LEIA: Latent View-invariant Embeddings for Implicit 3D Articulation

Archana Swaminathan[1], Anubhav Gupta[1], Kamal Gupta[1], Shishira R Maiya[1], Vatsal Agarwal[1], and Abhinav Shrivastava[1]

University of Maryland, College Park MD 20742, USA
https://archana1998.github.io/leia/

Abstract. Neural Radiance Fields (NeRFs) have revolutionized the reconstruction of static scenes and objects in 3D, offering unprecedented quality. However, extending NeRFs to model dynamic objects or object articulations remains a challenging problem. Previous works have tackled this issue by focusing on part-level reconstruction and motion estimation for objects, but they often rely on heuristics regarding the number of moving parts or object categories, which can limit their practical use. In this work, we introduce LEIA, a novel approach for representing dynamic 3D objects. Our method involves observing the object at distinct time steps or "states" and conditioning a hypernetwork on the current state, using this to parameterize our NeRF. This approach allows us to learn a view-invariant latent representation for each state. We further demonstrate that by interpolating between these states, we can generate novel articulation configurations in 3D space that were previously unseen. Our experimental results highlight the effectiveness of our method in articulating objects in a manner that is independent of the viewing angle and joint configuration. Notably, our approach outperforms previous methods that rely on motion information for articulation registration.

Keywords: Articulated Objects · Neural Radiance Fields · 3D Vision

1 Introduction

Our world is full of dynamic objects moving and interacting in space and time. Humans develop this (rather impressive) understanding of how these everyday objects move and interact in three-dimensional space at a very early stage of the brain development [31]. The task involves understanding not only the static geometry but also the dynamic movements and spatial relationships between parts of an object, often referred to as articulations of the object. Understanding and representing object articulations from images and/or videos is also pivotal in enabling machines to perceive and navigate the physical world with finesse. In this work, we propose a novel method to model the object articulations by learning view-invariant latent embeddings of the 3D object from multiview images. Existing works in modeling object articulation in the three dimensional space typically use priors in the form of pre-trained large scale models [51], videos [27],

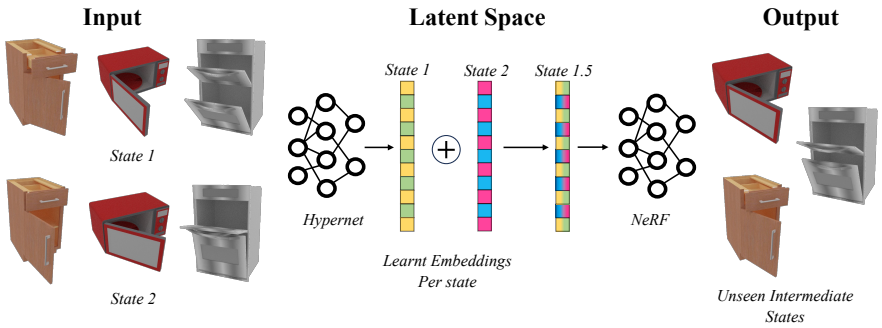

Fig. 1: Our method **LEIA**, takes in multi-view images of an object in four articulation states and is able to learn a view-invariant latent embedding for the state. We show that we can interpolate between the latents to generate any number of intermediate unseen states for the object using LEIA, given the camera position.

or assumptions regarding rigidity/shape of the object [17]. These approaches often fail to generalize for the long-tail of objects, especially in the case when a video of the object with articulation is unavailable, or in presence of large and multiple articulations in the object. In our work, we postulate that multi-view images of an object in different states can provide enough signal to model its 3D shape and articulation, even without priors from large-scale foundation models or information from videos. We achieve this by learning a generalizable, view-invariant latent embedding of the object for different states. We train these embeddings jointly with a hypernetwork that predicts weights of a NeRF [19] parameterizing the object state. The hypernetwork can be trained using multi-view images of discretized states of the object, each state representing a different object articulation. In Figure 1, we show three objects from our dataset in two different articulation states. The latent embeddings representing each state or articulation can be interpolated during test time to generate the weights of a NeRF that is able to reconstruct a never seen before state of the object.

Our key contributions can be summarized as following:

- We introduce an end-to-end method LEIA for generating novel states for articulated objects solely with multiview images captured at multiple states.
- We demonstrate that interpolating between embeddings can generate states of articulations of object not seen during training. The embedding space becomes interpolable with a manifold loss that encourages the latents to follow a structure that establishes a linear relationship between them, by minimizing the distance between the nearest neighbours in the latent space.
- Remarkably, LEIA achieves this without the need for any ground-truth 3D supervision, motion information, or articulation codes, establishing its versatility and effectiveness in capturing complex articulations.
- Our analyses demonstrate LEIA's robustness to single and multiple articulations, as well as combinations of motions. We can disentangle articulations

in different object parts if multiple are movable, making it scalable without constraints on the number of parts or motion types, unlike prior work.

2 Related Work

Neural Radiance Fields. Neural Radiance Fields [1, 19, 21, 42] have proven to be a massive success in modeling 3D scenes, due to the high fidelity of 3D reconstruction and novel view synthesis for static scenes. For dynamic scenes, Neural Volumes [18] utilizes an encoder-decoder voxel-based representation, complemented by an implicit voxel warp field. Occupancy Flow [24] tackled non-rigid geometry by assigning a motion vector to each point in both space and time, but requires full 3D ground truth supervision. Some of the first dynamic NeRF approaches [26, 28, 39] optimize an underlying volumetric deformable function and [15] conditions the NeRF on time. [16, 26, 45] followed this work, by learning a 5D spatiotemporal neural field. However, the approaches above usually require a video as input and do not handle well the case of large articulations in everyday objects given multiple states of objects as input.

3D Representations for Articulation. Due to the fine-grained nature of the task, deep neural network models for representing articulation require conditioning or refining parts of the architecture to suit the task. Category-specific reconstruction of deformable objects from images [7, 11–13], sparked interest in identifying and recovering the deformation in the 3D space [13, 22, 52]. Several works focus on shape reconstruction from videos [38, 47–49] that estimate a shape template for humans and animals. Other works that learn articulation from videos include Qian et al. [29] that detects and segments articulation planes from in-the-wild videos and [44, 53] which decouple the static and dynamic parts of videos. These works focus on modeling humans and animals where a large amount of data is available. Preliminary work in using images and shape started with A-SDF [28], which generates unseen articulations using Signed Distance Function [25], by providing an input shape and articulation code. CaDeX [14] is a unified representation for shape and motion, obtained from a point cloud input. Ditto [10] is a similar work that uses implicit representations for joint geometry and articulation modeling, with ground truth point clouds. CLA-NeRF [40] learns unseen articulated states from observing multi-view image input along with articulation information. Wei et. al [43] obtains an SDF-based articulable representation of common objects by feeding in images of articulated states across multiple categories, and CARTO [9] uses stereo images as input and uses a stereo encoder to infer the 3D shape, 6D pose etc. of multiple unknown objects. [4] focuses on manipulating object shapes to deform according to specified articulation commands. Moving away from 3D supervision and articulation annotations, Jiayi et al. [17] proposed PARIS, a method that is able to obtain unseen articulated states of an object, given multi-view images of just the start and end state of articulation. PARIS employs a composite rendering based approach, by decoupling the object into a static and moving part and then separately estimating and compositing the static and mobile neural radiance fields. In our work,

Table 1: Comparison of LEIA with existing methods. We show that LEIA is the first approach that does not use or learn any explicit prior along with not having any articulation input. This gives us flexibility in scaling to modeling articulations of objects with more than one part, and can thus handle a wide range of motion. We also have one universal model that can learn to represent both prismatic and revolute motion, unlike PARIS that has two separate models. 3D Sup. refers to "3D Supervision".

Method	Image/Shape Input	Dataset	Articulation Input	3D Sup.?	Prior	# States	# Parts
A-SDF [28]	Shape Code	Shape2Motion	Articulation Code	✓	Articulation	≥ 4	≥ 1
CLA-NeRF [40]	Multiview Images	PartNet-Mobility	Articulated Pose	✗	Articulation	≥ 4	1
NASAM [43]	Multiview Images++	PartNet-Mobility	N/A	✗	Category	≥ 4	≥ 1
CARTO [9]	Stereo Images	*Custom*	Joint Code	✓	Articulation	≥ 4	1
Ditto [10]	Point Clouds	Shape2Motion	Annotations	✓	Articulation	2	1
PARIS [17]	Multiview Images	PartNet-Mobility	N/A	✗	Motion (learned)	2	1
LEIA	**Multiview Images**	**PartNet-Mobility**	**N/A**	✓	**No**	**4**	$\geq$ **1**

we start LEIA with a similar input setting, by preparing multi-view renderings of objects in different articulation states. Unlike PARIS, we do not decouple the object into static and moveable parts, we rather learn to predict the weights of a single NeRF for any unseen articulated state using a state-modulated Hyper-Network. We aren't limited by the decoupling and the learned motion prior in the case of PARIS, which enables us to scale and learn any amount and kind of motion an object can possibly have, including combinations. We show a summed comparison of LEIA with prior work in Table 1.

Hypernetworks with Implicit Neural Representations. Hypernetworks, a specialized class of networks designed to predict parameters for another network, aim to achieve generalization across novel tasks [8]. In the realm of stylizing 3D scenes, [3] employed a hypernetwork for applying diverse styles, while scene reconstructions from limited data points were achieved by [35, 36]. Despite the promise shown in representation, these hypernetworks operate on input data points, necessitating test-time optimizations and rendering them unsuitable for compression tasks. Rather than using the provided data (image/video) as input for the hypernetwork, an alternative approach involves employing an auto-decoder framework. In this framework, a learnable latent, without the need for an encoder, represents a data point. This technique, applied by [33] to represent a dataset of videos, assigns each latent to a distinct video. While this method yields a representation for each set of frames, the lack of decoupling in spatial-temporal coordinates limits its scalability to real-world frames. On a related note, [34] extended a similar approach to 3D shapes and scenes, effectively acquiring a latent representation suitable for various downstream tasks. We use a similar framework for learning the latents corresponding to each articulated state in our method.

3 Method

Background. Our architecture is based on neural radiance fields, or NeRF [19], which parameterizes the radiance and volume density at a 3D location of a

scene as observed from a camera placed in a particular position using a neural network. $F_{\text{NeRF}} : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$, where $\mathbf{c} = (r, g, b)$, and σ represent the radiance and volume density, and $\mathbf{x}$ and $\mathbf{d}$ represent the 3D location and viewing direction respectively. To render a pixel, the radiance $C(\mathbf{r})$ of a camera ray $\mathbf{r}(s) = \mathbf{o} + s \cdot \mathbf{d}$ is integrated from near to far bounds s_n and s_f such that, $C(\mathbf{r}) = \int_{s_n}^{s_f} T(s)\sigma(\mathbf{r}(s))\mathbf{c}(\mathbf{r}(s), \mathbf{d})\, ds$, where the function $T(s)$ denotes the accumulated transmittance from s_n to s. To optimize the parameters of the MLP, a loss function is used that measures the discrepancy between the ground-truth and rendered images. Traditionally, the L2 loss is used for this purpose. However, to make the training more robust to outliers and to improve convergence, the Smooth L1 Loss can be employed as an alternative. The Smooth L1 Loss is a combination of L1 and L2 losses, behaving like L1 loss for large errors and like L2 loss for small errors. The loss function for the NeRF model using the Smooth L1 Loss is defined as:

$$\text{SmoothL1Loss}(x) = \begin{cases} 0.5 \cdot x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{1}$$

$$L_{\text{SmoothL1-NeRF}} = \sum_{r \in R} \text{SmoothL1Loss}\left(\hat{C}(\mathbf{r}) - C(\mathbf{r})\right) \tag{2}$$

where R is the set of rays used for sampling, and $\hat{C}(\mathbf{r})$ and $C(\mathbf{r})$ represent the ground-truth and rendered colors, respectively. By minimizing this loss function, the NeRF model learns to accurately reproduce the radiance of the scene, leading to high-quality image synthesis from novel viewpoints. While this formulation works well for generating novel scene viewpoints for a static scenes, it cannot handle dynamic scenes. [45] tried to address this shortcoming by making the MLP learn a spatiotemporal radiance field and used time, t, as an additional input. This works in principle but it becomes expensive to scale to lengthier videos due to constantly increasing sampling space for NeRF. We use the $L_{\text{SmoothL1-NeRF}}$ loss along with AdamW optimization to train LEIA, along with a L_{mask} that is the BCE loss between the predicted opacity and ground truth foreground loss.

3.1 Approach

Hypernetworks, or hypernets for short, are neural networks that generate weights for another neural network, known as the target network. Hypernets can be conditioned on various domains and can predict the weights for multiple neural networks simultaneously, if trained appropriately.

In this work, we use a hypernet to modulate a Neural Radiance Field (NeRF) based on the state of articulation, thereby learning a parametrization for each state. We employ a learnable latent embedding, Z, as input to this hypernet $h_l, l \in L$ where L is the number of layers of the hypernet. This approach allows us to not only modulate the NeRF but also create useful representations for each state. Our system can thus be represented as follows:

$$\begin{aligned} F_{\theta_t}(\mathbf{x}, \mathbf{d}) &= (\mathbf{c}_t, \sigma_t) \\ \theta_t &= h(z_t), z_t \in Z \end{aligned} \tag{3}$$

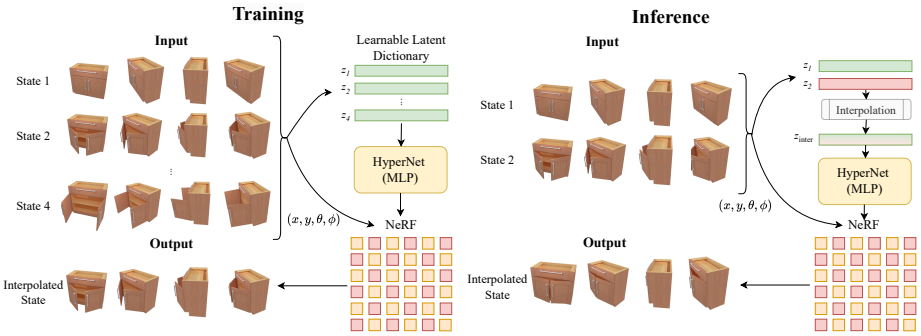

Fig. 2: Overview of our method. We take multi-view images in different states as input. A learnable latent dictionary based off an autoencoder learns an embedding per state id. The latent embedding is used as an input to the hypernet, that modulates and generates weights of the NeRF to reconstruct the state that is fed in. At inference time, we do a weighed interpolation of the learnt latents to obtain a corresponding newly generated intermediate state.

The latents Z, give us an additional ability to interpolate across seen states and generate novel states of the object, as we show in our work. The set of latents, Z, in Eq. (3) can be understood as a learnable dictionary where keys are the state ids. Formally, we represent this as

$$Z = \{\, t : \; z_t \mid t \in [0, 1, 2...T]\} \tag{4}$$

where, T is the total number of discretized states sampled for the object. Each of these states, z_t, is used as an input to the hypernet to get the parameterization for NeRF as shown in Figure 2. Once parameterized, the NeRF is trained using multi-view images of the object taken from various camera angles, providing a comprehensive view of the object's articulation in the current state. We sample only one state per batch.

Directly predicting the weights θ of the base network f_θ, using the hypernet h_l, is expensive, parameter-heavy, and unsuitable for compression. Hence, we follow [32, 37] and instead predict low-rank matrices, which are then applied to the base network weights. This type of modulation acts as a form of subnetwork selection, analogous to systems proposed in [6, 30]. For a base network f_θ with L layers, our formulation now looks like

$$\begin{aligned} f_\theta((\mathbf{x}, \mathbf{d}) | \theta_t^{l_1}, \theta_t^{l_2} ... \theta_t^{l_L}) &= \mathbf{c}_t, \sigma_t \\ \theta_t^l &= \eta(P^l \times Q^l) \circ \theta^l \\ h_l(z_t) &= [P^l, Q^l] \end{aligned} \tag{5}$$

where θ^l represents the weights of the l-th layer and θ_t^l denotes the modulated weights for frame t. Here, η signifies an activation function on the matrix-product of low rank matrices $\mathbf{P}^l \in \mathbb{R}^{K \times r}$, $\mathbf{Q}^l \in \mathbb{R}^{r \times K}$, where K is the width of the

base network f_θ and rank $r \ll K$. These matrices are responsible for adjusting the weights θ_l as dictated by the corresponding hypernetwork h_l. Note that all hypernetworks use the same latent $z_t \in \mathbb{R}^D$ as input. The rank r acts as a hyperparameter that controls the compression-performance trade-off. We further elaborate on details about model architecture and design choices for the hypernet, NeRF and the learnable latent dictionary in the supplementary.

3.2 Interpolation

Given an object with states t_1 and t_2, and a scale α, the task of state interpolation involves creating $\alpha - 1$ coherent states between the given states. In order to achieve this, we do a linear interpolation on the state latents z_1 and z_2 and pass the resulting latent through the hypernetwork. This gives us the weight modulation required in the NeRF, and the updated base network is used to obtain different viewpoints of the intermediate state.

$$\begin{aligned} z_{\text{inter}} &= (1 - \beta_i) \cdot z_t + \beta_i \cdot z_{t-1} \\ \mathbf{c}_{\text{inter}}, \sigma_{\text{inter}} &= f_{\theta_{\text{inter}}}(\mathbf{x}, \mathbf{d}; h(z_{\text{inter}})) \end{aligned} \tag{6}$$

$$\text{where, } \beta_i \in \left[\frac{1}{\alpha}, \frac{2}{\alpha}, ..., \frac{\alpha - 1}{\alpha}\right] \tag{7}$$

essentially generating $\alpha - 1$ states between any two given states. In our experiments, we quantitatively evaluate the result of the interpolation with unseen ground truth obtained from our dataset of all states, by linearly averaging the two extreme states present in the training, to obtain a latent for a state id.

3.3 Model Architecture

In our experiments, both the base network f_θ and hypernetworks h_l are simple MLPs that take in a coordinate input, and an id depicting the state of articulation. A network l_n is used to learn the latent dictionary for the states, which is a linear layer learnt in conjuction with the hypernet, and uses the `torch.nn.Embedding` class as a lookup table for the learnable latent embeddings, depicted by z_t for $t \in T$. This latent is fed to the hypernetwork h_l, that modulates the weights of the base network f_θ for reconstructing the output of the corresponding state id t. The base network f_θ is based off the NeRF architecture of Instant-NGP [21]. This architecture has separate fully-connected blocks for the geometry and texture of the scene, and we use separate hypernets to modulate each layer of these two FC blocks.

3.4 Optimization: Loss functions and regularizers

Latent Manifold Loss. We use the latent manifold loss function that enforces a structured latent space by encouraging local consistency among learnt latents. For each latent vector, the loss computes the average Euclidean distance between

the vector and its K nearest-neighbours on the manifold. This process enforces a smooth and continuous latent manifold, which is beneficial for models that rely on meaningful linear interpolations between the points on the manifold, and for tasks where the geometry of the latent space is crucial. Mathematically, we represent this loss for a particular state-id i as:

$$\mathcal{L}_{\text{manifold}}(\mathbf{l}_i) = \frac{1}{K} \sum_{k=1}^{K} \|\mathbf{l}_i - \mathbf{n}_k\|_2^2 \tag{8}$$

where $\mathbf{n}_k$ are the K nearest neighbors of $\mathbf{l}_i$ in the latent space, and $\|\cdot\|_2$ denotes the Euclidean (L2) norm. The loss is averaged over the selected latents and their nearest neighbors to ensure local uniformity in the manifold's geometry, where K is a hyperparameter that determines the number of neighbors considered.

Depth and Occlusion Regularization. Our depth and occlusion regularizations are designed to refine the clarity of rendered images by addressing occlusion and depth smoothness [23, 50]. The occlusion regularization loss, L_{occ}, aims to mitigate the obscuring of objects located beyond a specified depth threshold during rendering. This is accomplished by generating a binary mask, m_k, where the mask elements are set to 1 up to a certain index M reflecting the regularization range, and 0 thereafter. The occlusion loss is then articulated as the normalized sum of the product of the mask and the sampled density values σ_k along a ray, as given by

$$L_{\text{occ}} = \frac{1}{K} \sum_{k=1}^{K} \sigma_k \cdot m_k, \tag{9}$$

where K is the total number of sampled points on the ray. This formulation drives the model to prefer representations that reduce occlusions close to the camera, ensuring objects further away are not improperly concealed.

For depth continuity, the depth smoothness regularization loss, $L_{\text{DS}}(\theta, \mathbf{R})$, enforces the gradual transition of depth values among neighboring pixels, reducing sharp depth disparities that cause visual inconsistencies. If a ray r intersects with a single depth value, this value is evaluated directly. In contrast, for multiple depth values, the loss is the aggregate of squared differences between adjacent depth estimates. Formally, for depth predictions $\hat{d}_0(r)$, the loss is quantified as

$$L_{\text{DS}}(\theta, \mathbf{R}) = \sum_{r \in \mathbf{R}} \sum_{i,j=1}^{\text{Patch}-1} \left(\hat{d}_0(r_{ij}) - \hat{d}_0(r_{(i+1)j})\right)^2 + \left(\hat{d}_0(r_{ij}) - \hat{d}_0(r_{i(j+1)})\right)^2. \tag{10}$$

This measure is averaged across the rays to yield a measure of the depth map's smoothness. By integrating these regularizers into our training regimen, we significantly dampen the disturbances introduced by overlapping and obscured pixels, resulting in a more consistent interpolation between states. These benefits are substantiated through extensive ablation studies and quantitative analyses presented in the experiments.

Positional Encoding of the Latent. We incorporate positional encoding to capture the order of input elements, crucial for understanding articulation. Positional encoding injects the sequence with its inherent order, a key factor in articulation semantics. We employ the scheme from [41], which uses sine and cosine functions parameterized to encode varying frequencies. The computed positional encodings are added to the latent vector z_t, enriching it with semantic information that more accurately encodes the state of articulation. We analyze the effects of positional encoding on our model in the experimental section.

4 Experiments

4.1 Setup

Datasets. In this work, we use the PartNet-Mobility dataset [2, 20, 46], a large-scale synthetic collection of articulated objects in over 40 categories. The dataset has a variety of articulations defined, with objects comprising of single and multi-joint parts. Two types of motion, revolute (rotational) joint and prismatic (translation) articulation are represented in the dataset. To make up our training dataset, we choose 100 camera views that are arranged in a dome-like setup, capturing the upper hemisphere of the object. This is similar to the setup used by PARIS. We use the SAPIEN [46] library that the PartNet-Mobility dataset was released with, to render our RGB images at linearly spaced intervals of articulation to make up frames of a video that depicts the range of motion and obtain the camera parameters accordingly, converting it into the Blender [5] coordinate system to fit in our codebase. We use instances from 8 different common household item categories `storage, microwave, laptop, oven, washer, dishwasher, sunglasses, box` and choose 1-4 objects per category, bringing our total number of objects to 12. We also show the efficacy of our method on 68 images from a real-world scene of a chest of drawers, which are captured with a mobile phone and post-processed to remove background. Throughout LEIA, we train with a total of four states and interpolate between the two extreme states.

Baselines. The closest prior work with a setup similar to ours is PARIS, which takes in multi-view images of objects in two states and disentangles the static and moving part of the object. While we run experiments with four input states, we don't focus on learning a defined articulation for the object, instead we are able to recognize any arbitrary motion and combinations of motions given relevant states. We emphasize on the following differences:

1. PARIS uses separate models to train objects that have rotation and translation motion, respectively. While they have a method to estimate the type of motion, once determined, it is necessary to train on the appropriate model.
2. As PARIS also learns motion parameters for the articulation, given their pipeline of disentanglement of the static and moving parts of the object, this restricts the learning of motion parameters if there are multiple parts in the object, moving differently.

Table 2: Quantitative Results for Interpolated State Reconstruction. We compared our method with the PARIS baseline, trained on selected objects from the SAPIEN dataset. The results from three experiments are summarized below. The VanillaInt experiment involves simple interpolation of the latents. Our best-performing method, LEIA, introduces structure to the latents with a manifold loss and regularizers. Although PARIS learns a motion prior and LEIA implicitly performs state interpolation, both methods perform similarly for single-part objects. However, our approach excels with multi-part objects, outperforming PARIS significantly due to its flexibility in handling various motion types and articulations without constraints.

		Single-Part Articulation						Multi-Part Articulation						
		45135	7128	10211	101917	103778	12085	44781	45427	45575	101297	7187	102377	
Metrics	Methods	Storage1	Microwave	Laptop	Oven	Washer	Dishwasher	Storage2	Storage3	Storage4	Sunglasses	Oven	Box	Average
PSNR↑	PARIS	**28.66**	25.94	24.97	28.13	34.46	**27.24**	26.35	24.41	25.73	32.33	29.30	26.20	27.81
	VanillaInt	24.20	22.00	22.63	24.53	36.03	24.16	30.96	29.78	29.10	32.11	30.35	27.85	27.81
	LEIA	26.69	**26.38**	**25.04**	**28.71**	**36.14**	26.49	**31.07**	**29.78**	**29.80**	**35.60**	**30.80**	**28.05**	**29.55**
SSIM↑	PARIS	**0.99**	**0.97**	**0.97**	0.97	0.98	**0.96**	0.95	**0.95**	0.94	0.96	**0.96**	0.93	**0.96**
	VanillaInt	0.95	0.91	0.90	0.93	**0.99**	0.92	0.95	**0.95**	0.93	**0.97**	0.95	0.93	0.94
	LEIA	0.97	0.95	0.95	**0.98**	**0.99**	0.95	**0.96**	**0.95**	**0.95**	**0.97**	**0.96**	**0.95**	**0.96**
LPIPS↓	PARIS	**0.02**	**0.06**	0.19	0.03	0.03	**0.06**	0.05	0.05	**0.04**	**0.06**	**0.05**	**0.08**	**0.06**
	VanillaInt	0.05	0.10	0.17	0.03	0.02	0.08	**0.03**	**0.03**	0.05	0.09	**0.05**	0.13	0.07
	LEIA	0.03	0.07	**0.10**	**0.02**	**0.02**	0.07	**0.03**	**0.03**	**0.04**	0.09	**0.05**	0.12	**0.06**
CD ↓	PARIS	**0.18**	0.08	0.12	0.75	0.06	0.45	0.62	0.42	0.60	**0.20**	0.99	0.91	0.45
	VanillaInt	0.24	0.07	**0.06**	**0.10**	0.06	**0.20**	**0.48**	0.47	0.46	0.97	0.50	0.83	0.37
	LEIA	0.29	**0.06**	0.36	**0.10**	**0.05**	0.27	0.52	**0.38**	**0.41**	0.96	**0.35**	**0.62**	**0.36**

3. LEIA is motion-prior free, and we have a universal architecture that implicitly learns intermediate states, so we are able to train with multiple types of motion occurring in the same object, with no change in the code.

For comparing with PARIS, we run authors' original source code with our dataset, using the hyperparameters provided by the authors and training until convergence. We test our work and PARIS with the appearance quality of the reconstructed image of the interpolated state. We also set up a simple baseline that does just vanilla interpolation between the learnt latent embeddings, without enforcing any structure or constraints on them. We call this baseline VanillaInt and compare it with LEIA, which has been finetuned to do interpolation with the addition of the manifold loss and the depth and occlusion regularization for denoising the resulting output.

Quantitative Metrics. We use three appearance quality metrics, PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), and LPIPS (Learned Perceptual Image Patch Similarity). These are computed between reconstructed images of the interpolated state, and the unseen ground truth of the state from our dataset. We also report the Chamfer Distance to measure quality of 3D reconstruction by doing point sampling.

4.2 Main Results

Novel State Synthesis. We show the qualitative and quantitative results of interpolating our learned latent embeddings corresponding to the start and end states, in Table 2 and Figure 3. Our results show that the latent embeddings can

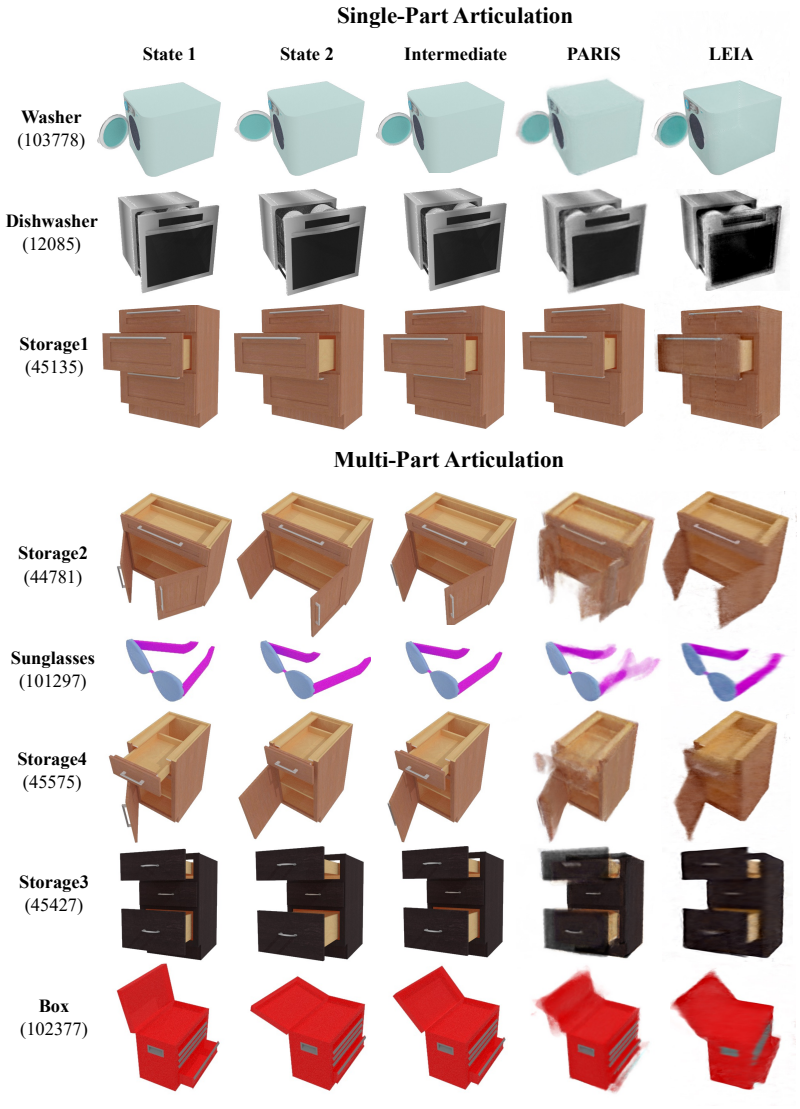

Fig. 3: Qualitative Results. We show results of PARIS and LEIA for reconstructing the unseen intermediate state, for both single and multiple articulations. We see that PARIS especially fails when there are two parts of the object moving differently, as the motion parameters are not registered correctly. LEIA handles this case successfully as it is not dependent on part disentanglement to identify and register articulation. LEIA also performs comparable to PARIS for single-part articulation, despite not having a dedicated model for the motion or part disentanglement.

be linearly interpolated, and have a structure in the latent space. We can generate any number of states between the start and the end state by doing a weighted

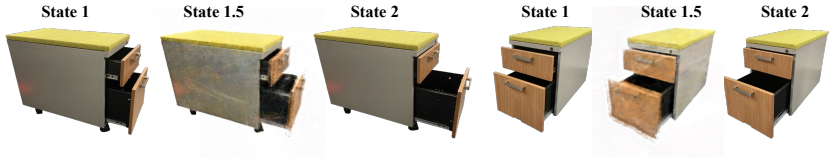

Fig. 4: Real World Results. LEIA is able to faithfully interpolate and reconstruct between two states of images from our real world data, proving its ability to generalize and work in an in-the-wild setting.

combination of the latents representing the start and the end states. For training, we choose four states, using the two additional states compared to PARIS to enforce structure in the latent space with the latent manifold loss, thereby making them amenable to interpolation. This helps us extend our formulation to multiple articulations without restriction. We also show results of our baseline VanillaInt and show that the numbers are boosted with the addition of the latent manifold loss and regularizers. For single-part objects, we perform comparably to PARIS despite having no part disentanglement and no dedicated model to represent the exact type of motion. For multiple-part articulations, we beat PARIS across all objects, as shown in Figure 3. Our comparable performance in 3D reconstruction shows that LEIA's learnt view-invariant latent embeddings are powerful in preserving spatio-temporal consistency. We also show that our method can work in the real world setting, in Figure 4. Ablations are shown for number of states, latent manifold loss, and depth and occlusion regularizations in Table 3 and Figure 6.

Analysis on Disentanglement of Joints. To check separability of the latent space, we train LEIA with multi-view images across multiple states, captured from a video that showed three parts of an object `Storage1`, move one after the other. We do a t-SNE based dimensionality reduction on the embeddings and plot them, shown in Figure 5. The latents exhibit clear separability, with the motion of each joint represented in a smooth trajectory, moving outward as the respective drawer in the object moves away from the starting "closed" state. This shows that the separated latent space is an indicator of multiple joints, positioning us to be capable of learning representations of multiple articulations.

4.3 Ablation Analysis

We perform ablations to tune various design choices, with results shown in Table 3 and Figure 6. The first experiment we performed was testing out the impact of the latent manifold loss. We notice qualitatively that without the manifold loss, our latent space lacked structure and the linear relationship between the embeddings was not captured. As shown in Figure 6, the intermediate state overfits to the extreme state without the latent manifold while the addition of the loss enables it to accurately capture the intermediate state. This figure also

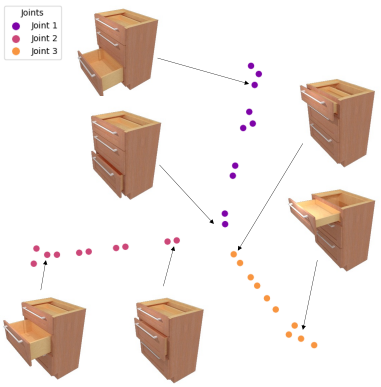

Fig. 5: t-SNE plot. After dimensionality reduction on jointly-learned state embeddings of an object with different moving parts. Our learned representations are separated and follow a smooth trajectory for each of the moving parts.

Table 3: Ablations. Our ablations reveal that latent manifold loss, depth, and occlusion regularization enhance LEIA's visual metrics. Opting for four states improved latent structure with the manifold loss.

Latent Manifold Loss	**PSNR**	**SSIM**	**LPIPS**
with	**29.40**	**0.95**	**0.05**
without	28.54	0.94	0.06
Depth Regularization	**PSNR**	**SSIM**	**LPIPS**
with	**29.63**	**0.96**	**0.05**
without	26.93	0.93	0.07
Occlusion Regularization	**PSNR**	**SSIM**	**LPIPS**
with	**29.64**	**0.95**	**0.05**
without	28.64	0.95	0.06
Positional encoding	**PSNR**	**SSIM**	**LPIPS**
with	27.11	0.94	0.06
without	**28.48**	**0.95**	**0.06**
Number of States	**PSNR**	**SSIM**	**LPIPS**
2	28.04	0.95	0.06
4	**29.69**	**0.96**	**0.05**

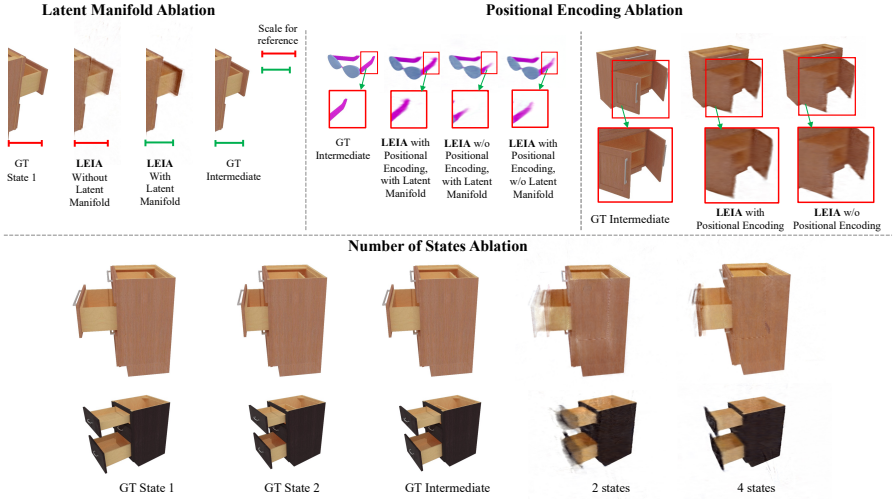

Fig. 6: Qualitative Results of Ablations. We show some qualitative results of using latent manifold loss, positional encoding and varying the number of states. Using the manifold loss prevents us from overfitting to the extreme states, and using four states help us interpolate with a lot of clarity. Positional encoding helps add missing information for thin parts, but for large objects it doesn't help much.

shows the effect of positional encoding which adds detail when the latent embedding fails to capture it, as evidenced in the results of `sunglasses`. This can

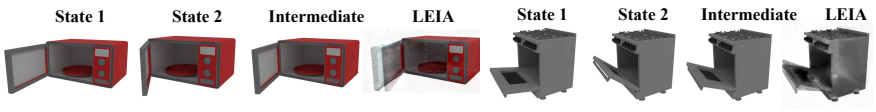

Fig. 7: Failure cases. Our model fails at at reconstructing the geometry correctly at camera positions where the two states we interpolate between have a change in the visible shape, like these examples where the microwave and oven being closed shows a deformation in the figure as compared to the open state, as the relationship between the motion is not easily captured in the latent due to the structure change.

happen for thin parts of the object that may not be not well-captured across camera views. Notably, the inverse happens in the case of large objects, where the positional encoding adds extra noise to the reconstruction, resulting in over-smoothing. We also show how adding just two more intermediate states in LEIA makes a huge difference in reconstruction results for both single and multi-part objects, enabling us to scale and be flexible. Quantitative numbers for depth and occlusion regularization ablation are shown in Table 3. However, they didn't universally help in the cases where the latent embedding already learned the state representation well enough but reduced noise when it appeared.

5 Limitations

While LEIA works in achieving good quality reconstruction of intermediate states, our latent embeddings do not yet ensure accurate physical consistency in the motion of the intermediates, which is a tradeoff we chose while opting for not decoupling the object and learning the motion parameters for the moveable part. This allowed us to scale our approach to interpolate between multiple moving parts of the object, which would be more representative of the real world where common objects can be articulated in multiple ways. LEIA also struggles when there is severe occlusion, as referenced by Figure 7, as the occlusion occurs at gaps in the learnt embeddings between two states.

6 Conclusion

In this work, we present LEIA, a method capable of successfully interpolating between two discrete articulation states of a deformable object with moving parts. LEIA handles multiple joints, including cases where moving parts are separated by static regions. It outperforms existing methods, especially when multiple parts move independently. The learned latent embeddings are view-invariant and separable, demonstrating LEIA's scalability and flexibility. We conducted comprehensive evaluations on synthetic and real data to investigate inherent challenges. Despite advancements, significant occlusion scenarios remain challenging. We hope future research builds upon our work.

Acknowledgements

This work was partially supported by NSF CAREER Award (#2238769) to Abhinav Shrivastava, and IARPA via Department of Interior/Interior Business Center (DOI/IBC) contract number 140D0423C0076. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, IARPA, DOI/IBC, or the U.S. Government.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5470–5479 (June 2022)
2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
3. Chiang, P.Z., Tsai, M.S., Tseng, H.Y., Lai, W.S., Chiu, W.C.: Stylizing 3d scene via implicit representation and hypernetwork. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1475–1484 (2022)
4. Chu, R., Liu, Z., Ye, X., Tan, X., Qi, X., Fu, C.W., Jia, J.: Command-driven articulated object understanding and manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8813–8823 (June 2023)
5. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018), `http://www.blender.org`
6. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. arXiv preprint arXiv:1803.03635 (2018)
7. Goel, S., Kanazawa, A., , Malik, J.: Shape and viewpoints without keypoints. In: ECCV (2020)
8. Ha, D., Dai, A.M., Le, Q.V.: Hypernetworks. In: International Conference on Learning Representations (2017), `https://openreview.net/forum?id=rkpACe1lx`
9. Heppert, N., Irshad, M.Z., Zakharov, S., Liu, K., Ambrus, R.A., Bohg, J., Valada, A., Kollar, T.: Carto: Category and joint agnostic reconstruction of articulated objects. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog. pp. 21201–21210 (2023)
10. Jiang, Z., Hsu, C.C., Zhu, Y.: Ditto: Building digital twins of articulated objects from interaction. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
11. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: ECCV (2018)
12. Kokkinos, F., Kokkinos, I.: To the point: Correspondence-driven monocular 3d category reconstruction. In: Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems (2021), `https://openreview.net/forum?id=AWMU04iXQ08`

13. Kulkarni, N., Gupta, A., Fouhey, D.F., Tulsiani, S.: Articulation-aware canonical surface mapping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 452–461 (2020)
14. Lei, J., Daniilidis, K.: Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6624–6634 (June 2022)
15. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., Lv, Z.: Neural 3d video synthesis from multi-view video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
16. Lin, H., Peng, S., Xu, Z., Yan, Y., Shuai, Q., Bao, H., Zhou, X.: Efficient neural radiance fields for interactive free-viewpoint video. In: SIGGRAPH Asia Conference Proceedings (2022)
17. Liu, J., Mahdavi-Amiri, A., Savva, M.: PARIS: Part-level reconstruction and motion analysis for articulated objects. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2023)
18. Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: Learning dynamic renderable volumes from images. ACM Trans. Graph. **38**(4), 65:1–65:14 (Jul 2019)
19. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
20. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
21. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). https://doi.org/10.1145/3528223.3530127, https://doi.org/10.1145/3528223.3530127
22. Neverova, N., Novotny, D., Khalidov, V., Szafraniec, M., Labatut, P., Vedaldi, A.: Continuous surface embeddings (2020)
23. Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S.M., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2022)
24. Niemeyer, M., Mescheder, L., Oechsle, M., Geiger, A.: Occupancy flow: 4d reconstruction by learning particle dynamics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
25. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
26. Park, K., Sinha, U., Barron, J.T., Bouaziz, S., Goldman, D.B., Seitz, S.M., Martin-Brualla, R.: Nerfies: Deformable neural radiance fields. ICCV (2021)
27. Park, K., Sinha, U., Hedman, P., Barron, J.T., Bouaziz, S., Goldman, D.B., Martin-Brualla, R., Seitz, S.M.: Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. arXiv preprint arXiv:2106.13228 (2021)
28. Pumarola, A., Corona, E., Pons-Moll, G., Moreno-Noguer, F.: D-NeRF: Neural Radiance Fields for Dynamic Scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)

29. Qian, S., Jin, L., Rockwell, C., Chen, S., Fouhey, D.F.: Understanding 3d object articulation in internet videos. In: CVPR (2022)
30. Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., Rastegari, M.: What's hidden in a randomly weighted neural network? In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11893–11902 (2020)
31. Saffran, J.R., Aslin, R.N., Newport, E.L.: Statistical learning by 8-month-old infants. Science **274**(5294), 1926–1928 (1996)
32. Schwarz, J.R., Tack, J., Teh, Y.W., Lee, J., Shin, J.: Modality-agnostic variational compression of implicit neural representations. arXiv preprint arXiv:2301.09479 (2023)
33. Sen, B., Agarwal, A., Namboodiri, V.P., Jawahar, C.: Inr-v: A continuous representation space for video-based generative tasks. arXiv preprint arXiv:2210.16579 (2022)
34. Sen, B., Singh, G., Agarwal, A., Agaram, R., Krishna, K.M., Sridhar, S.: Hyp-nerf: Learning improved nerf priors using a hypernetwork. arXiv preprint arXiv:2306.06093 (2023)
35. Sitzmann, V., Rezchikov, S., Freeman, B., Tenenbaum, J., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. Advances in Neural Information Processing Systems **34**, 19313–19325 (2021)
36. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. Advances in Neural Information Processing Systems **32** (2019)
37. Skorokhodov, I., Ignatyev, S., Elhoseiny, M.: Adversarial generation of continuous images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10753–10764 (2021)
38. Tan, J., Yang, G., Ramanan, D.: Distilling neural fields for real-time articulated shape reconstruction. In: CVPR (2023)
39. Tretschk, E., Tewari, A., Golyanik, V., Zollhöfer, M., Lassner, C., Theobalt, C.: Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In: IEEE International Conference on Computer Vision (ICCV). IEEE (2021)
40. Tseng, W.C., Liao, H.J., Lin, Y.C., Sun, M.: Cla-nerf: Category-level articulated neural radiance field. In: ICRA (2022)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
42. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. NeurIPS (2021)
43. Wei, F., Chabra, R., Ma, L., Lassner, C., Zollhoefer, M., Rusinkiewicz, S., Sweeney, C., Newcombe, R., Slavcheva, M.: Self-supervised neural articulated shape and appearance models. In: Proceedings IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
44. Wu, T., Zhong, F., Tagliasacchi, A., Cole, F., Oztireli, C.: D^2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 32653–32666. Curran Associates,

Inc. (2022), https://proceedings.neurips.cc/paper_files/paper/2022/file/d2cc447db9e56c13b993c11b45956281-Paper-Conference.pdf
45. Xian, W., Huang, J.B., Kopf, J., Kim, C.: Space-time neural irradiance fields for free-viewpoint video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9421–9431 (2021)
46. Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., Yi, L., Chang, A.X., Guibas, L.J., Su, H.: SAPIEN: A simulated part-based interactive environment. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
47. Yang, G., Sun, D., Jampani, V., Vlasic, D., Cole, F., Chang, H., Ramanan, D., Freeman, W.T., Liu, C.: Lasr: Learning articulated shape reconstruction from a monocular video. In: CVPR (2021)
48. Yang, G., Sun, D., Jampani, V., Vlasic, D., Cole, F., Liu, C., Ramanan, D.: Viser: Video-specific surface embeddings for articulated 3d shape reconstruction. In: NeurIPS (2021)
49. Yang, G., Vo, M., Neverova, N., Ramanan, D., Vedaldi, A., Joo, H.: Banmo: Building animatable 3d neural models from many casual videos. In: CVPR (2022)
50. Yang, J., Pavone, M., Wang, Y.: Freenerf: Improving few-shot neural rendering with free frequency regularization (2023)
51. Yao, C.H., Hung, W.C., Li, Y., Rubinstein, M., Yang, M.H., Jampani, V.: Lassie: Learning articulated shape from sparse image ensemble via 3d part discovery. In: NeurIPS (2022)
52. Yao, C.H., Hung, W.C., Li, Y., Rubinstein, M., Yang, M.H., Jampani, V.: LASSIE: Learning Articulated Shape from Sparse Image Ensemble via 3d part discovery. In: Advances in Neural Information Processing Systems (NeurIPS) (2022)
53. Yuan, W., Lv, Z., Schmidt, T., Lovegrove, S.: Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13144–13152 (2021)