

CLA-NeRF: Category-Level Articulated Neural Radiance Field

Wei-Cheng Tseng¹, Hung-Ju Liao¹, Lin Yen-Chen² and Min Sun^{1,3}

Abstract—We propose CLA-NeRF – a Category-Level Articulated Neural Radiance Field that can perform view synthesis, part segmentation, and articulated pose estimation. CLA-NeRF is trained at the object category level using no CAD models and no depth, but a set of RGB images with ground truth camera poses and part segments. During inference, it only takes a few RGB views (i.e., few-shot) of an unseen 3D object instance within the known category to infer the object part segmentation and the neural radiance field. Given an articulated pose as input, CLA-NeRF can perform articulation-aware volume rendering to generate the corresponding RGB image at any camera pose. Moreover, the articulated pose of an object can be estimated via inverse rendering. In our experiments, we evaluate the framework across five categories on both synthetic and real-world data. In all cases, our method shows realistic deformation results and accurate articulated pose estimation. We believe that both few-shot articulated object rendering and articulated pose estimation open doors for robots to perceive and interact with unseen articulated objects.

I. INTRODUCTION

Our living environment is full of articulated objects: objects composed of more than one rigid parts (links) connected by joints allowing rotational or translational motion, such as doors, refrigerators, scissors, and laptops. Endowing robots with the ability to perceive and interact with these objects requires a detailed understanding of the objects’ part-level poses, 3D shape, and materials. Prior works [1], [2] on estimating these properties of articulated objects often assume the object’s CAD model and thus cannot generalize to objects unseen during training.

To address this limitation, several recent works have explored category-level representations for articulated objects. These representations do not assume CAD models during testing and therefore can achieve intra-category generalization. For instance, ANCSH [3], designed specifically for articulated object pose estimation, uses the 3D coordinates in the canonical frame as the representation where the canonical frame is determined by authors who manually align the center and orientation of different CAD models. A-SDF [4], focusing on articulated object shape reconstruction, uses deep implicit signed distance function [5] as the representation and factors the latent space into shape codes and articulation angles. Although these representations have shown impressive results, both of them are limited by the requirement of access to ground truth 3D geometry during training, which is costly to scale up for articulated objects [6]. Furthermore, during testing, both works require depth images as inputs. This poses additional requirements on hardware

and may not work on articulated objects that are thin or highly reflective, e.g., scissors.

In this work, we seek to relax these requirements and build a category-level representation for articulated objects that doesn’t require 3D CAD models or depth sensing during both training and testing — only using RGB images with camera poses and part segmentation labels for training, and RGB images alone for testing. To this end, we introduce CLA-NeRF, a Category-Level Articulated NeRF representation that supports multiple downstream tasks including novel view synthesis, part segmentation, and articulated pose estimation. Our representation is based on Neural Radiance Fields (NeRF [7]), a method that has shown impressive performance on novel view synthesis of a specific scene by encoding volumetric density and color through a neural network. As NeRF typically requires a lengthy optimization process for each scene independently, we follow recent works [8], [9] to directly predict NeRFs from one or several RGB images in a feed-forward manner. However, simply doing so cannot capture articulated objects’ part attributes (e.g., part poses and segmentation) and joint attributes (e.g., joint axis). We, therefore, propose to explicitly model the object articulation by predicting a part segmentation field in addition to the volumetric density and color. Joint attributes can then be inferred by performing line-fitting on the part segmentation field.

In the experiments, we focus on modeling objects with revolute joints that cause 1D rotational motion (e.g., eyeglasses). We show that CLA-NeRF can render the object and its part segmentation map at unseen articulated poses by performing articulation-aware volume rendering. Additionally, it can perform category-level articulated pose estimation with RGB inputs by minimizing the residual between the rendered and observed pixels. We note that these tasks are not possible with existing NeRF formulations [7], [8], [9] which explicitly model the camera poses but don’t consider the object articulation. To the best of our knowledge, our work is the first to model general articulated objects with neural radiance fields.

We summarize our primary contributions as follows, and more information are provided in our project website¹:

- We propose CLA-NeRF, a differentiable representation for articulated objects that explicitly models the part and joint attributes. The proposed representation disentangles camera pose, part pose, part segmentation, and joint attributes, allowing us to independently control each property during rendering.

¹National Tsing Hua University

²Massachusetts Institute of Technology

³Appier Inc.

¹https://weichengtseng.github.io/project_website/icra22/index.html

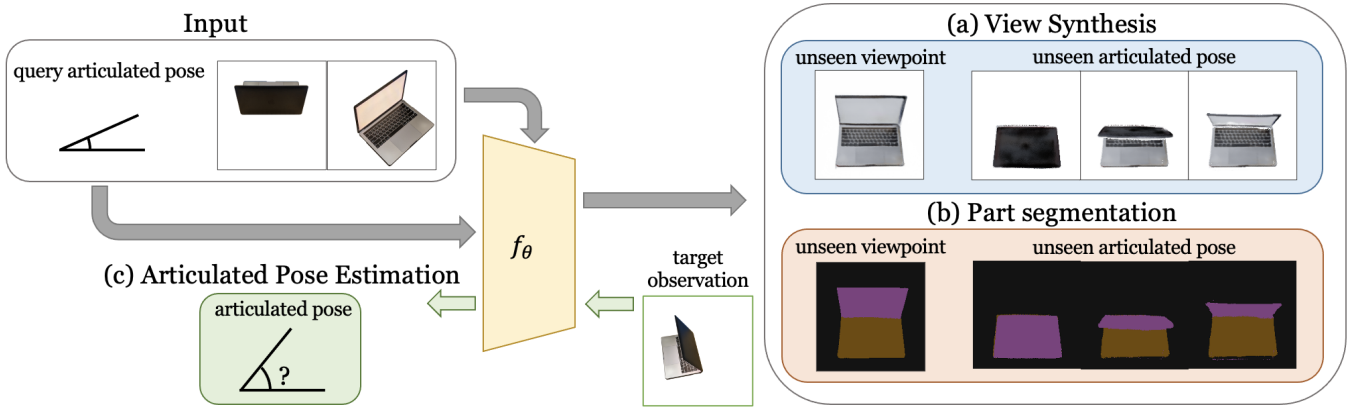


Fig. 1. We present a framework that takes a few visual observations and corresponding camera poses as input; then, we can perform (a) view synthesis and (b) part segmentation from unseen viewpoints and articulated poses. Moreover, (c) the articulated pose can be estimated via inversely optimizing the 3D deformation through our framework to match the target visual observation.

- We show that the proposed representation can perform category-level articulated pose estimation through analysis-by-synthesis with only RGB inputs. To the best of our knowledge, existing works for this task all require depth inputs [3], [10], [11], [12].

II. RELATED WORKS

A. Articulated 3D Shape Representations

Meshes and rigging techniques [13] are widely used to model the shape and deformation of articulated objects. Leveraging the abundant prior knowledge of human bodies, efficient techniques [14], [15], [16], [17], [18], [19], [20], [21], [22], [23] have been developed to model the deformation of a wide variety of body shapes. However, creating watertight meshes and rigs remains a labor-intensive process for general articulated objects whose part and joint attributes are less constrained. For the robotics community, it is very costly, if not impossible, to hire specially trained experts to model all sorts of articulated objects exist in our daily life. Recently, NASA [24] proposes to represent articulated shapes with a neural indicator function that successfully circumvents the complexity of meshes and the issue of watertightness. A-SDF [4] uses neural networks to encode signed distance function for articulated shape modeling. It’s trained on multiple instances of the same category and learns a disentangled latent space that allows it to synthesize novel shapes at unseen articulated poses. However, both of them require ground truth 3D models for training and thus still suffer from the scalability issue. Concurrently with our work, NARF [25] also proposes to explicitly consider articulation within NeRF and show impressive results on view synthesis of human bodies. Compared to NARF, our method differs in two aspects. First, our method focuses on general articulated objects and thus doesn’t assume known joint attributes (e.g., root joint’s pose, bone length) during test time. Instead, we infer them from the predicted segmentation field and further show results on articulated pose estimation. Second, our method uses RGB images and part segmentation labels as supervision, while NARF uses RGB images and joint attributes. We believe both works complement each other

and further supports the possibility that explicitly considering articulation within NeRF can lead to better generalization.

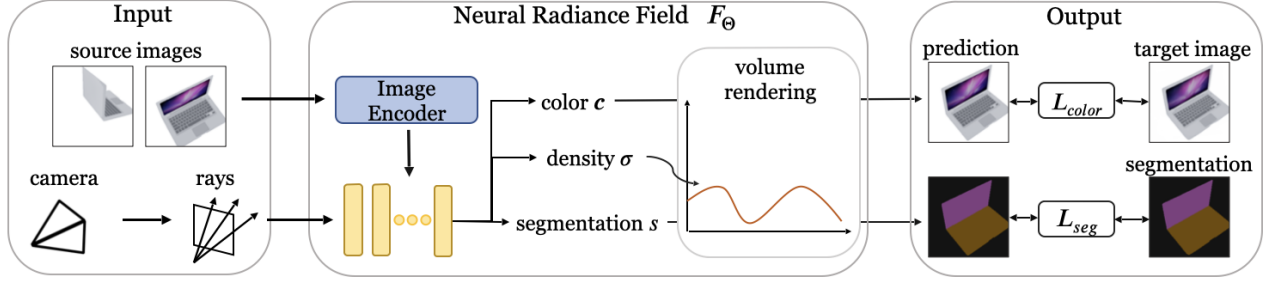
B. Articulated Object Pose Estimation

Most existing approaches for articulated object pose estimation requires instance-level information. They either assume the articulated object’s exact CAD model [1], [2] or need to generate the object’s motion through deliberate interaction before inference [26], [27], [28], [29], [30], [31]. Both directions require the robot to learn about each object from scratch, no matter how similar the object is to those it has previously experienced. To address this issue, recent works have proposed to predict canonicalized object coordinates [32] for category-level articulated object pose estimation [3], [10]. However, such representation is designed specifically for articulated pose estimation and can’t perform other tasks such as shape reconstruction or view synthesis. Additionally, it requires articulated objects’ ground truth 3D geometries for training and depth images for testing. As for inferring articulated pose from visual data, [12] proposed to use a mixture density network that consumes an RGB-D image to predict the probability of the joint attribute and articulated pose. ScrewNet[11] takes multiple depth images with different articulated poses and the same camera pose as input to predict joint attribute and articulated pose. [33] extended [34] by including reasoning about the applied actions along with the observed motion of the object while estimating its kinematic structure. Different from these works, we focus on building a category-level representation that only requires 2D supervision. Also, we demonstrate results on view synthesis besides articulated pose estimation.

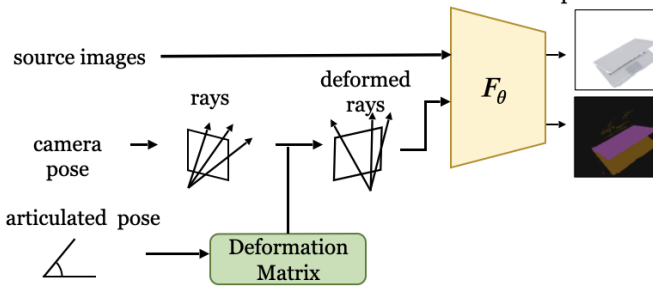
C. Preliminaries: NeRF

NeRF learns to synthesize novel views associated with unseen camera poses given a collection of RGB images with known camera poses. Specifically, NeRF represents a scene as a volumetric field of density σ and RGB color \mathbf{c} . The density models the shape of the scene and the color models the view-dependent appearance of occupied regions of the scene, both of which lie within a bounded 3D volume. A

(a) Training



(b) Deformation



(c) Estimate Articulate Pose

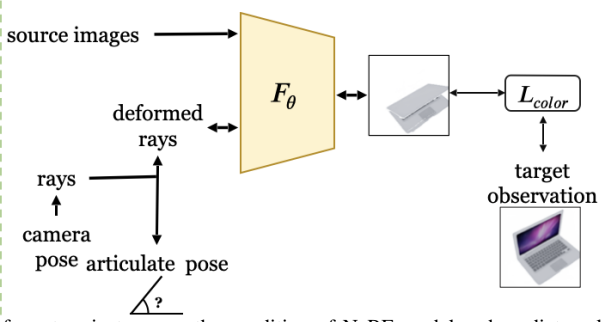


Fig. 2. The overview of our framework. (a) Our framework retrieves features from two instance as the condition of NeRF model and predicts color \mathbf{c} , density σ and segmentation \mathbf{s} . The volume rendering is applied to predict rendered results. (b) We calculate the deformation matrix based on the articulated pose. Then, we deform the sampled rays with the deformation matrix. Finally, the deformed visual image is rendered using our learned framework. (c) The articulated pose is estimated via inversely minimizing $\mathcal{L}_{\text{color}}$.

multilayer perceptron (MLP) parameterized by the weights Θ is used to predict the density σ and RGB color \mathbf{c} of each point by taking its 3D position $\mathbf{x} = (x, y, z)$ and unit-norm viewing direction \mathbf{d} as input, where $(\sigma, \mathbf{c}) \leftarrow F_{\Theta}(\gamma(\mathbf{x}), \mathbf{d})$ and $\gamma(\cdot)$ is a high-frequency positional encoding [35]. To render a pixel, NeRF emits a camera ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ from the camera center \mathbf{o} along the direction \mathbf{d} passing through that pixel on the image plane. Along the ray, K points $\{\mathbf{x}_k = \mathbf{r}(t_k)\}_{k=1}^K$ are sampled for use as input to the MLP which outputs a set of densities and colors $\{\sigma_k, \mathbf{c}_k\}_{k=1}^K$. These values are then used to estimate the color $\hat{\mathbf{C}}(\mathbf{r})$ of that pixel following volume rendering [36] approximated with numerical quadrature [37]:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^K \hat{T}_k (1 - \exp(-\sigma_k(t_{k+1} - t_k))) \mathbf{c}_k, \quad (1)$$

$$\text{with } \hat{T}_k = \exp(-\sum_{k' < k} \sigma_{k'}(t_{k'+1} - t_{k'}))$$

where \hat{T}_k can be interpreted as the probability of the ray successfully transmits to point $\mathbf{r}(t_k)$. NeRF is then trained to minimize a photometric loss $\mathcal{L} = \sum_{\mathbf{r} \in \mathbf{R}} \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2$, using some sampled set of rays $\mathbf{r} \in \mathbf{R}$ where $\mathbf{C}(\mathbf{r})$ is the observed RGB value of the pixel corresponding to ray \mathbf{r} in some image. To improve rendering efficiency one may train two MLPs: one “coarse” and one “fine”, where the coarse model serves to bias the samples that are used for the fine model. For more details, we refer readers to Mildenhall et al. [7].

While NeRF originally needs to optimize the representation for every scene independently, several recent works [8], [9], [38] on category-level NeRF have been proposed to directly predict a NeRF conditioned on one or few input

images.

III. METHOD

Although NeRF has shown impressive results on modeling the appearance of static objects, its formulation only allows control over the camera poses during rendering. Therefore, it cannot render a deformable articulated object (e.g., laptop) at different articulated poses (e.g., closing vs. opening) because it has more than 6 degree of freedom (DoF). CLA-NeRF is designed to tackle these issues. Instead of simply predicting colors \mathbf{c} and densities σ for each 3D location, we propose to additionally estimate part segmentation \mathbf{s} . Instead of only controlling the camera poses during rendering, our formulation allows user to input articulated poses. And instead of casting rays solely based on camera poses, we also transform camera rays based on query articulated poses, predicted part segmentation, and inferred joint attributes during volume rendering. These modifications together allow CLA-NeRF to render articulated objects at unseen articulated poses.

A. Category-Level Semantic NeRF

Here we first describe how we extend NeRF to predict part segmentation. For each 3D location \mathbf{x} and viewing direction \mathbf{d} , we add another linear layer on top of NeRF’s MLP backbone to predict part segmentation:

$$(\sigma, \mathbf{c}, \mathbf{s}) = F_{\Theta}(\gamma(\mathbf{x}), \mathbf{d}) \quad (2)$$

where \mathbf{s} is the segmentation logits with $P+1$ dimension (P parts and background).

With the volumetric field of predicted part segmentation, we can predict which part a pixel belongs to following the

procedure we used to approximate the volume rendering of RGB:

$$\hat{\mathbf{S}}(\mathbf{r}) = \sum_{k=1}^K \hat{T}_k (1 - \exp(-\sigma_k(t_{k+1} - t_k))) \mathbf{s}_k, \quad (3)$$

$$\text{with } \hat{T}_k = \exp(-\sum_{k' < k} \sigma_{k'}(t_{k'+1} - t_{k'}))$$

where \mathbf{s}_k is the predicted part segmentation of sampled point $\mathbf{r}(t_k)$. The new model can then be trained with both color loss $\mathcal{L}_{\text{color}}$ and segmentation loss \mathcal{L}_{seg} :

$$\mathcal{L}_{\text{color}} = \sum_{\mathbf{r} \in \mathbf{R}} \left[\|\hat{\mathbf{C}}_c(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2 + \|\hat{\mathbf{C}}_f(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2 \right] \quad (4)$$

$$\mathcal{L}_{\text{seg}} = - \sum_{\mathbf{r} \in \mathbf{R}} \left[\sum_{i=1}^{P+1} p^i(\mathbf{r}) \log \hat{p}_c^i(\mathbf{r}) + p^i(\mathbf{r}) \log \hat{p}_f^i(\mathbf{r}) \right] \quad (5)$$

where $\hat{p}^i(\mathbf{r}) = \frac{\exp(\hat{\mathbf{S}}^i(\mathbf{r}))}{\sum_{j=1}^P \exp(\hat{\mathbf{S}}^j(\mathbf{r}))}$

Here, $\mathbf{C}(\mathbf{r})$, $\hat{\mathbf{C}}_c(\mathbf{r})$ and $\hat{\mathbf{C}}_f(\mathbf{r})$ are the ground truth color, color predicted by the coarse network, and color predicted by the fine network for ray \mathbf{r} , respectively. In the segmentation loss \mathcal{L}_{seg} , p^i is the ground truth probability of part i , while \hat{p}_c^i and \hat{p}_f^i represent the probability predicted by the coarse and fine network for ray \mathbf{r} . In summary, the color loss $\mathcal{L}_{\text{color}}$ is the L2 distance between ground truth color and the color predicted by both coarse and fine networks, and the segmentation loss \mathcal{L}_{seg} is a multi-class cross-entropy loss that encourages the rendered semantic labels to be consistent with the provided labels. A coefficient λ is used to modulate these two losses during training: $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{color}} + \lambda \cdot \mathcal{L}_{\text{seg}}$.

We note that the current formulation still requires lengthy optimization for each articulated object and does not share knowledge between different objects. To make our method generalize to objects within the same category, we customize the framework of previous works [8], [9] to directly predict the proposed semantic NeRF given one or a few input images of the articulated object. For brevity, we explain the framework with a single input image I . First, we extract the image feature with an image encoder E to form a feature map $W = E(\mathbf{I})$. Then, we project each sampled 3D point \mathbf{x} to the input image plane and get the projected coordinate $\pi(\mathbf{x})$. Finally, we augment the input to NeRF F_Θ with the associated feature $W(\pi(\mathbf{x}))$, resulting in the following formulation:

$$(\sigma, \mathbf{c}, \mathbf{s}) = F_\Theta(\gamma(\mathbf{x}), \mathbf{d}, W(\pi(\mathbf{x}))) \quad (6)$$

The model is jointly trained on a collection of articulated objects belonging to the same category. Relative camera poses between multi-view images and the corresponding part segmentation labels are used for supervision.

B. Joint Attributes Estimation

In this work, we consider 1D revolute joints. The joint attributes consist of the direction of the rotation axis \mathbf{u} as well as a pivot point \mathbf{v} on the rotation axis. Given an input image of the articulated object, we propose to infer the joint

attributes from the predicted segmentation field through ray marching. For each pixel on the image plane, we cast a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ starting from the camera center \mathbf{o} along the direction \mathbf{d} passing through that pixel. We then sample K points $\{\mathbf{x}_k = \mathbf{r}(t_k)\}_{k=1}^K$ along the ray and feed them into the semantic NeRF to get their predicted density and part segmentation $\{\sigma_k, \mathbf{s}_k\}_{k=1}^K$. Since the 1D revolute joint lies at the intersection of two parts, we filter the sampled points to collect points that are close to the intersection:

$$\mathbf{X}_{\text{intersection}} = \{\mathbf{x}_k \mid \arg \max(\mathbf{s}_k) \neq \arg \max(\mathbf{s}_{k+1}) \wedge \sigma_k \geq H\} \quad (7)$$

where H is a predefined threshold to remove points with low density. After collecting $\mathbf{X}_{\text{intersection}}$ from all the pixels, we can perform linear regression on these 3D points to estimate both rotation axis \mathbf{u} and the pivot point \mathbf{v} .

C. Articulation-aware Volume Rendering

After predicting the part segmentation field and joint attributes \mathbf{J} , we discuss a modified volume rendering procedure that allows us to perform view synthesis at unseen articulated poses. Given an input articulated pose \mathbf{a} specified by users, we construct deformation matrices $\{\mathbf{D}^i(\mathbf{a}, \mathbf{J})\}_{i=1}^{P+1}$ that describe the rigid transformation between part i and the root part. During volume rendering, we deform the rays with each part's deformation matrix \mathbf{D}^i and collect the outputs for all $P+1$ parts:

$$\{\sigma^i, \mathbf{c}^i, \mathbf{s}^i = F_\Theta(\gamma(\mathbf{D}^i \mathbf{x}), \mathbf{d}, W(\pi(\mathbf{D}^i \mathbf{x})))\}_{i=1}^{P+1} \quad (8)$$

To merge these outputs for articulation-aware volumetric rendering, we weight all the fields with predicted part segmentation \hat{p} , where \hat{p}^i indicates the estimated probability of being classified as part i . The predicted color $\hat{\mathbf{C}}(\mathbf{r})$ and segmentation $\hat{\mathbf{S}}(\mathbf{r})$ are therefore the weighted sum of each part:

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{k=1}^K \hat{T}_k \sum_i^{P+1} \hat{p}^i(t_k) (1 - \exp(-\sigma_k^i(t_{k+1} - t_k))) \mathbf{c}^i(t_k) \quad (9)$$

$$\hat{\mathbf{S}}(\mathbf{r}) = \sum_{k=1}^K \hat{T}_k \sum_i^{P+1} \hat{p}^i(t_k) (1 - \exp(-\sigma_k^i(t_{k+1} - t_k))) \mathbf{s}^i(t_k) \quad (10)$$

where the accumulated transmittance is

$$\hat{T}_k = \exp(-\sum_{k' < k} \sum_i^{P+1} \hat{p}^i(t_{k'}) \sigma_{k'}^i(t_{k'+1} - t_{k'})) \quad (11)$$

D. Articulated Pose Estimation

Here we explain how we perform category-level articulated object pose estimation with CLA-NeRF. We assume the semantic NeRF F_Θ of an articulated object has already been predicted from source images and both the camera intrinsics and extrinsics are known. The goal is to estimate the articulated pose \mathbf{a} of a given input image I . Unlike CLA-NeRF's training procedure which optimizes Θ using image observations and part segmentations, we instead solve the

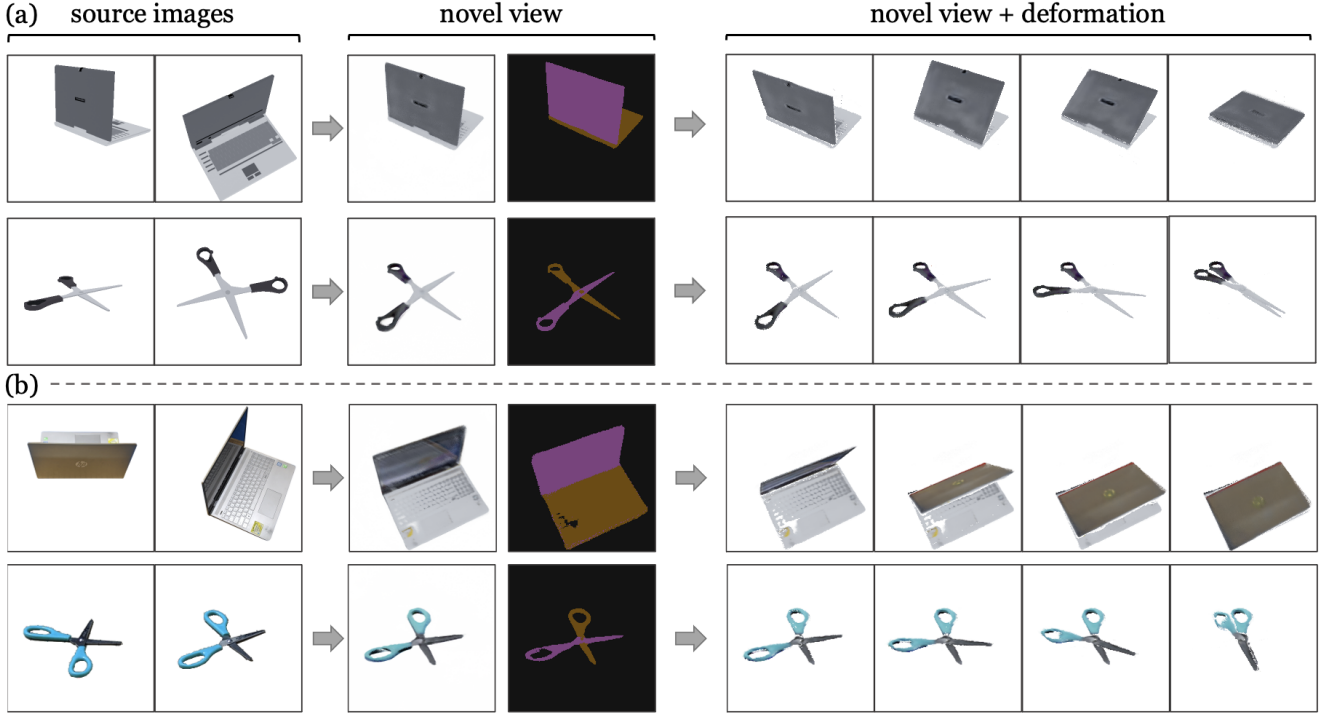


Fig. 3. The typical results on (a) synthetic data and (b) real-world data. We can find that the part in the object is consistently deformed with the joint parameter.

inverse problem [39] of recovering the articulated pose \mathbf{a} given the weights Θ and the image I :

$$\hat{\mathbf{a}} = \operatorname{argmin}_{\mathbf{a} \in \mathcal{A}} \mathcal{L}_{\text{color}}(\mathbf{a} | \Theta, \mathbf{d}, \mathbf{J}) \quad (12)$$

To solve this optimization problem, we iteratively perform gradient-based optimization to minimize the residuals between the rendered image and the observed image.

IV. EXPERIMENTS

We evaluate CLA-NeRF on three different tasks: view synthesis, part segmentation, and articulated pose estimation.

A. Dataset

a) Synthetic data: We consider the “laptop”, “scissors”, “eyeglasses”, “stapler” and “pliers” classes of SAPIEN [40], [41], [42] with 46, 54, 65, 24 and 23 instances respectively. We split these instances into two sets: training and held-out. We train on 200 observations of each training instance at a resolution of 200×200 pixels. Camera poses are randomly generated on a sphere with the object at the origin. Transparencies and specularities are disabled. We further render the held-out instances to construct a dataset for performance evaluation.

b) Real-world data: To further test our method, we manually collect real-world images and the corresponding camera poses for the “laptop” and “scissors” with articulate poses at $[0^\circ, 30^\circ, 60^\circ, 90^\circ]$.

B. View Synthesis and Part Segmentation

We show the qualitative results in Fig. 3 and quantitative results in Table I for the synthetic data. We find that CLA-NeRF successfully renders the held-out object at different

TABLE I

QUANTITATIVE RESULT INCLUDING SEGMENTATION, ARTICULATE POSE, AND NOVEL VIEW SYNTHESIS OF OUR FRAMEWORK EVALUATED ON THE DATASET GENERATED FROM SAPIEN[40].

	Novel View Synthesis				Segmentation	
	MSE↓	PSNR↑	SSIM↑	LPIPS↓	Pixel Acc↑	mIoU↑
Laptop	0.0811	23.89	0.94	0.1323	0.981	0.971
Scissors	0.0722	24.01	0.92	0.1456	0.989	0.969
Eyeglasses	0.0991	23.72	0.89	0.1755	0.973	0.941
Stapler	0.0771	26.91	0.96	0.1022	0.969	0.940
Pliers	0.0413	25.90	0.96	0.0711	0.971	0.940

TABLE II

QUANTITATIVE RESULTS FOR OUR REAL-WORLD DATA.

category	MSE↓	PSNR↑	SSIM↑	LPIPS↓
Laptop	0.1021	22.12	0.93	0.1600
Scissors	0.1281	23.39	0.92	0.1492

articulated poses. For the real data, we report the quantitative results in Table II and qualitative results in Fig. 3. The used metrics are MSE/PSNR/SSIM (higher is better) and LPIPS [43] (lower is better). We found that the network trained on synthetic data effectively infers the shape and texture of the real object, suggesting our model can transfer beyond the synthetic domain.

C. Articulated Pose Estimation

The results on synthetic data and real-world data are presented in Table III. Fig. 4 shows the L1 error with different articulate poses of source images and target images on the real-world dataset. We can first find that the error is the lowest when no deformation is required. Second, if the articulated poses are too different, the estimation will be

TABLE III

QUANTITATIVE RESULTS FOR ARTICULATED POSE ESTIMATION AND JOINT LOCALIZATION. WE SHOW THE POSE ERROR $\mathbf{a}_{\text{error}}$, ANGLE ERROR $\mathbf{u}_{\text{error}}$, DISTANCE ERROR $\mathbf{v}_{\text{error}}$. FOR THE REAL-WORLD RESULTS, PLEASE SEE SEC. IV-C.

Dataset	Synthetic									Real-World		
Approach	Ours			ScrewNet [11]			Abbatematteo et al. [12]			Ours		
	$\mathbf{a}_{\text{error}}$	$\mathbf{u}_{\text{error}}$	$\mathbf{v}_{\text{error}}$	$\mathbf{a}_{\text{error}}$	$\mathbf{u}_{\text{error}}$	$\mathbf{v}_{\text{error}}$	$\mathbf{a}_{\text{error}}$	$\mathbf{u}_{\text{error}}$	$\mathbf{v}_{\text{error}}$	$\mathbf{a}_{\text{error}}$ (sim2real)	$\mathbf{a}_{\text{error}}$ (generalize)	$\mathbf{a}_{\text{error}}$ (overfit)
Laptop	0.138	0.010	0.091	0.129	0.019	0.062	0.137	0.012	0.041	0.179	0.174	0.179
Scissors	0.130	0.016	0.120	0.116	0.149	0.136	0.131	0.037	0.041	0.179	0.170	0.170
Eyeglasses	0.151	0.109	0.071	0.141	0.140	0.136	0.149	0.108	0.082	-	-	-
Stalper	0.182	0.021	0.010	0.119	0.146	0.101	0.172	0.031	0.008	-	-	-
Pliers	0.171	0.010	0.010	0.121	0.132	0.102	0.183	0.009	0.009	-	-	-

TABLE IV

WE ALSO EVALUATE OUR APPROACH WITH POSE ERROR $\mathbf{a}_{\text{error}}$, ANGLE ERROR $\mathbf{u}_{\text{error}}$, DISTANCE ERROR $\mathbf{v}_{\text{error}}$ ON SHAPE2MOTION VALIDATION SET. NOTE THAT ANCSH [3] REQUIRES DEPTH TO ESTIMATE POSE.

Approach	Ours			ANCSH [3]		
	$\mathbf{a}_{\text{error}}$	$\mathbf{u}_{\text{error}}$	$\mathbf{v}_{\text{error}}$	$\mathbf{a}_{\text{error}}$	$\mathbf{u}_{\text{error}}$	$\mathbf{v}_{\text{error}}$
Laptop	0.179	0.011	0.110	0.169	0.009	0.017
Eyeglasses	0.169	0.109	0.091	0.076	0.039	0.016

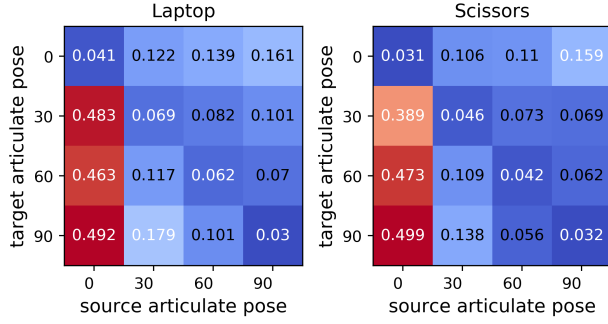


Fig. 4. Error heatmap of articulate pose estimation.

less accurate. It is because we only optimize articulated pose with $\mathcal{L}_{\text{color}}$ and local minimum occur during the optimization process.

To test the limit of our method, we compare our method with ScrewNet [11] and Abbatematteo et al. [12] on our dataset (Table III). Besides, we also evaluate CLA-NeRF on the Shape2Motion dataset without fine-tuning. The results compare against ANSCH [3] are shown in Table IV. Despite not using depth images as inputs and not finetuned, we find our model to only perform slightly worse than ANSCH [3] and ScrewNet [11] and [12]. It shows that the proposed representation is a promising direction for category-level articulated pose estimation.

To understand whether the articulated pose estimation can be improved, we finetune the model in two manners. First, we finetune the framework with a set of real-world objects, then, we test it on unseen real-world objects. Second, we directly finetune the framework on specific object and test on it. These fine-tuning approaches are labeled as **generalize** and **overfit** in Table III, respectively. Only minor improvement is observed.

D. Failure Cases

Despite the promising results shown in Sec. IV-B and Sec. IV-C, there are some failure cases that need to be discussed.

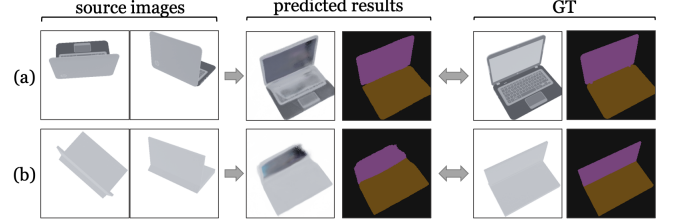


Fig. 5. Failure cases. (a) Incorrect keyboard appearance due to missing observation in source images. (b) Incorrect geometry due to lack of texture.

First, since our framework only takes few instances as conditions, if query camera poses are highly distinct from the source camera pose, the appearance may have some defect. From Fig. 5 (a), the predicted screen color is not the same as the ground truth. However, the geometry of the object and its part segmentation are reconstructed correctly, so the articulated pose estimation isn't affected by this issue. Besides, despite the color of the screen isn't accurate, the appearance still matches the normal appearance of the laptop screen. Second, for the CAD model without texture, it is hard for our framework to correctly infer its geometry. From Fig. 5 (b), we find that the shape of the screen is distorted. Besides, our joint localization is inaccurate if the source images are closed laptop. It is because the NeRF model the two part of laptop attach with each other, so the intersection points form a surface. Therefore, the joint axis can't be estimated correctly.

V. CONCLUSION

We propose a framework that only takes few instances of an articulate object with different viewpoints as references; then, infers the corresponding deformable neural radiance field to predict the image and part segmentation with the specified camera pose. With the well-trained framework, the articulate pose of an object can be estimated via inversely optimize the deformation condition. In the experiments, we evaluate the framework in both synthesis objects collected from SAPIEN and our manually collected real-world data. In all cases, our method shows realistic deformation results and accurate articulated pose estimation.

VI. ACKNOWLEDGE

This project is funded by Ministry of Science and Technology of Taiwan (MOST 109-2634-F-007-016) and supported by NOVATEK Fellowship.

REFERENCES

- [1] Karthik Desingh, Shiyang Lu, Anthony Opipari, and Odest Chadwicke Jenkins. Factored pose estimation of articulated objects using efficient nonparametric belief propagation. In *ICRA*, 2019.
- [2] Frank Michel, Alexander Krull, Eric Brachmann, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Pose estimation of kinematic chain instances via object coordinate regression. In *BMVC*, 2015.
- [3] Xiaolong Li, He Wang, Li Yi, Leonidas Guibas, A. Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *CVPR*, 2020.
- [4] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-SDF: Learning disentangled signed distance functions for articulated shape representation. *arXiv preprint arXiv: 2104.07645*, 2021.
- [5] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.
- [6] Ankur Handa, Andrey Kurenkov, and Miles Brundage. Part I: Indexing datasets of 3d indoor objects. <https://sim2realai.github.io/Synthetic-Datasets-of-Objects-Part-I/>, 2019.
- [7] 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. In *ECCV*, 2020.
- [8] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, 2021.
- [9] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021.
- [10] Yijia Weng, He Wang, Qiang Zhou, Yuzhe Qin, Yueqi Duan, Qingnan Fan, Baoquan Chen, Hao Su, and Leonidas J. Guibas. Captra: Category-level pose tracking for rigid and articulated objects from point clouds. *arXiv preprint arXiv:2104.03437*, 2021.
- [11] Ajinkya Jain, Rudolf Lioutikov, Caleb Chuck, and Scott Niekum. Screwnet: Category-independent articulation model estimation from depth images using screw theory. In *arXiv preprint*, 2020.
- [12] Ben Abbatematteo, Stefanie Tellex, and George Konidaris. Learning to generalize kinematic models to novel objects. In *Proceedings of the Conference on Robot Learning*, 2020.
- [13] Alec Jacobson, Zhigang Deng, Ladislav Kavan, and JP Lewis. Skinnet: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*, 2014.
- [14] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 2015.
- [15] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *ICCV*, 2019.
- [16] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.
- [17] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *CVPR*, 2021.
- [18] Muhammed Kocabas, Nikos Athanasios, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020.
- [19] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.
- [20] Jason Y. Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3d human-object spatial arrangements from a single image in the wild. In *ECCV*, 2020.
- [21] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In *3DV*, 2018.
- [22] Wei-Cheng Tseng, Jin-Siang Lin, Yao-Min Feng, and Min Sun. Toward robust long range policy transfer, 2021.
- [23] Wei-Cheng Tseng, Wei Wei, Da-Cheng Juan, and Min Sun. Metacpr: Generalize to unseen large number of agents with communication pattern recognition module. *CoRR*, abs/2112.07222, 2021.
- [24] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. In *ECCV*, 2020.
- [25] Atsuhiko Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. *arXiv preprint arXiv:2104.03110*, 2021.
- [26] Dov Katz and Oliver Brock. Manipulating articulated objects with interactive perception. In *2008 IEEE International Conference on Robotics and Automation*, pages 272–277. IEEE, 2008.
- [27] Dov Katz, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 5003–5010. IEEE, 2013.
- [28] Roberto Martín Martín and Oliver Brock. Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In *IROS*, 2014.
- [29] Roberto Martín-Martín, Sebastian Höfer, and Oliver Brock. An integrated approach to visual perception of articulated objects. In *ICRA*, 2016.
- [30] Karol Hausman, Scott Niekum, Sarah Osentoski, and Gaurav S Sukhatme. Active articulation model estimation through interactive perception. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3305–3312. IEEE, 2015.
- [31] Sudeep Pillai, Matthew R. Walter, and Seth Teller. Learning articulated motions from visual demonstration, 2015.
- [32] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019.
- [33] Ajinkya Jain and Scott Niekum. Learning hybrid object kinematics for efficient hierarchical planning under uncertainty. In *IROS*, 2020.
- [34] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects, 2011.
- [35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *NeurIPS*, 2017.
- [36] James T. Kajiya and Brian P. Von Herzen. Ray tracing volume densities. *SIGGRAPH*, 1984.
- [37] Nelson Max. Optical models for direct volume rendering. *IEEE TVCG*, 1995.
- [38] Konstantinos Rematas, Ricardo Martin-Brualla, and Vittorio Ferrari. ShaRF: Shape-conditioned radiance fields from a single view. In *ICML*, 2021.
- [39] Lin Yen-Chen, Pete Florence, Jonathan T. Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. iNeRF: Inverting neural radiance fields for pose estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [40] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *CVPR*, 2020.
- [41] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
- [42] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, June 2019.
- [43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.