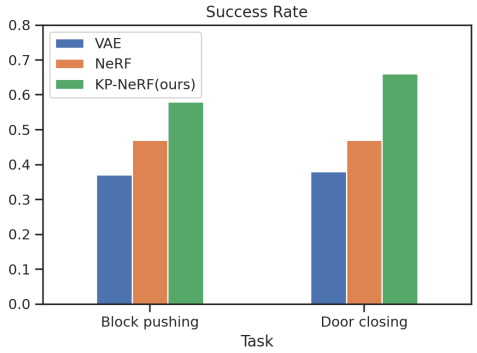


Fig. 7. Average success rate for each task evaluated over 100 trials with 40 control steps. Our approach outperforms the baseline methods in both the block pushing task and the door closing task.

defined as pushing the target block to the goal position with distance less than 0.1. The door closing task is to close the door to be within 0.1 radians from the goal position. A view

of the target state is provided to specify the goal. Aside from KP-NeRF and NeRF, we also include the learned embedding through a variational autoencoder (VAE) as a 2D baseline to compare with. Subsequently, a dynamics model is learned with the loss function in Equation 8. MPC control is then applied with the learned dynamics function for all methods.

As shown in Figure 7, our method outperforms both NeRF and VAE in both block pushing and door closing in achieving higher success rate. This could be attributed to the fact that 1) our model produces better 3D-aware scene modeling as shown in previous subsections and 2) our keypoint-based representation serves as a more appropriate space to compute distance on when performing MPC control. A rollout for the task is presented in Figure 6.

IV. Discussions and Future Work

In this paper, we proposed a method to learn a 3D-aware keypoint representation for a dynamical scene. By introducing keypoint relative local encodings as inductive bias, the model was able to leverage object appearances invariant to relative keypoints to model complex dynamical scenes. We show that it is beneficial to learn such keypoint-conditioned neural radiance fields that produce superior performance among tasks including view synthesis, video prediction, and robot manipulation.

Meanwhile, there are several limitations that exist currently in this presented work. First, our experiments, while showing promising results, were only conducted in simulated tabletop environments. In the future, we are interested in expanding this to real-world robot scenarios, particularly for in-hand manipulation. Moreover, we find that there is still room of improvement in binding keypoints to consistent position of dynamical elements in the scene. It would be desirable to further investigate methods to enforce more consistent constraints that regularizes the keypoint locations towards that direction.

References

[1] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*. USA: CRC Press, Inc., 1994.

[2] A. M. Dollar and R. D. Howe, "The highly adaptive sdm hand: Design and performance evaluation," *The international journal of robotics research*, vol. 29, no. 5, pp. 585–597, 2010.

[3] A. S. Morgan*, B. Wen*, J. Liang, A. Boularias, A. Dollar, and K. Bekris, "Vision-driven compliant manipulation for reliable; high-precision assembly tasks," *Robotics: Science and Systems XVII*, Jul 2021.

[4] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, 2017.

[5] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3d neural scene representations for visuomotor control," in *Conference on Robot Learning*. PMLR, 2022, pp. 112–123.

[6] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," *Advances in neural information processing systems*, vol. 29, 2016.

[7] L. Yen-Chen, M. Bauza, and P. Isola, "Experience-embedded visual foresight," in *Conference on Robot Learning*. PMLR, 2020, pp. 1015–1024.

[8] H. Suh and R. Tedrake, "The surprising effectiveness of linear models for visual foresight in object pile manipulation," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2020, pp. 347–363.

[9] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[10] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller, "Embed to control: A locally linear latent dynamics model for control from raw images," 2015.

[11] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2555–2565.

[12] B. Chen, P. Abbeel, and D. Pathak, "Unsupervised learning of visual 3d keypoints for control," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1539–1549.

[13] W. Wang, M. Kobilarov, and G. D. Hager, "Learn proportional derivative controllable latent space from pixels," *arXiv preprint arXiv:2110.08239*, 2021.

[14] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: Learning dynamic renderable volumes from images," *arXiv preprint arXiv:1906.07751*, 2019.

[15] H.-Y. F. Tung, R. Cheng, and K. Fragkiadaki, "Learning spatial common sense with geometry-aware recurrent networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2595–2603.

[16] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174.

[17] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *European conference on computer vision*. Springer, 2020, pp. 405–421.

[18] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.

[19] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.

[20] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.

[21] S.-Y. Su, F. Yu, M. Zollhöfer, and H. Rhodin, "A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose," in *Advances in Neural Information Processing Systems*, 2021.

[22] M. Morari and J. H. Lee, "Model predictive control: past, present and future," *Computers & Chemical Engineering*, vol. 23, no. 4-5, pp. 667–682, 1999.

[23] J. B. Rawlings, "Tutorial overview of model predictive control," *IEEE control systems magazine*, vol. 20, no. 3, pp. 38–52, 2000.

[24] L. Yen-Chen, P. Florence, J. T. Barron, A. Rodriguez, P. Isola, and T.-Y. Lin, "inerf: Inverting neural radiance fields for pose estimation," *arXiv preprint arXiv:2012.05877*, 2020.

[25] T. Jakab, A. Gupta, H. Bilen, and A. Vedaldi, "Unsupervised learning of object landmarks through conditional image generation," *Advances in neural information processing systems*, vol. 31, 2018.

[26] A. S. Morgan, K. Hang, and A. M. Dollar, "Object-agnostic dexterous manipulation of partially constrained trajectories," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5494–5501, 2020.

[27] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1714–1721.

[28] A. Lambert, A. Fishman, D. Fox, B. Boots, and F. Ramos, "Stein variational model predictive control," *arXiv preprint arXiv:2011.07641*, 2020.

[29] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2019. [Online]. Available: https://arxiv.org/abs/1910.10897

[30] S. Nair, S. Savarese, and C. Finn, "Goal-aware prediction: Learning to model what matters," in *International Conference on Machine Learning*. PMLR, 2020, pp. 7207–7219.