Learning Multi-Object Dynamics with Compositional NeRFs

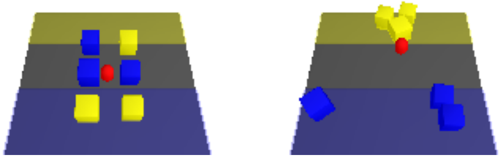
Danny Driess[1] Zhiao Huang[2] Yunzhu Li[3] Russ Tedrake[3] Marc Toussaint[1]

***Abstract*— We present a method to learn compositional predictive models from image observations based on implicit object encoders, Neural Radiance Fields (NeRFs), and graph neural networks. Most NeRF approaches are trained on a single scene, representing the whole scene with a global model. Instead, we present a compositional, object-centric auto-encoder framework that maps multiple views of the scene to a *set* of latent vectors representing each object separately. We train a graph neural network dynamics model in the latent space to achieve compositionality for dynamics prediction. A key feature is that the learned 3D information through the NeRF model enables us to incorporate structural priors in learning the dynamics models, making long-term predictions more stable. For planning, we utilize RRTs in the learned latent space, where we can exploit our model to make sampling the latent space informative and more efficient. Video: `https://dannydriess.github.io/compnerfdyn/`**

I. Introduction

A common approach to learn dynamic models from high-dimensional observations is to first map the observations into a lower-dimensional latent representation of the scene via an auto-encoder in order to then learn the dynamics in the latent space. While this method is applicable for a large variety of tasks, it raises multiple challenges. First, scenes in our world are *composed* of multiple objects. Therefore, a latent vector with a fixed size has difficulties in generalizing over different numbers of objects. Moreover, image observations are 2D, but the underlying physical processes happen in our 3D world. Many forward predictive models in visual observation spaces suffer from instabilities in making long-term predictions, often manifested in blurry image predictions [2]. One way to address these issues is to incorporate inductive biases in the model architectures. For instance, Li et al. [3] propose to use Neural Radiance Fields (NeRFs) [4] as a decoder within an auto-encoder for learning dynamics models in a latent space. NeRFs exhibit structural priors about the 3D world [5], [6], leading to increased performance over 2D baselines. However, the approach of [3] represents the whole scene as a single latent vector, which we found insufficient for scenes composed of multiple numbers of objects.

In the present work, we aim to overcome these challenges by incorporating inductive biases on the compositional nature of our world and its 3D structure both in learning the latent representations themselves and the dynamics model. We propose a compositional, object-centric auto-encoder framework whose latent vectors are used to learn a compositional forward dynamics model in that learned latent space based on graph neural networks (GNN). We learn an implicit object encoder that maps image observations of the scene from multiple views to a set of latent vectors that each represent an object in the scene separately. These latent object encodings parameterize NeRFs for each object. We apply compositional rendering techniques to synthesize images from multiple viewpoints, which forces the object-centric NeRF functions and the corresponding latent vectors to learn precise 3D configurations of the constituting objects. This 3D inductive bias both in the encoder and the compositional NeRF decoder enables us to incorporate priors from the models' own predictions about objects interactions into learning the GNN dynamics model, making long-term dynamics predictions more stable. Utilizing the structure of the model, we can generate informative samples for planning with a Rapidly Exploring Random Tree (RRT) in the latent space, enabling us to solve a challenging box sorting task from visual input.



(a) Initial scene with goal (b) Final achieved execution

Fig. 1: Planning scenario with learned model. The goal is to move the blue and yellow boxes to the colored areas with the red pusher.

II. Dynamics Model Learning Framework

A. Overview

Our framework consists of three parts, an object encoder Ω that turns observations into latent vectors $z_{1:m}$, a compositional NeRF-based decoder D that renders the latent vectors back into images of the scene in order to train the encoder, and a graph neural network dynamics model F_{GNN} that predicts the evolution of the scene in the latent space. Assume a scene is observed by RGB images $I^i \in \mathbb{R}^{3\times h_I \times w_I}$, $i = 1, \dots, V$ from V many camera views and that the scene contains m objects $j = 1, \dots, m$. We further assume to have camera projection matrices $K^i \in \mathbb{R}^{3\times 4}$ for each view and binary masks $M_j^i \in \{0,1\}^{h_I \times w_I}$ of each object j in view i. The encoder Ω fuses the information of object j observed from the multiple views such that

$$z_j = \Omega\left(I^{1:V}, K^{1:V}, M_j^{1:V}\right) \in \mathbb{R}^k \tag{1}$$

represents each object j separately. This Ω is trained end-to-end with the decoder D that reconstructs an image

$$I = D(z_{1:m}, K) \tag{2}$$

for arbitrary views specified by the camera matrix K from the set of latent object representations $z_{1:m}$.

B. Implicit Object Encoder

Instead of learning Ω defined in (1) as a direct mapping from images, camera matrices and masks to the latent

[1]TU Berlin. [2]UC San Diego.
[4]Massachusetts Institute of Technology.
This paper is an extended abstract of the preprint [1].

vectors, we first encode each object in the scene as a feature-valued *function* over 3D space. For each view, we project the query point $x \in \mathbb{R}^3$ from world into camera coordinates $K^i(x) \in \mathbb{R}^3$, where a feature $E(I^i, K^i(x)) \in \mathbb{R}^{n_o}$ at the projected coordinate is computed from the image using bilinear interpolation and the projected coordinate itself. Similar architectures of computing such pixel features from world coordinates have been proposed, e.g. in [7]–[9] for the *single* object case. Intuitively, $E(I^i, K^i(x))$ is a feature vector computed from what can be seen of the world at x in the image I^i from viewpoint i, taking into account its location relative to the camera origin of the view i. By summing over the individual views taking the masks $M_j^{1:V}$ into account, we define the feature function for object j as

$$y_j(x) = \frac{1}{p(x)} \sum_{i:\ K^i(x) \in M_j^i} E(I^i, K^i(x)) \in \mathbb{R}^{n_o} \tag{3}$$

with $p(x) = \sum_{i:\ K^i(x) \in M_j^i} 1$. An advantage of this formulation is that it naturally handles occlusions in different views and fuses the observations from different camera views consistently. Given the implicit object function $y_j(\cdot)$ of object j, we turn them into a latent vector $z_j \in \mathbb{R}^k$ representing object j with another network Φ by querying y_j on a workspace set $\mathcal{X}_h \in \mathbb{R}^{d \times h \times w}$ that is large enough to contain all objects. This produces an object feature voxel grid that is processed with a 3D convolutional neural network

$$z_j = \Phi(y_j) = \text{CNN}(y_j(\mathcal{X}_h)). \tag{4}$$

The resulting z_j's contain not only the appearance information of the objects, but also their spatial configurations in the scene with respect to other objects.

C. Decoder as Compositional, Conditional NeRF Model

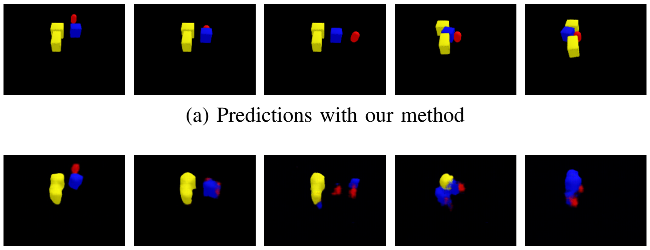
The decoder to train the encoder end-to-end is based on neural radiance fields. The general idea of NeRF [4] is to learn a function f that predicts, at a 3D world coordinate $x \in \mathbb{R}^3$, the (emitted) RGB color value $c(x) \in \mathbb{R}^3$ and volume density $\sigma(x) \in \mathbb{R}_{\geq 0}$. Compared to the standard NeRF formulation where one single model represents the whole scene, we associate separate NeRFs for each object $(\sigma_j(x), c_j(x)) = f_j(x) = f(x, z_j)$ by conditioning them on z_j. To turn those $f_{1:m}$ back into a global NeRF model that can be rendered, we sum the individual predicted object densities $\sigma(x) = \sum_{j=1}^m \sigma_j(x)$ and obtain the colors as $c(x) = \frac{1}{\sigma(x)} \sum_{j=1}^m \sigma_j(x) c_j(x)$. These composition formulas have been proposed multiple times in the literature, e.g. [10], [11]. This composition forces the individual NeRFs to learn the 3D configuration of each object individually and therefore ensures that each f_j only predicts the object where it is located in the 3D space.

D. Latent Dynamics Model with Graph Neural Networks

Having trained the implicit auto-encoder framework with an image reconstruction loss, we learn a dynamics model

$$z_{1:m}^{t+1} = F_{\text{GNN}}\left(z_{1:m}^t, A^t, q^t\right) \tag{5}$$

in the latent space with a graph neural network, where $A^t \in \{0,1\}^{m \times m}$ is the adjacency matrix and q^t the action at time t. Following [12], we use multi-step message passing to deal with situations where multiple objects interact. The adjacency matrix A within the GNN plays a crucial role in determining which objects interact. While a dense adjacency matrix in principle works, we found that the prediction performance greatly increases if A reflects more accurately which objects can exchange forces or not. We exploit the density prediction of the NeRF for each object to determine the adjacency matrix from the models' own predictions during training and planning by computing the collision integrals $\int_{\mathcal{X}} [\sigma(x, z_i) > \kappa][\sigma(x, z_j) > \kappa] \, \mathrm{d}x$ over the density predictions for objects i and j to check if they can interact.



(a) Predictions with our method

(b) Predictions with CNN decoder baseline

Fig. 2: Forward predictions of the model when applying an action sequence to the red pusher after observing the scene only initially. As one can see, with our proposed method in (a), the predictions are very sharp, while with the CNN decoder baseline (b) after only a few steps the predictions are of little use.

III. Latent-Space RRT

We grow a tree in the latent space of all objects for solving a planning task. Instead of sampling in the latent space directly, which is not guaranteed to produce valid samples [13], we sample in the space of center-of-mass configurations of all objects to efficiently expand the tree. These center-of-masses of the objects as well as the cost function evaluation is computed from the models' own predictions.

IV. Experiments

We consider a rigid-body scenario with multiple objects, where the goal is to push the objects with the red pusher to specified target regions, see Fig. 1. Refer to the video `https://dannydriess.github.io/compnerfdyn/` for more visualizations. This scenario is challenging due to multiple reasons. First, it is composed of many objects, implying a broad scene distribution and that many objects can interact. The mechanics of such multi-body pushing is non-trivial, contact between multiple objects at the same time can be established and broken at multiple phases of the motion. Furthermore, we do not assume that the red pusher starts in contact with an object. Hence, long-term predictions inherently have to be made in order to establish contact, before any object movement is registered. Fig. 2 shows the forward prediction of the model when applying an action sequence to the red pusher from an initial observation only. The CNN decoder baseline (Fig. 2b) replaces the compositional NeRF decoder with a CNN-based deconvolution decoder. The rest of the architecture stays the same. As one can see, even after 38 time-steps predicted into the future, the predictions with our model (Fig. 2a) still produce sharp objects, while with the CNN baseline (Fig. 2b) the predictions are very unstable. Fig. 1 shows the result of applying our method to solve a box sorting task.

References

[1] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint, "Learning multi-object dynamics with compositional neural radiance fields," *arXiv preprint arXiv:2202.11855*, 2022.

[2] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.

[3] 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.

[4] 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*, 2020, pp. 405–421.

[5] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robotics and Automation Letters*, 2022.

[6] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-nerf: Using a neural radiance field to grasp transparent objects," *arXiv preprint arXiv:2110.14217*, 2021.

[7] 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.

[8] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2304–2314.

[9] J.-S. Ha, D. Driess, and M. Toussaint, "Learning neural implicit functions as object representations for robotic manipulation," *arXiv preprint arXiv:2112.04812*, 2021.

[10] M. Niemeyer and A. Geiger, "Giraffe: Representing scenes as compositional generative neural feature fields," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[11] K. Stelzner, K. Kersting, and A. R. Kosiorek, "Decomposing 3d scenes into objects via unsupervised volume segmentation," *arXiv preprint arXiv:2104.01148*, 2021.

[12] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, "Propagation networks for model-based control under partial observation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1205–1211.

[13] B. Ichter and M. Pavone, "Robot motion planning in learned latent spaces," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2407–2414, 2019.