NEURAL RADIANCE FIELD CODEBOOKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Compositional representations of the world are a promising step towards enabling high-level scene understanding and efficient transfer to downstream tasks. Learning such representations for complex scenes and tasks remains an open challenge. Towards this goal, we introduce Neural Radiance Field Codebooks (NRC), a scalable method for learning object-centric representations through novel view reconstruction. NRC learns to reconstruct scenes from novel views using a dictionary of object codes which are decoded through a volumetric renderer. This enables the discovery of reoccurring visual and geometric patterns across scenes which are transferable to downstream tasks. We show that NRC representations transfer well to object navigation in THOR, outperforming 2D and 3D representation learning methods by 3.1% success rate. We demonstrate that our approach is able to perform unsupervised segmentation for more complex synthetic (THOR) and real scenes (NYU Depth) better than prior methods (.101 ARI). Finally, we show that NRC improves on the task of depth ordering by 5.5% accuracy in THOR.

1 INTRODUCTION

Compositional understanding of the world is a key characteristic of human vision (Rosch et al., 1976; Johnson et al., 2003). Models with such capabilities enable us to reason about the relationship between objects, understand novel scenarios, and navigate the world around us. Modeling the world as compositional elements allows knowledge to be shared across instances of the same category. For example, when driving down the road we can infer how other cars will act and plan accordingly without any prior interaction with those specific cars. Learning complex, compositional representations without explicit supervision remains an open challenge.

A fundamental question is what level of supervision and structure of data is needed to learn such models of the world. Classic work in unsupervised segmentation (Shi & Malik, 2000) leverages color and locality to parse images into graphical components. A more recent line of work (Burgess et al., 2019; Greff et al., 2019; Locatello et al., 2020; Lin et al., 2020; Monnier et al., 2021; Smirnov et al., 2021) has demonstrated the potential of reconstruction as a guiding objective. These methods encode images into sparse, discrete elements then reconstruct them. However, learning from reconstructing static images has been shown to have limitations (Karazija et al., 2021). Although great progress has been made, such methods still depend on color cues, and fail for scenes with more complex textures (Papa et al., 2022).

Given advances in neural rendering (Mildenhall et al., 2021), reconstructing scenes from novel views has emerged as a promising source of supervision. In principle, reconstruction from different views should require geometric understanding of the scene. Recent works have integrated NeRF with object-centric inductive biases to decompose scenes (Stelzner et al., 2021; Yu et al., 2021b; Sajjadi et al., 2022a). Although these methods have shown to be effective, they are limited to simple synthetic scenes with few objects. Additionally, scenes are decomposed into object instances, therefore semantic and geometric information is not aggregated at the object category level.

With this in consideration, we propose Neural Radiance Field Codebooks (NRC), a method that learns a shared set of object codes by finding reoccurring geometric and visual patterns across scenes. NRC leverages the novel view reconstruction objective to learn correspondences between multiple views of the same object, while the shared codebook enables learning correspondences between similar objects across scenes. In contrast to most prior works (Stelzner et al., 2021; Yu et al., 2021b; Sajjadi et al., 2022a) which fix the number of objects per scene, NRC differentiably learns the number of categorical object codes.

Figure 1: Rendered examples of learned codes from the ProcTHOR dataset with the original image. We observe that NRC categorizes images based on reoccurring geometric and visual patterns. Although the couches in the top row differ in visual appearance, they are assigned the same code based on their geometric similarity. In the bottom row, we observe that different textured floors are categorized based on their shared planar geometry.

We show the utility of NRC learned representations for the tasks of unsupervised segmentation, object navigation, and depth ordering. For segmentation of ProcTHOR (Deitke et al., 2022) scenes we show .101 ARI improvement. On real-world images (NYU Depth (Silberman et al., 2012)) we show promising qualitative results (Figure 3) and 0.041 ARI improvement over comparable unsupervised baselines (Stelzner et al., 2021; Yu et al., 2021b). For object navigation (Kolve et al., 2017) and depth ordering (Yang et al., 2011), where geometric understanding is crucial, we observe 3.1% improvement in navigation success rate and 5.5% improvement in depth ordering accuracy compared to prior methods. Interestingly, we also find evidence that the learned codes categorize objects by not only visual appearance, but also geometric structure (Figure 1).

2 Related Work

Object-Centric Learning Object-centric learning works aim to build compositional models of the world from building blocks which share meaningful properties and regularities across scenes. Prior works such as MONet (Burgess et al., 2019), IODINE (Greff et al., 2019), Slot Attention (Locatello et al., 2020), and Monnier et al. (2021) have demonstrated the potential for disentangling objects from images. Recent work in this line has shown the ability to decompose videos (Kabra et al., 2021; Kipf et al., 2021). Marionette (Smirnov et al., 2021) learns a shared dictionary for decomposing scenes of

2D sprites. We draw inspiration from MarioNette for learning codebooks, but differ in that we model the image formation process and intra-code variation and dynamically add codes to our dictionary.

3D Object-Centric Learning Recent work has to shown novel view reconstruction to be a promising approach for disentangling object representations. uORF (Yu et al., 2021b) and ObSuRF (Stelzner et al., 2021) combine Slot Attention with Neural Radiance Fields (Mildenhall et al., 2021) to decompose scenes. COLF (Smith et al., 2022) replaces the volumetric renderer with light fields to enable faster rendering. SRT (Sajjadi et al., 2022b) encodes scenes into a set of latent vectors which are used to condition a light field. OSRT (Sajjadi et al., 2022b) extends SRT by explicitly assigning regions of the image to latent vectors. Although great progress has been made, these methods still have been limited to synthetic and relatively simple scenes. Our work differs from previous 3D object-centric works in that we learn reoccurring objects codes across scenes and explicitly localize the learned codes. Additionally, we learn can model an unbounded number of objects per scene compared to prior work which fixes this hyper-parameter a priori. We show that this approach generalizes to more complex synthetic and real-world scenes.

Neural Rendering Advances in neural rendering, in particular Neural Radiance Fields (NeRF) (Mildenhall et al., 2021), have enabled a host of new applications (Jang & Agapito, 2021; Mildenhall et al., 2022; Pumarola et al., 2021; Park et al., 2021; Lazova et al., 2022; Niemeyer & Geiger, 2021). NERF differentiably renders novel views of a scene by optimizing a continuous volumetric scene function given a sparse set of input views. The original formulation of NeRF learned one representation for each scene, however recent work (Yu et al., 2021a; Jain et al., 2021; Kosiorek et al., 2021) has shown that conditioning the volumetric scene function on images from new scenes enables generation of novel views from a few images.

Dictionary/Codebook Learning Dictionary (Codebook) learning (Olshausen & Field, 1997) involves learning of a specific set of atoms or codes that potentially form a basis and span the input space through sparse combinations. Codebooks have been widely used for generative and discriminative tasks across vision (Elad & Aharon, 2006; Mairal et al., 2008), NLP (Mcauliffe & Blei, 2007) and signal processing (Huang & Aviyente, 2006). Learning sparse representations based on codes assists in large-scale algorithms relying on latent representations. More recently codebooks have been shown to be crucial in scaling discrete representation learning (Van Den Oord et al., 2017; Kusupati et al., 2021). Marionette (Smirnov et al., 2021) is an object-centric representation learning method that relies on codebooks, unlike most other methods that are developed around set latent representations (Sajjadi et al., 2022a;b; Locatello et al., 2020; Yu et al., 2021b). Object-centric codebooks help in semantic grounding for transfer between category instances and is important for large-scale representation learning across diverse scenes and objects.

3 METHOD

Given observations, RGB images, from a set of n scenes, $\mathbb{S} := \{S_i\}\ i \in [n]$, our goal is to decompose each scene, S_i into a set of learned codes, $\mathbb{L} := \{l_i \in \mathcal{R}^d\}\ i \in [k]$, selected from a shared dictionary with k atoms that have d-dimensional assigned latents. These prototypical codes capture the visual and geometric properties of object categories and are decoded through a volumetric renderer. Note that not all n scenes need to have all k objects present within them. The prototype codebook enables the model to learn correspondences between similar objects across scenes. At the same time, the integration of 3D information through volumetric rendering and multiple views allows the model to attribute different views of the same object to a single code – akin to data augmentation.

Figure 2 illustrates the key components and pipeline of our method. Given an input frame and novel frame, $O_h, O'_h \in \mathcal{R}^{3\times H\times W}$ respectively from scene S_h, we first encode O_h into a spatial feature map, $f_h \in R^{d\times H/k\times W/k}$, using a convolutional network, F_θ. We project each point, (x, y, z) in world coordinates of the novel view to camera coordinates in the input frame, (x, y), using the relative camera pose following the procedure of Yu et al. (2021a). Given (x, y) we select the spatial feature $f_h^{x,y} \in \mathcal{R}^d$ from the patch that contains the projected coordinates. Using $f_h^{(x,y)}$ we assign the closest code in our learned dictionary of latents $\mathbb{L}$ using L_2 distance. We concatenate positional encodings as a function of the relative position of (x', y') in the patch and absolute position in the image. Next we pass the assigned latent code to a variation module which transforms the category level code to an

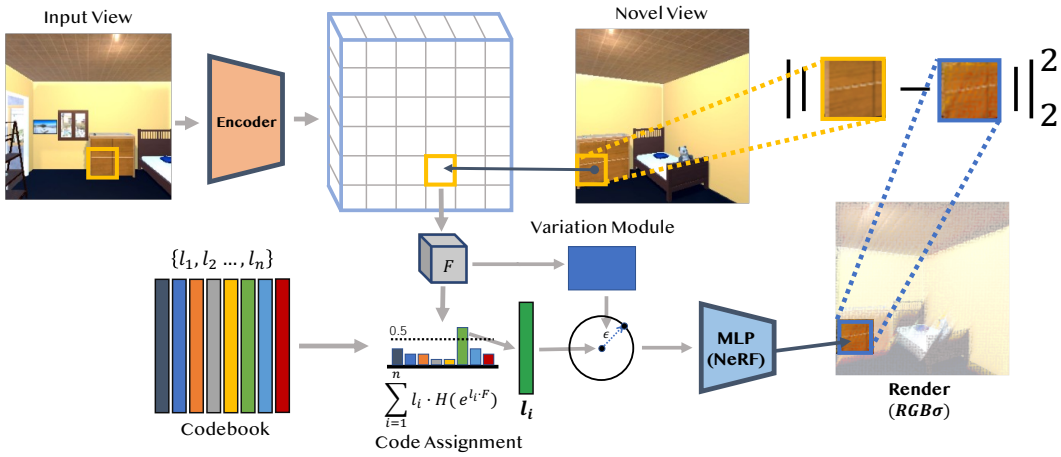

Figure 2: An overview of NRC. We learn a set of shared codes for decomposing scenes into objects. Each point in the scene is assigned one of n latent codes from the codebook. The variation module models the intra-code variation between objects by perturbing the code in latent space. A conditional NERF model renders the scene and is compared to the ground truth novel view for supervision.

instance code. This allows us to model variations within particular object categories. The instance code is used to condition the NeRF network, $H_{\theta'}$, which outputs a color and density. Following NeRF, A pixel-wise L_2 loss with the ground truth novel view is used as supervision.

Assigning Categorical Codes Given the features for a point in the novel view, $f_h^{x,y}$, we need to assign a code, l^*, chosen from the shared dictionary, $\mathbb{L}$. We do so with a $\arg\max$ 1-nearest-neighbors during inference:

$$l^*(x,y) \leftarrow \overset{\text{(STE)}}{\underset{l_i;\ i \in [k]}{\arg\max}} \frac{e^{-\|l_i - f_h^{x,y}\|_2}}{\sum_{j=1}^{k} e^{-\|l_j - f_h^{x,y}\|_2}} \tag{1}$$

During training (back propagation), we use the $\mathrm{softmax}$ relaxation of the $\arg\max$ in 1-nearest neighbors to facilitate learning. This is done using the straight-through-estimator (STE) (Bengio et al., 2013) for the ease of implementation as shown in Equation 1.

Elastic Addition of New Categorical Codes The number of codes should depend on the complexity of the scenes that are being modeled. In contrast, prior methods have assumed a fixed number of objects per scene. Learning when to add new codes is non-trivial because the number and selection of codes is discrete and non-differentiable. To circumvent this problem, we use a series of step functions with a straight-through-estimator (STE) to sequentially add elements to the dictionary. Each code $l_i \in \mathbb{L}$ is gated according to the following:

$$l_i \leftarrow \mathcal{T}\left(\boldsymbol{\sigma}(s - {}^{i^2}/\lambda), \frac{1}{2}\right) \cdot l_i; \quad \mathcal{T}(a,t) := \begin{cases} 1, & a > t \\ 0, & a \leq t \end{cases} \tag{2}$$

$\mathcal{T}(.)$ is a binarization function in the forward pass and lets the gradients pass through using STE in the back pass, $\boldsymbol{\sigma}(\cdot)$ is the sigmoid function, λ is a scaling hyperparameter and s is a learnable scoring parameter whose magnitude correlated with the overall capacity (number of codes) required to model the scenes accurately and signifies when to add a new code. A new code l_i is added when s exceeds the threshold i^2/λ. This formulation can be viewed as the discrete analog of a gaussian prior over the number of elements, k in the codebook: $P(k) = e^{-k^2/\lambda}$.

Modeling Intra-Code Variation Each prototype in the learned dictionary encodes the geometric and visual characteristics of a class category. Once a prototypical code has been assigned to a point in the novel frame, the model must account for variation across instances. We model this variation in

latent space using an encoder that takes in both the spatial feature map, $f_h^{x,y}$ and the prototype code $l^*(x, y)$. We rescale the norm of the variation vector by ϵ, a hyperparameter, to ensure the instance and prototype codes are close in latent space. We get our instance code

$$l^*_{\text{instance}}(x, y) = l^*(x, y) + \epsilon \cdot \frac{G_{\theta'}([l^*(x, y), f_h^{x,y}])}{||G_{\theta'}([l^*(x, y), f_h^{x,y}])||_2}. \tag{3}$$

We concatenate $f_h^{x,y}$ and $l^*(x, y)$ as input to the variation module, $G_{\theta'}$, which we model as a 3-layer convolutional network. $G_{\theta'}$ provides us with a d dimensional perturbation vector to model variations in our object prototype code and transform it to an instance code.

Decoding and Rendering The loss that we use to supervise our pipeline is the same as that of Yu et al. (2021a). We use a 3-layer MLP, $H_{\hat{\theta}}$, to transform from our instance code to a pixel prediction of the novel view. In reality $H_{\hat{\theta}}$, is a NeRF which is meant to reconstruct the whole scene. This is written as:

$$H_{\hat{\theta}}(l^*_{\text{instance}}(x, y), \mathbf{p}, \mathbf{d}) = (\mathbf{c}, \sigma), \tag{4}$$

where $\mathbf{p} = (x, y, z)$ is a coordinate in the scene, $\mathbf{d} \in \mathcal{R}^3$ is a view direction, $\mathbf{c}$ is the RGB value at $\mathbf{p}$ in the direction of $\mathbf{d}$ and σ is the volume density at that point. Recall that (x, y, z) corresponds to (x, y) in the input frame O_h. We can project (x, y, z) into the camera coordinates of the novel view O'_h to get (x', y'). This pixel (x', y') the novel view corresponds to (x, y) in the input frame, meaning they represent the same point in world coordinates. To get an RGB value for (x, y), we use volume rendering along the ray from camera view O_h into the scene, given by

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t) \cdot \sigma(t) \cdot \mathbf{c}(t) \cdot dt, \tag{5}$$

where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(s) \cdot ds\right)$ handles absorbance. Given a target view with pose $\mathbf{P}$, the ray to the target camera is given by $\mathbf{r}(t) = \mathbf{o} + t \cdot \mathbf{d}$ where $\mathbf{d}$ is a unit direction vector which passes through (x, y). The volume rendering for a particular pixel occurs along this ray. Let $\mathbf{d}'$ be the direction associated with the novel view (x', y') and $\mathbf{r}'(t) = \mathbf{o} + t \cdot \mathbf{d}'$. The pixel intensity at (x', y') is given by $\hat{\mathbf{C}}' = \hat{\mathbf{C}}(\mathbf{r}')$ and our final loss is

$$\mathcal{L}(O_h, O'_h, x, y) = \|\hat{\mathbf{C}}' - O'_h(x', y')\|_2 + s, \tag{6}$$

where $O'_h(x', y')$ is the ground-truth pixel value at (x', y') scoring parameter for code addition. We penalize s in the loss to encourage learning a minimal number of codes.

4 Experiments

We evaluate our decomposition and representations on several downstream tasks: unsupervised segmentation (real and synthetic), object navigation, and depth ordering. NRC shows improvement over baseline methods on all the three tasks. Prior works in object-centric learning have focused on unsupervised segmentation for evaluating the quality of their decomposition, often only on synthetic data. We show that NRC representations are also effective for downstream applications that require geometric and semantic understanding of scenes such as object navigation and depth ordering.

4.1 Datasets

ProcTHOR & RoboTHOR THOR (Kolve et al., 2017) consists of interactive home environments built in the Unity game engine. We benchmark on the task of object navigation in RoboTHOR (Deitke et al., 2020), a variant of the THOR environment aimed at sim2real benchmarking. Object navigation consists of an agent moving through different scenes to locate specified objects. It is a natural application for 3D object-centric representations as it requires understanding of the scene layout and object knowledge. RoboTHOR consists of 89 indoor scenes split between train, validation, and test. ProcTHOR (Deitke et al., 2022) consists of procedurally generated indoor scenes similar to RoboTHOR. Examples of THOR scenes can be found in Appendix B.

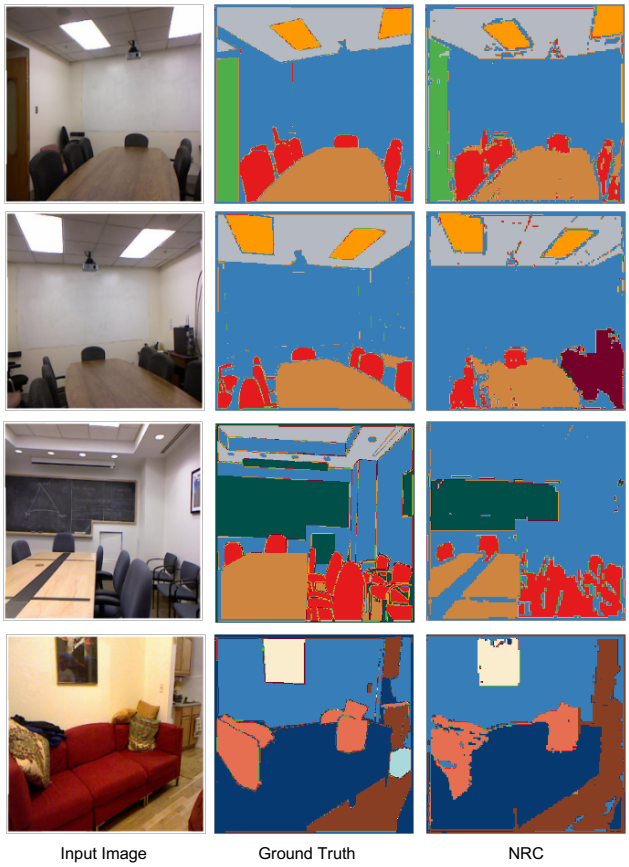

Figure 3: Unsupervised segmentation results for NYU-Depth. NRC is able to segment scenes that have significant object category overlap with ProcTHOR. Most object-centric representation works have focused on synthetic datasets for segmentation. We show the first results for object-centric unsupervised segmentation of real-world scenes.

CLEVR-3D CLEVR-3D (Johnson et al., 2017) is a synthetic dataset consisting of geometric primitives from multiple views and is used for unsupervised segmentation. Following the convention of Stelzner et al. (2021), we test on the first 500 scenes of the validation set and report foreground-adjusted random index (FG-ARI). Adjusted random index (ARI) Yeung & Ruzzo (2001) measures the agreement between two clusterings and is a standard metric for unsupervised segmentation. In our case the two clusterings are the predicted and ground truth segmentations. Foreground adjusted random index only measures the ARI for pixels belonging to foreground objects. For comparison to prior works, we consider segmentations at both the class and instance level to be correct for CLEVR-3D, ProcTHOR, and NYU Depth. Further details can be found in Appendix B.

NYU Depth The NYU Depth Dataset (Silberman et al., 2012) consists of images from real-world indoor scenes accompanied by depth and segmentation maps. Methods are trained on the ProcThor dataset then evaluated for unsupervised segmentation on NYU Depth. Note that NYU Depth does not provide ground truth camera pose, therefore novel view reconstruction methods cannot be trained on it. We chose NYU Depth because it has similar object categories and scene layouts compared to THOR. We report the adjusted random index (ARI).

4.2 Unsupervised Segmentation

Experimental Setup We evaluate NRC, ObSuRF, uORF, and MarioNette for unsupervised segmentation on ProcTHOR, CLEVR-3D, and NYU Depth. We compare with MarioNette because it uses a similar code mechanism for reconstruction. For CLEVR-3D we report FG-ARI for comparison

Table 1: Segmentation results (ARI) for NRC and comparable methods. We find that for more complex datasets, ProcTHOR and NYU Depth, NRC outperforms other methods.

Method	ProcTHOR (ARI)	NYU Depth (ARI)	CLEVR-3D (FG-ARI)
MarioNette	.127	.035	-
uORF	.193	.115	.962
ObSuRF	.228	.141	**.978**
NRC	**.295**	**.182**	.977

to prior works and ARI for others. For NYU Depth evaluation we use the representations trained on ProcTHOR and only consider classes that are seen in training data set.

Results Quantitative results can be found in Table 1. We find that for NYU Depth and ProcTHOR which have more complicated scenes, NRC significantly outperforms the baselines. Figure 1 shows examples of the object codes learned by ProcTHOR and Figure 3 shows segmentation examples of real-world images. To our knowledge, this is the *first* object-centric learning method which has shown unsupervised segmentation results for real-world images. We find evidence that NRC categorizes similar objects across scenes based on both geometry and visual appearance. In the top row of Figure 1, couches with differing visual appearance are assigned the same code due to their shared geometry. In the middle row of Figure 1, we show examples where different views of the same floor and floors of different texture are categorized by the same code.

4.3 Object Navigation

Experimental Setup We design the object navigation experiments in THOR to understand how well the learned representations transfer from observational data to embodied navigation (Anderson et al., 2018; Batra et al., 2020). Prior works in object-centric representation learning have primarily focused on unsupervised segmentation as the downstream task. Object navigation consists of an embodied agent with the goal of moving through indoor scenes to specified objects. The agent can rotate its camera and move in discrete directions. At each step, the input to the agent is the current RGB frame relayed by the camera.

For the representation learning component of the experiment, we collect observational video data from a heuristic planner (Appendix A), which walks through procedurally generated ProcTHOR scenes. In total the dataset consists of 1.5 million video frames from 500 indoor scenes. For further dataset details and example videos see Appendix B.

After training on the videos from ProcTHOR, we freeze the visual representations following standard practice (Khandelwal et al., 2022). We train a policy using DD-PPO (Wijmans et al., 2019) for 200M steps on the training set of RoboTHOR then evaluate on the test set. We report success rate (SR) and success weighted by path length (SPL). Success is defined as the agent signaling the stop action within 1 meter of the goal object with the object in its view. SPL is defined as $\frac{1}{N}\sum_{i=1}^{N} S_i \frac{\ell_i}{\max(p_i,\ell_i)}$, where l_i is the shortest possible path, p_i is the taken path, and S_i is the binary indicator of success for episode i.

We evaluate the baselines of ObSuRF, uORF, Video MoCo (Feichtenhofer et al., 2021), and Embed-CLIP (Khandelwal et al., 2022). ObSuRF and uORF are 3D, object-centric methods, and Video MoCo is a contrastive video representation learning method. We include Video MoCo for comparison as it was designed for large-scale, discriminative tasks, while ObSuRF and uORF were primarily intended for segmentation. For further implementation details on the baseline methods see Appendix A.

Results Table 2 showcases the performance of NRC and baselines on RoboTHOR ObjectNav Challenge. NRC outperforms the best baseline by about 3% in success rate and by a 20% relative improvement in SPL. These significant performance gains point to the value of NRC's object-centric representations that outperform models like EmbCLIP, trained on massive amounts of internet data, and state-of-the-art video representation learning methods like Video MoCo. Another key observation is that NRC is at least up to 18.8% and 66% improvements in success rate and relative SPL over

Table 2: Results for object navigation on the RoboTHOR ObjectNav Challenge. Visual representations are trained on observations from 500 scenes of ProcThor. A policy is learned on top of the frozen visual representations by training on the ObjectNav task in RoboTHOR training scenes. Finally the reported results are obtained by evaluating on RoboTHOR test scenes.

Method	Success Rate (%)	SPL
uORF (Yu et al., 2021b)	31.3	.146
ImageNet Pretraining	33.4	.150
ObSuRF (Stelzner et al., 2021)	38.9	.167
Video MoCo (Feichtenhofer et al., 2021)	43.9	.184
EmbCLIP (Khandelwal et al., 2022)	47.0	.200
NRC (Ours)	**50.1**	**.239**

the recently proposed set-latent based object-centric representation learning methods like uORF and ObSuRF. This can be attributed to the strong design-choices of NRC.

4.4 Depth Ordering

Experimental Setup We compare NRC on the task of ordering objects based on their depth from the camera. Understanding the relative depth of objects requires both geometric and semantic understanding of a scene. For this task we evaluate on the ProcTHOR test dataset which provides dense depth and segmentation maps. Following the convention of Ehsani et al. (2018), we determine ground truth depth of each object by computing the mean depth of all pixels associated with its ground truth segmentation mask.

For evaluation, we select pairs of objects in a scene and the goal is to predict which object is closer. We take the segmentation that has the largest IoU with the ground truth mask as the predicted object mask. To determine the predicted ordering, we compute the mean predicted depth of each pixel associated with the predicted object mask. All representations are trained on the ProcTHOR dataset and evaluated on the ProcTHOR test set. In total we evaluate 2,000 object pairs.

Table 3: We compare depth ordering results on RoboTHOR with other geometrically-aware representations. Given pairs of objects in the scene, the model must infer which object is closer. We report the accuracy as the number of correct orderings over the total number of object pairs.

Method	Depth Order Acc. (%)
uORF (Yu et al., 2021b)	13.5
ObSuRF (Stelzner et al., 2021)	18.3
NRC (Ours)	**23.8**

Results Table 3 compares the depth ordering capability between NRC and other object-centric baselines. Depth ordering requires accurate segmentation and depth estimation. Owing to its stronger segmentation performance and better depth awareness, NRC is 5.5% and 10.3% more accurate than ObSuRF and uORF respectively. The fine-grained per-pixel allocation of categorical latent codes in NRC allows for better depth ordering over existing object-centric methods.

4.5 Ablation Study

We present an ablation study on the ProcTHOR dataset to determine the effect of the variation module and learnable codebook size on unsupervised segmentation performance. Quantitative results can be found in Table 4. We observe that performance improves by $\sim 9\%$ when intra-class variation is explicitly modeled. Intuitively, allowing for small variation between instances of the same category should lead to better representations and allow for greater expressiveness.

We also find that learning the number of codes moderately improves performance. Albeit, if number of codes is found via hyper-parameter tuning, performance can be matched (Table 5). Still, differentiably learning the length of the codebook avoids computationally expensive hyper-parameter tuning.

Table 4: Ablation study for modeling the intra-code variation and learning the number of codes evaluated on unsupervised segmentation in ProcTHOR. Default fixed number of codes is set to 25.

Method + (Ablation)	ProcTHOR (ARI)
NRC	.182
NRC + Learned # of Codes	.197
NRC + Variation Module	.284
NRC + Variation Module + Learned # of Codes	**.295**

5 Limitations

Though we show steps towards decomposing real-world images, the objects NRC can segment are limited to those seen in synthetic environments. Learning compositional models from large-scale, real-world video and images remains an open challenge. Novel view reconstruction requires camera pose, which is not available for most images or videos. Some data sets such as Ego4D (Grauman et al., 2022) provide data from inertial measurement units which could be used to approximate camera pose, although this approach is prone to drift.

An incorrect assumption that our method and most prior object-centric works make is that scenes are static. Assuming static scenes is a widely adopted design choice by most prior object-centric methods and NRC. However, it rarely is the case that scenes are free of movement due to the physical dynamics of our world. Recently, Kipf et al. (2021); Pumarola et al. (2021) made strides in learning representations from dynamic scenes.

Although NRC is relatively efficient compared to the other NeRF based methods, the NeRF sampling procedure is compute and memory intensive. Sajjadi et al. (2022a) and Smith et al. (2022) leverage object-centric light fields to reduce memory and compute costs. The efficiency improvement from modeling scenes as light fields is orthogonal to NRC and could be combined.

A final challenge inherent to novel view reconstruction is finding appropriate corresponding frames of videos. For example, if two subsequent frames differ by a 60° rotation of the camera, then most of the scene in the subsequent frame will completely new. Therefore, constructing the content in the novel view is ill-posed. Pairing frames with overlapping frustums is a potential solution, although the content of the scene may not be contained in the intersecting volume of the frustums.

6 Conclusion

Compositional, object-centric understanding of the world is a fundamental characteristic of human vision and such representations have potential to enable high-level reasoning and efficient transfer to downstream tasks. Towards this goal, we presented Neural Radiance Field Codebooks (NRC), a new approach for learning geometry-aware, object-centric representations through novel view reconstruction. By jointly learning a shared dictionary of object codes through a differentiable renderer and explicitly localizing object codes within the scene NRC learns finds reoccurring geometric and visual similarities to form objects categories. We observe NRC representations are effective for tasks that require geometric understanding such as object navigation and depth ordering. Through experiments we show that NRC representations improve performance on object navigation and depth ordering compared to strong baselines by 3.1% success rate and 5.5% accuracy respectively. Additionally, we find our method is capable of scaling to complex scenes with more objects and greater diversity. NRC shows relative ARI improvement over baselines for unsupervised segmentation by 29.4% on ProcTHOR and 29.0% on NYU Depth. Qualitatively, NRC representations trained on synthetic data from ProcTHOR show reasonable transfer to real-world scenes from NYU Depth.

NRC makes strides towards learning compositional models from complex scenes, however there are still significant limitations in the requirement for camera pose, assumptions that the scene is static, and difficulty finding corresponding video frames. Although NRC shows the potential of novel view reconstruction, the fundamental question remains as to what the necessary supervision is for fully disentangling the visual world.

7 REPRODUCIBILITY

We include our code in the supplementary material. Hyper-parameters and implementation details for reproducing experiments can be found in Appendix A. Details for evaluation procedures and datasets can be found in Section 4 and Appendix B.

REFERENCES

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3164–3174, 2020.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, et al. Procthor: Large-scale embodied ai using procedural generation. *arXiv preprint arXiv:2206.06994*, 2022.

Kiana Ehsani, Roozbeh Mottaghi, and Ali Farhadi. Segan: Segmenting and generating the invisible. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6144–6153, 2018.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3299–3309, 2021.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pp. 2424–2433. PMLR, 2019.

Ke Huang and Selin Aviyente. Sparse representation for signal classification. *Advances in neural information processing systems*, 19, 2006.

Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5885–5894, 2021.

Wonbong Jang and Lourdes Agapito. Codenerf: Disentangled neural radiance fields for object categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12949–12958, 2021.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Scott P Johnson, Dima Amso, and Jonathan A Slemmer. Development of object concepts in infancy: Evidence for early learning in an eye-tracking paradigm. *Proceedings of the National Academy of Sciences*, 100(18):10568–10573, 2003.

Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matt Botvinick, Alexander Lerchner, and Chris Burgess. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. *Advances in Neural Information Processing Systems*, 34:20146–20159, 2021.

Laurynas Karazija, Iro Laina, and Christian Rupprecht. Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation. *arXiv preprint arXiv:2111.10265*, 2021.

Apoorv Khandelwal, Luca Weihs, Roozbeh Mottaghi, and Aniruddha Kembhavi. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14829–14838, 2022.

Thomas Kipf, Gamaleldin F Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Adam R Kosiorek, Heiko Strathmann, Daniel Zoran, Pol Moreno, Rosalia Schneider, Sona Mokrá, and Danilo Jimenez Rezende. Nerf-vae: A geometry aware 3d scene generative model. In *International Conference on Machine Learning*, pp. 5742–5752. PMLR, 2021.

Aditya Kusupati, Matthew Wallingford, Vivek Ramanujan, Raghav Somani, Jae Sung Park, Krishna Pillutla, Prateek Jain, Sham Kakade, and Ali Farhadi. Llc: Accurate, multi-purpose learnt low-dimensional binary codes. *Advances in Neural Information Processing Systems*, 34:23900–23913, 2021.

Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. *arXiv preprint arXiv:2204.10850*, 2022.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. Space: Unsupervised object-oriented scene representation via spatial attention and decomposition. *arXiv preprint arXiv:2001.02407*, 2020.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

J Mairal, F Bach, J Ponce, G Sapiro, and A Zisserman. Learning discriminative dictionaries for local image analysis. In *26th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

Jon Mcauliffe and David Blei. Supervised topic models. *Advances in neural information processing systems*, 20, 2007.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. Nerf in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16190–16199, 2022.

Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised layered image decomposition into object prototypes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8640–8650, 2021.

Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11453–11464, 2021.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

Samuele Papa, Ole Winther, and Andrea Dittadi. Inductive biases for object-centric representations in the presence of complex textures. In *UAI 2022 Workshop on Causal Representation Learning*, 2022.

Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5865–5874, 2021.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10318–10327, 2021.

Eleanor Rosch, Carolyn B Mervis, Wayne D Gray, David M Johnson, and Penny Boyes-Braem. Basic objects in natural categories. *Cognitive psychology*, 8(3):382–439, 1976.

Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetić, Mario Lučić, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. *arXiv preprint arXiv:2206.06922*, 2022a.

Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6229–6238, 2022b.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.

Dmitriy Smirnov, Michaël Gharbi, Matthew Fisher, Vitor Guizilini, Alexei A. Efros, and Justin Solomon. MarioNette: Self-supervised sprite learning. In *Advances in Neural Information Processing Systems*, 2021.

Cameron Smith, Hong-Xing Yu, Sergey Zakharov, Fredo Durand, Joshua B Tenenbaum, Jiajun Wu, and Vincent Sitzmann. Unsupervised discovery and composition of object light fields. *arXiv preprint arXiv:2205.03923*, 2022.

Karl Stelzner, Kristian Kersting, and Adam R Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv preprint arXiv:2104.01148*, 2021.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019.

Yi Yang, Sam Hallman, Deva Ramanan, and Charless C Fowlkes. Layered object models for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1731–1743, 2011.

Ka Yee Yeung and Walter L. Ruzzo. Details of the adjusted rand index and clustering algorithms supplement to the paper " an empirical study on principal component analysis for clustering gene expression data " (to appear in bioinformatics). 2001.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4578–4587, 2021a.

Hong-Xing Yu, Leonidas J Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. *arXiv preprint arXiv:2107.07905*, 2021b.

A Implementation Details

A.1 OSRT

We do not compare with OSRT Sajjadi et al. (2022a) on data sets that were not evaluated in the original paper. We contacted the authors, but a public implementation has not been released yet at the time of this work.

A.2 uORF

We use the implementation available at `https://github.com/KovenYu/uORF` We train on the training set of ProcTHOR which contains $\sim$ 1.5 million images. We train three different variations of the uORF no GAN model: from scratch, fine-tune with encoder and decoder pre-trained on CLEVR-567, and fine-tune with all three components pre-trained on CLEVR-567. For variations that train the slot attention component from scratch, we try setting the number of slots to 8 (default) and 16. We train for 10 epochs when loading from a pre-trained model, and for 15 epochs when training from scratch; roughly half of the epochs are coarse epochs, and we set percept_in to roughly one third of the epochs. We train for less epochs because our dataset is much larger. Our images within scene batch size is 4, and we turn off shuffling to ensure that batches contain images that are close together within the scene. We pass in the fixed locality parameter, but also set the decoder's locality parameter to false. To compensate between differences between our dataset and CLEVR-567, we tune nss scale, object scale, near plane, and far plane. We train models on a single NVIDIA A40.

A.3 OBSURF

We use the official implementation provided by Stelzner et al. (2021) at `https://github.com/stelzner/obsurf` with default hyper-parameters. For object navigation experiments in RoboTHOR, we take the trained encoder and learn a policy network with an A3C architecture using DD-PPO for 200M steps. For the encoder, we use the original architecture described in Stelzner et al. (2021).

B Datasets

B.1 ProcTHOR

The ProcTHOR dataset we collect consists of the first 500 scenes for training and 50 for test. We collect 10 episodes each which consists of 300 action steps. See Figure 4 for a sample of the dataset.

B.2 CLEVR-3D

CLEVR-3D (Johnson et al., 2017) is a synthetic dataset which consists of geometric primitives placed on a monochrome background. Following the convention of (Stelzner et al., 2021) we evaluate on this benchmark for unsupervised object segmentation. Following convention we evaluate on the first 500 scenes of the validation set and report foreground - adjusted random index.

Figure 4: Snippets of Videos from ProcTHOR dataset used for training NRC.

Table 5: Performance of NRC with a fixed codebook of various sizes on ProcTHOR.

Codebook Size	ProcTHOR (ARI)
10	.247
20	.256
30	.266
40	.279
50	.286
60	.293

Table 6: The size of the learned codebook averaged over 5 runs on each dataset.

	Codebook Size
CLEVR-3D	15.6 ± 1.2
ProcTHOR	53.8 ± 2.4