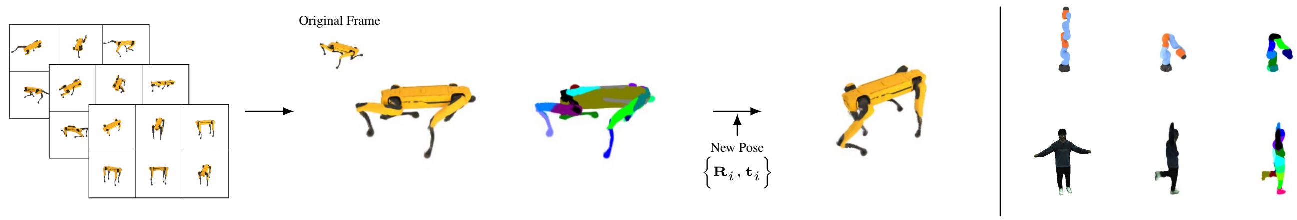
Watch It Move: Unsupervised Discovery of 3D Joints for Re-Posing of Articulated Objects

Atsuhiro Noguchi[†, ‡*] Umar Iqbal[†] Jonathan Tremblay[†] Tatsuya Harada[‡,§] Orazio Gallo[†]
[†]NVIDIA [‡]The University of Tokyo [§]RIKEN



(a) Multi-View Video (b) RGB and Parts Reconstruction (c) Re-Posing (d) Other Categories

Figure 1: Our method learns to render novel views of an articulated, moving object by "watching" it move in a multi-view video sequence with associated foreground masks. Simultaneously, it discovers the object's parts and joints with *no additional supervision*. The learned structure allows us to *explicitly* re-pose the object, by roto-translating each part around its joint. In (c) and (d) we re-pose objects from multiple categories to configurations never seen in training, which is possible thanks to the structure we discover from the input videos.

Abstract

Rendering articulated objects while controlling their poses is critical to applications such as virtual reality or animation for movies. Manipulating the pose of an object, however, requires the understanding of its underlying structure, that is, its joints and how they interact with each other. Unfortunately, assuming the structure to be known, as existing methods do, precludes the ability to work on new object categories. We propose to learn both the appearance and the structure of previously unseen articulated objects by observing them move from multiple views, with no joints annotation supervision, or information about the structure. We observe that 3D points that are static relative to one another should belong to the same part, and that adjacent parts that move relative to each other must be connected by a joint. To leverage this insight, we model the object parts in 3D as ellipsoids, which allows us to identify joints. We combine this explicit representation with an implicit one that compensates for the approximation introduced. We show that our method works for different structures, from quadrupeds, to single-arm robots, to humans. The code is available at https://github.com/NVlabs/watch-it-move *and a version of this manuscript that uses animations is at* https://arxiv.org/abs/2112.11347.

1. Introduction

Using images to infer both the appearance and the functional structure of generic, real-world objects is a fundamental goal of computer vision. From a practical standpoint, it would allow us to render and manipulate physical objects in the *metaverse*. But its appeal goes further, as it requires pushing the boundaries of our ability to learn from data with no direct supervision.

Our community made dramatic progress towards appearance capture and novel view synthesis, particularly for static scenes [1, 6, 29, 35, 45, 65, 68]. Several recent methods can also capture dynamic scenes and *reenact* their motion [28, 39, 43, 54, 56]. We use the term "reenacting" to highlight that these methods cannot explicitly control the pose of the dynamic objects. Rather, they replay through the poses that were observed. *Re-posing* an articulated object—*i.e.*, the explicit manipulation of its pose—requires knowing the location of the joints and how the different parts of the object interact with each other[1]. Learning to predict the location of joints in 3D is a well-studied task, at least for humans, and it is generally tackled using 2D [17,18,22,44,55,63] or 3D [16,20,25–27,50,70] ground truth information. When not using joints supervision, existing pose manipulation methods rely on a predefined model, that is, a template structure [23, 48]. However, annotations are expensive and object-specific, which is why they are only available for limited classes of objects, such as people or faces [15, 41, 47].

We aim at re-posing an articulated object from a category not seen before, using only a multi-view video and corresponding foreground mask, as shown in Figure 1. Our approach requires no additional supervision, no prior knowledge about the structure, nor networks pre-trained on aux-

*Work partially done when Atsuhiro Noguchi was an intern at NVIDIA.

[1] Image-to-image translation methods (e.g., [34]) can also re-pose, but we focus on methods that allow for the explicit definition of the target pose.

iliary tasks: we learn the appearance and the structure of the object by just watching it move. Like existing methods [8, 36], to express explicit pose changes, we treat the articulated object as a set of posed parts, each connected to other parts through joints. However, rather than relying on direct supervision, we note that a joint is a 3D point around which a part must rotate to produce the piece-wise, rigid deformation observed in the input images. This allows us to get indirect supervision for the locations of the joints from the image reconstruction loss.

Our approach, inspired by neural implicit representations, is scene-specific and predicts the color and the signed-distance function (SDF) of any 3D point, allowing us to generate any desired frame by volumetric rendering [57]. We also learn certain properties of the object *explicitly*. Specifically, we model the object as a set of ellipsoids. A functional part of the object can be represented by one or more ellipsoids, as shown in Figure 2. We optimize the geometric properties of the ellipsoids, *i.e.*, their size and pose, for each frame of the input sequence. The color and density of a 3D point, then, can be predicted from the combined contribution of the ellipsoids. Because these ellipsoids only afford a coarse approximation of the object, we also estimate a residual with respect to this explicit part-based representation. In addition to regularizing the optimization landscape, this representation provides a key advantage: the relative motion of the parts can be explicitly observed over time, which offers clues on the locations of the joints. Note that this applies to unobserved categories, and requires no prior knowledge on the number of parts that compose it. Because we do not use any prior on the structure of the object or supervision annotations, our method can re-pose any articulated object from a single multi-view video sequence and the corresponding foreground masks. The pose of the object can be manipulated by applying the appropriate roto-translation to the different joints. Figures 1(c) and (d) show examples of object re-posing for different categories, structures, and number of parts—all of which were unknown at training time. Our method

- is the first to learn a re-poseable shape representation from multi-view videos and foreground masks, without additional supervision or prior knowledge of the underlying structure,
- it discovers the number and location of physically meaningful joints—also learned with no annotations, and
- it is structure agnostic and can thus be learned for previously unseen articulated object categories.
- Our reconstruction and re-posing results are on par or better than those of category-specific methods that use prior knowledge.

2. Related Work

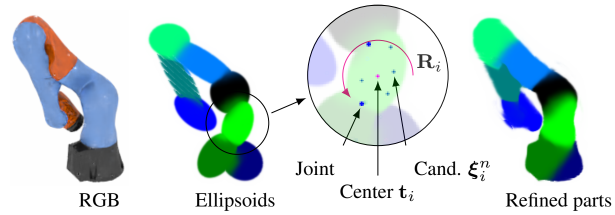


Figure 2: We explicitly represent each object's part as an ellipsoid centered at $\mathbf{t}_i$ and oriented with $\mathbf{R}_i$ (magenta arrow). We identify the part's joints from a pool of candidates, $\boldsymbol{\xi}_i^n$. The final reconstruction is obtained by estimating a residual w.r.t. the ellipsoids.

2.1. Object Re-Posing and Novel-View synthesis

Synthesizing images of articulated objects under novel poses and viewpoints is critical to several applications. Earlier methods formulated the problem as conditional image-to-image translation [3,9,31,34,42,62,64,69]. Given an image of an object and a target pose, these methods use a generator model to transfer a given image to a target pose. The conditioning pose is usually obtained from 2D keypoints or parametric meshes. However, keypoints or mesh models are available for handful of object categories (*e.g.*, faces, human body, and hands), preventing these methods from generalizing to arbitrary object classes.

More recently, NeRF [35] ignited a wave of research on synthesizing novel views of an object by using a sparse set of multi-view images [1, 29, 35, 38, 57, 65]. These methods learn an implicit 3D representation that provides the color and density of each point in 3D space. Photorealistic images can then be generated using volumetric rendering. Since the implicit 3D representation they use is continuos and topology-agnostic, these methods can reconstruct arbitrary, *static* objects. Many follow-up works extend NeRF to model dynamic scenes, using single- or multi-view videos for training [28, 39, 43, 54, 56]. However, these methods only "reenact" the video used for training, and do not offer control over the articulated pose. We build on these developments and propose a method that also provides control over the articulated pose of the objects.

To allow re-posing the objects, various implicit representations for articulated objects have also been proposed, especially for humans. They allow novel view and pose synthesis, but require ground truth poses [8,36,49], or dense 3D meshes [24, 30, 40, 41, 53] annotations for the training image. In contrast, we propose a re-poseable 3D implicit representation trained only from multi-view videos and foreground masks of a previously unseen object category. We simultaneously decompose the parts, estimate the connections between them, and reconstruct the image with no prior information about the structure of the object.

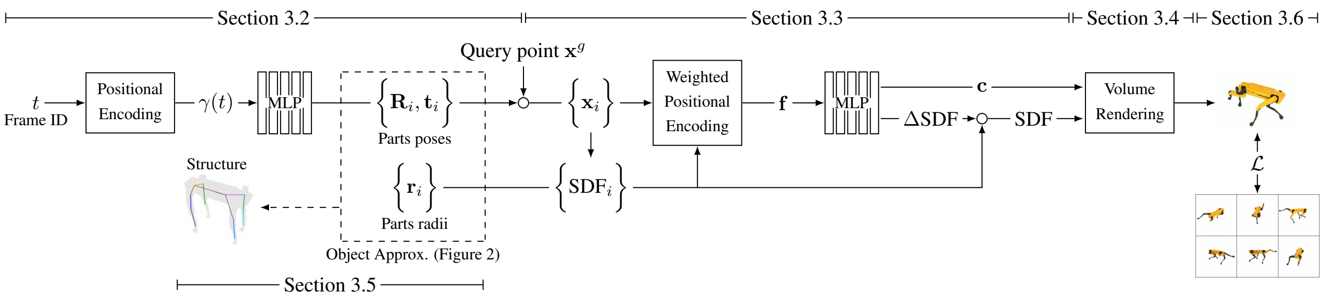


Figure 3: Given a multi-view video of an articulated, dynamic object, an MLP associates a frame id with the configuration of the object's parts, approximated with ellipsoids. To predict the color of a 3D query point in global coordinates, we assemble a feature $\mathbf{f}$ from its coordinates and signed-distance functions, expressed with respect to all of the parts. From $\mathbf{f}$, a second MLP produces the color and a residual SDF to perform volume rendering. Simultaneously, from the parts we discover the structure. We train the system end to end, thus back-propagating the error all the way to the poses and radii of the parts. For clarity, some components and loss terms are not shown here.

2.2. Discovery of 3D Joints of Articulated Objects

Explicitly re-posing an object is straightforward if the joints locations are given, but localizing the 3D joints is challenging. Ground-truth 3D joints supervision simplifies the problem [16, 20, 25–27, 50, 70]. However, 3D annotations are expensive to gather and, perhaps more importantly, they make the resulting algorithms category-specific. Other works simplify the problem and rely on 2D annotations and multi-view [17, 18, 22, 44, 55, 63] or temporal [37] information for 3D supervision. Although 2D joints are cheaper to annotate, the process is still time-consuming and hard to scale to a large number of objects and classes. To address this, some recent methods aim to discover the joints of articulated objects using self-supervised learning [19, 23, 48]. While these methods show impressive results, they still rely on carefully designed, object-specific templates and/or prior information, which cannot be directly applied to other object classes. Other methods can handle arbitrary objects but provide only 2D landmarks [7, 33, 52, 67]. There exist some works that discover 3D keypoints using self-supervision and multi-view data, but they are limited to rigid objects [51]. Other works estimate the parts and the structure of arbitrary objects from videos, but they require ground truth 2D trajectory of keypoints [10, 59]. In contrast to these methods, our approach discovers 3D joints of articulated objects and does not require any 2D or 3D annotations, predefined template, or any other prior knowledge about the object, which makes it category-agnostic. Additionally, most of these methods only provide locations for a sparse set of keypoints/joints and do not provide any information about the surface geometry or texture of the object. While some methods provide self-supervised dense part labels, they are limited to 2D information [14, 46]. In contrast, our re-posable shape representation provides dense part labels, 3D surface geometry, as well as the texture of each part (implicitly).

Our work is also related to the recent methods for 3D shape representation that use implicit functions [2, 11, 12], which represent a deformable 3D shape as a composition of simple shape elements, *e.g.*, 3D Gaussians (where the level set is an ellipsoid). Each element contributes to the implicit surface of the shape. However, these methods cannot learn surface textures, they require the ground truth 3D shape for training, and do not learn the physical connectivity between parts, which prevents explicit re-posing. Concurrent methods circumvent the need of 3D shape supervision, but do not allow for explicit re-posing [60, 61].

3. Method

3.1. Overview

Our method takes as input a sequence of T multi-view, posed images of an articulated, moving object, $\mathcal{O}$, and foreground masks indicating its silhouette. From those, we learn to render novel views of $\mathcal{O}$. We also discover plausible joints that allow us to render $\mathcal{O}$ in a new pose and from a novel viewpoint, *without any additional supervision*. We use a hybrid representation of the object that combines an *explicit* rough approximation of its body, and a subsequent *implicit* refinement. More concretely, we represent $\mathcal{O}$ explicitly as a set of P parts, each approximated with an ellipsoid, see Figure 2. Rather than assuming P to be known, we over-segment the object and subsequently merge the different parts as needed (Section 3.5.2). The ellipsoid representing part i is parametrized with its three-dimensional radius $\mathbf{r}_i$. (Throughout the paper, we use bold for vectors and matrices.) Its pose at time t is represented by the translation of its center of mass, $\mathbf{t}_i(t)$, and a rotation matrix, $\mathbf{R}_i(t)$. To discover and localize the object's joints, which define the relationship between different parts, we observe that a 3D point is a meaningful joint if roto-translating a part around it explains a pose change in the reconstructed image.

There are four components to our method. First, a trainable module estimates the pose of each part, at each frame

$t \in [1, T]$ (Section 3.2). From the posed parts, we propose to discover the underlying structure (Section 3.5), which we use as regularization during training, and to re-pose the object at inference. The third is a module that also uses the pose of the parts, and is trained to predict the color and signed-distance function of points in 3D (Section 3.3). The final component renders the output view by performing volumetric rendering on these predictions [35, 57] (Section 3.4). We train the system end-to-end (Figure 3).

3.2. Pose Estimation

As shown in Figure 2, we represent each part of the object $\mathcal{O}$ as an ellipsoid e_i, such that their union, $\mathcal{E}$, approximates the object's 3D shape Ω: $\mathcal{E} = \bigcup_{i=1}^{P} e_i \approx \Omega$. Each ellipsoid has a learnable three-dimensional radius parameter $\mathbf{r}_i$. Using the frame id t as input, we train an MLP, $\mathcal{T}_\Theta$, that outputs the global rotation $\mathbf{R}_i(t)$ (represented with a 3×3 rotation matrix) and translation $\mathbf{t}_i(t)$ of each e_i. We initialize these parameters randomly and observe that the optimization is reasonably robust to different initializations. Following common practice [35], rather than feeding t to $\mathcal{T}_\Theta$ directly, we use positional encoding $\gamma(t)$, where $\gamma(\cdot) = \{\cos(\alpha\ \cdot)\}_{\{\alpha=1:50\}}$. Since we overfit our system to a single scene, we can directly optimize the rotation and translation of each part in the global coordinate system, for each time frame: $\mathcal{T}_\Theta : \gamma(t) \rightarrow \{\mathbf{R}_i(t), \mathbf{t}_i(t)\}_{\{i=1:P\}}$. By predicting rotations and translations in the global coordinate system, we naturally force the pose of the object to be estimated consistently across views.

3.3. Shape and Appearance Decoder

Similar to NeuS [57], we seek to estimate the color, $\mathbf{c}$, and signed-distance functions (SDFs), d, at any 3D point, $\mathbf{x}^g$, to perform volumetric rendering. Since the ellipsoids alone cannot accurately capture the object's shape, we use a second MLP, $\mathcal{S}_\Theta$, to predict a residual. To ensure that the final shape does not deviate significantly from $\mathcal{E}$, we represent this as a residual SDFs, Δd, which is bounded by construction. We first convert the query point $\mathbf{x}^g$, expressed in global coordinates, to the local coordinate system of each part $\mathbf{x}_i(t) = (\mathbf{R}_i(t))^{-1}(\mathbf{x}^g - \mathbf{t}_i(t))$, and we apply weighted positional encoding [36] to compute a feature vector

$$\mathbf{f} = \textsc{cat}\{\mathrm{w}_i^{\mathrm{PE}}\ \gamma(\mathbf{x}_i(t))\}_{\{i=1:P\}}, \tag{1}$$

where CAT is the concatenation operation. The weights in Equation 1 are computed as

$$\mathbf{w}^{\mathrm{PE}} = \mathrm{softmax}\left\{-s^{\mathrm{PE}} d_i\right\}_{\{i=1:P\}}, \tag{2}$$

where $d_i = \mathrm{SDF}_i(\mathbf{x}_i(t), e_i)$, and s^{PE} is a learnable temperature parameter for the softmax. The SDFs from the ellipsoids can be computed directly from their radii and poses

Figure 4: Reconstruction and part segmentation rendered from novel perspectives.

(see Supplementary). Note that $\mathbf{f}$ effectively subsumes the current estimates of the ellipsoids, their pose, and the location of the sampled point. We then feed $\mathbf{f}$ to a second MLP

$$\mathcal{S}_\Theta : \mathbf{f}(\mathbf{x}^g) \rightarrow (\mathbf{c}, \widetilde{\Delta d})\big|_{\mathbf{x}^g}, \tag{3}$$

where $\mathbf{c}$ is color of $\mathbf{x}^g$, and $\widetilde{\Delta d}$ a residual with respect to the SDFs estimated from the ellipsoids, which we compress as

$$\Delta d = d_{\mathrm{max}} \tanh(s \widetilde{\Delta d}), \tag{4}$$

where d_{max} is the maximum value of Δd and s is a learnable scale parameter. The final SDF can be computed as

$$d = -\frac{1}{s^d} \mathrm{logsumexp}\left\{-s^d d_i\right\}_{\{i=1:P\}} + \Delta d, \tag{5}$$

where and s^d is a learnable scaling factor. Following NeuS [57], we compute the S-density from the signed-distance function, d, and use it with the color estimate of the 3D point to volume render the desired image. We also regularize the SDF with the Eikonal loss [13]:

$$\mathcal{L}_{\mathrm{SDF}} = \mathbb{E}[(|\nabla d|_2 - 1)^2]. \tag{6}$$

We predict SDFs rather than densities because Δd is bounded (Equation 4), and thus it naturally bounds the difference between the estimated surface positions of object Ω and the ellipsoid approximation, $\mathcal{E}$.

3.4. Rendering

The color of an output pixel can be predicted by volumetric rendering using the signed-distance function, as proposed in NeuS [57], and which we briefly describe here for completeness. The discrete opacity of the j-th point along the 3D ray corresponding to the output pixel can be computed as $\alpha_j = \max\left((\Phi_s(d_j) - \Phi_s(d_{j+1}))/\Phi_s(d_j), 0\right)$, where d_j is the signed-distance function at the point (represented in global coordinates) and Φ_s is a sigmoid function.

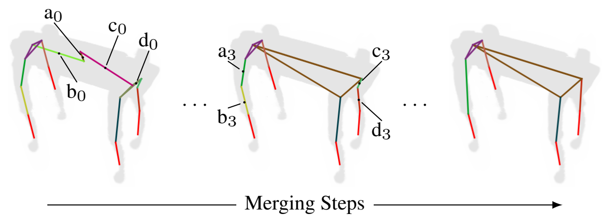


Figure 5: We do not assume the number of parts to be known. Rather, we intentionally over-segment the object for training. After convergence, we merge parts that are static relative to each other. In this case, the first three steps of our procedure merge a_0, b_0, c_0, and d_0 into a single part. The final two steps merge a_3 with b_3, and c_3 with d_3. A polygon (*e.g.*, the triangular body of the robot) indicates a part that is connected to more than two parts.

From this equation we compute the accumulated transmittance along the ray, $T_j = \prod_{k=1}^{j-1}(1-\alpha_k)$, which we use to estimate the color of the output pixel as $\hat{\mathbf{C}} = \sum_j T_j \alpha_j \mathbf{c}_j$. Similarly, the foreground mask can be rendered as $\hat{\mathrm{M}} = \Sigma_j T_j \alpha_j$. The photometric reconstruction loss $\mathcal{L}_{\text{photo}}$ is then

$$\mathcal{L}_{\text{photo}} = \mathbb{E}_{t,\text{ray}}[||\hat{\mathbf{C}} - \mathbf{C}_{GT}||_2^2 + ||\hat{\mathbf{M}} - \mathbf{M}_{GT}||_2^2]. \quad (7)$$

We refer to the paper by Wang *et al*. for more details [57].

3.5. Discovery of the 3D Joints

So far, we have described the object's parts as an *unstructured* set of ellipsoids $\mathcal{E}$. That is, each part's transformation is applied in the global coordinate frame, and the parts act independently of each other. However, because these ellipsoids represent the object explicitly, they allow us to discover the underlying structure. Specifically, we make two observations. First, a point inside part e_i that coincides with (is close to) a point in part e_j as the relative pose between the two parts changes, is likely to be a joint that connects the two parts. We detail how we leverage this insight in Section 3.5.1. Because we do not know the number of parts a priori, we start by over-segmenting the object. Our second insight is that two connected parts that maintain the same relative pose throughout the sequence can be merged: the joint between them is not necessary to explain the poses observed in the input sequence, Section 3.5.2. While we discover the final structure and finalize the part merging after convergence, we also compute their respective losses at training time for additional regularization (see Section 3.6).

3.5.1 Structure Discovery

We start by sampling N equally spaced joint candidates, $\boldsymbol{\xi}_i^n$, for each part e_i, see Figure 2. We provide more details about the sampling in the Supplementary. In order to discover connections we first compute the distance between all candidates over all frames and for every part pair (i, j)

$$l_{i,j}^{m,n} = \sum_t \left(||\boldsymbol{\xi}_i^m - \boldsymbol{\xi}_j^n||_2^2 + \lambda_l ||\mathbf{t}_i - \mathbf{t}_j||_2^2\right), \quad (8)$$

where the second term penalizes connections between parts that are far from each other, λ_l is a regularization coefficient, and t is the frame id. To prevent the distance from changing too quickly, we smooth it across training iterations

$$\bar{l}_{i,j}^{m,n}(\tau+1) \leftarrow (1-\epsilon)\cdot \bar{l}_{i,j}^{m,n}(\tau) + \epsilon \cdot l_{i,j}^{m,n}(\tau), \quad (9)$$

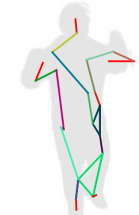
where ϵ is a momentum, and τ the training iteration. We compute the cost of connecting parts i and j as

$$\bar{l}_{i,j}(\tau) = \min_{n,m} \bar{l}_{i,j}^{m,n}(\tau). \quad (10)$$

We sort the list of $\bar{l}_{i,j}$'s for all parts in ascending order and traverse it to connect the parts that are closest (lowest cost). We assume the object's structure, Γ, to be an acyclic graph, so we require that there be a path between any two joints, and we do not allow connections that would create loops. We do not connect parts that violate this requirement, even if their $\bar{l}$ is the next lowest. This procedure allows us to determine the structure of any articulated object that can be modelled as an acyclic graph. We note that modeling the structure as a tree naturally yields a hierarchy for the parts (with an arbitrary definition of the root node), which is necessary for re-posing. We also compute the overall cost associated with a particular configuration Γ

$$\mathcal{L}_\Gamma = \sum_{(i,j)\in\Gamma} \bar{l}_{i,j}, \quad (11)$$

which we use to regularize our training procedure (see Section 3.6). Figure 5 shows the typical quality of the structure we identify. Note that a part can connect to multiple parts. While this approach is reasonably robust, we found that roughly one out of ten random initializations (Section 3.2) can yield to slightly incorrect discovered structures, as shown in the inset on the right.



3.5.2 Part Merging

Rather than assuming prior knowledge on the total number of parts, we over-segment the object and merge redundant parts. Specifically, we combine parts that are static with respect to each other throughout the sequence, see Figure 5. Differently put, we only preserve the articulations that are necessary to explain a change of pose in the input videos. The relative position between parts can be computed as $\mathbf{R}_i^j = \mathbf{R}_i^{-1}\mathbf{R}_j$ and $\mathbf{t}_i^j = \mathbf{R}_i^{-1}(\mathbf{t}_j - \mathbf{t}_i)$. We can then measure the relative motion as

$$D_{i,j} = \sigma_t(\mathbf{R}_i^j) + \lambda_{\text{motion}}\sigma_t(\mathbf{t}_i^j), \quad (12)$$

Figure 6: After learning the 3D joints and parts of a previously unseen object category from a multi-view video sequence, we can re-pose it by explicitly manipulating rotation and translation of each joint and part. Credits: human [41], dog [21].

where σ_t is the standard deviation over time. We use Equation 12 to define an additional loss term for our training:

$$\mathcal{L}_{\text{merge}} = \frac{1}{P^2} \sum_{i \neq j} D_{i,j} \Phi_1 \left(\frac{\bar{D} - D_{i,j}}{\bar{D}} \right), \tag{13}$$

where $\bar{D}$ is a hyperparameter, and Φ_1 is a sigmoid function. Additional training details can be found in the Supplementary. After the training is complete, we merge parts with limited motion relative to each other. Specifically, we compute Equation 12 for all pairs of parts and iteratively merge those for which $D_{i,j}$ is small. A few steps of this process are shown in Figure 5.

3.6. Training Strategy and Regularization

We train our system on a single scene, and in an end-to-end fashion. The process optimizes also for the parts' radii $\{\mathbf{r}_i\}$, in addition to training the parameters of the two MLPs. To help stabilize the training, we progressively increase the number of frames used for training as the training converges. This strategy yields a reasonable initialization of the structure, which is then adjusted to capture a consistent part decomposition and structure over the entire video.

Our loss function comprises several terms, including $\mathcal{L}_{\text{SDF}}$, $\mathcal{L}_{\text{photo}}$, $\mathcal{L}_{\Gamma}$, and $\mathcal{L}_{\text{merge}}$ in Equations 6, 7, 11, and 13, respectively. However, we only add $\mathcal{L}_{\text{merge}}$ after all the frames are added to the training. We describe additional regularization terms in the following, and we evaluate the contribution of each term in Section 4.5. The final loss is a weighted sum of these terms, see Supplementary.

Ellipsoid Surface Regularization Our explicit use of ellipsoids to approximate the shape of the object allows us to sample points from the surface at a low cost. The projection of sampled surface points onto the image should cover the

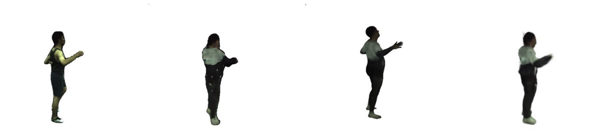


Figure 7: Re-posing comparison against the baseline methods, Section 4.3. For Kundu* *et al.* we use the neutral body model.

whole foreground mask of the object, and no pixels outside of it. We encourage this by minimizing the chamfer distance between the points sampled from the surface and the points sampled from the foreground mask:

$$\mathcal{L}_{\mathcal{E}} = \frac{1}{N_{\mathcal{E}}} \sum_i \min_j ||\mathbf{p}_i^{\mathcal{E}} - \mathbf{p}_j^M||_2^2 + \frac{1}{N_M} \sum_j \min_i ||\mathbf{p}_i^{\mathcal{E}} - \mathbf{p}_j^M||_2^2,$$

where $\mathbf{p}_i^{\mathcal{E}}$ are the coordinates of points randomly sampled from the surface of $\mathcal{E}$ and projected into the image space, $N_{\mathcal{E}}$ is their number, $\mathbf{p}_j^M$ are the coordinates of points randomly sampled from the mask M_{GT}, and N_M is their number.

Part Coverage Loss Similarly, the centers of the parts, $\mathbf{t}_i$, should be distributed over the foreground mask, rather than being concentrated in a region. Therefore, the centers of the parts are also learned to minimize the chamfer distance to the foreground mask:

$$\mathcal{L}_{\mathbf{t}} = \frac{1}{P} \sum_i \min_j ||\mathbf{t}_i - \mathbf{p}_j^M||_2^2 + \frac{1}{N_M} \sum_j \min_i ||\mathbf{t}_i - \mathbf{p}_j^M||_2^2. \tag{14}$$

Separation Loss We further discourage the parts themselves to be concentrated in a single region by penalizing small distances between their centers:

$$\mathcal{L}_{\text{separation}} = \frac{1}{P^2} \sum_{i \neq j} \exp \left(\frac{|\mathbf{t}_i - \mathbf{t}_j|_2^2}{2\sigma^2} \right), \tag{15}$$

where σ controls the scale of the distances to be regularized.

4. Evaluation and Results

We evaluate our method's ability to re-pose objects and estimate their joints—both qualitatively and quantitatively.

4.1. Pose Manipulation

One critical advantage of our explicit representation is the ability to manipulate the pose of previously unseen categories without prior knowledge: we can directly use the structure we discover by watching the object move. Given the frame id of a particular sequence, we manipulate the corresponding pose by applying hand-crafted rotations and

Table 1: Quantitative evaluation.

	Novel view		Re-posing		Joints (mm)
	LPIPS↓	SSIM↑	LPIPS↓	SSIM↑	MPJPE↓
Kundu* *et al.* [23]	0.059	0.957	0.082	0.941	12.24
Schmidtke* *et al.* [48]	0.055	0.957	**0.061**	**0.953**	18.56
Ours w/o merge	**0.053**	**0.966**	0.064	**0.953**	**7.59**
Ours w/ merge	0.056	0.965	0.065	0.952	11.11

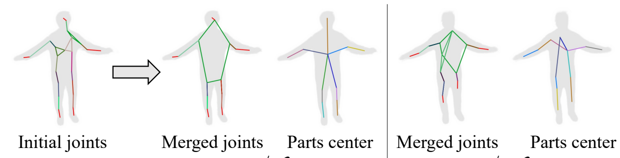
translations to each of the parts. To showcase our method's ability to discover the structure, we render a small dataset of seven structurally diverse robots. We provide more details about the rendered data in the Supplementary. We also trained our method on RGBD-dog, dataset of dogs [21]. Figures 1 and 6 show re-posing examples for objects with different structures, number of parts, and joints. More results are on the project's website. We do not estimate the range of motion at each joint, and leave it up to the user to define plausible poses.

4.2. Pose Manipulation for Humans

The structure our method discovers is plausible, as it allows for accurate re-posing, but it may not coincide exactly with the physical structure of the object. For a quantitative evaluation we focus on humans, because of the rich literature of methods and annotated data for this category. We use the ZJU-MoCap dataset [41] for our experiments. Specifically, after training our method, we use a subset of the training frames to learn a linear transformation from the joints of an SMPL model [32] to ours. Since the ZJU-MoCap dataset [41] has ground truth SMPL annotations, we can use this mapping to re-pose our model to target frames not observed in training, as shown in Figure 7. We use five subjects from the ZJU-MoCap dataset. For each sequence we use the first 80% of frames for training and the remaining for testing. The details of the mapping to and from the SMPL model are in the supplementary. Note that this mapping is for evaluation purposes only—our method allows for direct manipulation and does not need a SMPL model, as shown in Figure 6. The last two rows of Table 1 report the reconstruction quality of our re-posed renderings for the test frames, averaged over all the five subjects. We provide numbers for our model before and after merging (Section 3.5.2). We note that merging causes a small performance hit because it reduces the expressiveness (DOF) of the representation. However, even after merging, the performance remains competitive. Figure 8 shows the merging results with and without $\mathcal{L}_{\text{merge}}$. Using $\mathcal{L}_{\text{merge}}$ allows us to appropriately pull parts together that have the same motion, and to learn a more meaningful part decomposition.

4.3. Baselines Description and Evaluation

A direct numerical comparisons with the state-of-the-art is impossible: ours is the first work that can explicitly re-pose a dynamic object from a previously unseen category, without supervision (other than multi-view supervision), or prior knowledge of the underlying structure. Moreover existing methods are not scene-specific. The methods by Schmidtke *et al.* [48] and by Kundu *et al.* [23], both of which assume a template and only work for humans, are the closest existing solutions for unsupervised, direct pose manipulation. Although they tackle a more constrained task, we use them as inspiration for baselines that allow for a quantitative evaluation.



(a) w/ $\mathcal{L}_{\text{merge}}$ (b) w/o $\mathcal{L}_{\text{merge}}$

Figure 8: Effect of merging loss. (a) merging with $\mathcal{L}_{\text{merge}}$ and (b) without it.

Both Schmidtke *et al.* [48] and Kundu *et al.* [23] employ a CNN-based encoder, which allows them to work on scenes not seen in training. This gives an unfair advantage to our method, which overfits to a specific sequence. Therefore, we propose the modifications, which allow us to train both methods for a specific sequence, like ours.

We modify the method by Kundu *et al.* by swapping their CNN-based encoder with an MLP that overfits the SMPL parameters to each frame. These parameters are then used to adapt the SMPL mesh to the pose in the frame. We train the MLP by enforcing that the color of corresponding vertexes in different frames match. After convergence we compute the color of all the vertexes of the SMPL model by averaging the colors of the corresponding pixels in all the input frames. Re-posing their solution, then, reduces to manipulating the SMPL parameters. To adapt the approach of Schmidtke *et al.*, we replace their 2D template with a 3D template and their CNN encoder with an MLP that learns how to deform the 3D template to match the pose at the given time frame. Given a viewport we can project the template to a 2D representation, which can be converted to an RGB image with a second network. We denote both baselines with a * to indicate they are adapted from their original versions, and provide a diagram for each method in the Supplementary. A few considerations are in order. First, the architecture of both MLPs is the same as ours, and the number of the parameters to be predicted comparable. Second, while we make those methods scene-specific to remove our advantage, they still only work for people and still use a template or an SMPL model, like the original versions. They are our best effort at a fair comparison. We train both models on the same train/test split of the same five subjects we use for our method. For Kundu* *et al.* we use the neutral SMPL body model. A qualitative comparison can be seen in Figure 7. Table 1 reports LPIPS [66] and SSIM [58] for both reconstruction (*i.e.*, same pose as in one of the input

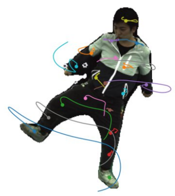


Figure 9: The joints our method discovers are plausible and stable across the sequence.

frames, but different view) and re-posing. Our method is on par or slightly better than these baselines despite making no assumptions about the structure of the object.

4.4. Joint Estimation Evaluation

Our method discovers plausible joints. That is, they allow to re-pose the object consistently with the input images, but they may not exactly coincide with the physical joints. Figure 9 offers a qualitative evaluation: our 3D joints appear to closely follow the physical joints locations and they are stable over time. For a quantitative evaluation, we use one tenth of the frames in each sequence to compute a linear mapping from our joints and joints candidates to the joints of the SMPL model provided by the dataset, as is common practice for methods that discover landmarks [33, 52, 67]. The details of the algorithm that regresses this mapping are in the Supplementary. We apply the linear mapping to the remaining frames to compute the mean per joint position error (MPJPE) [15]. We also compare with the baselines defined in Section 4.3. For Kundu* *et al.* we learn a linear mapping from the predicted SMPL vertices to the GT joints provided by the dataset, while for Schmidtke* *et al.* we compute the same linear mapping as for our method. Once again, our method performs on par, and sometimes even better despite the additional information available to the baseline methods.

4.5. Ablation Study

In our first ablation study we evaluate the effect of each loss term to the overall performance. We use subject 366 from the ZJU-MoCap dataset and train our model from scratch by disabling one loss term at the time. The results are shown in Table 2. We note that the additional terms have a marginal effect on the quality of the rendered images, but they do reduce the joints estimation error measurably. Although L_{t} slightly degrades the novel view synthesis performance, without it, we observe issues with the structure discovery. Qualitative results are shown in the Supplementary. In our second experiment, we evaluate the importance of training $\mathcal{S}_{\Theta}$ to predict an SDF residual Δd, instead of the SDF d itself, as done in Neural-GIF [53]. Table 2 confirms that predicting residual is critical to both the image reconstruction quality and the joints estimation.

Table 2: Ablation study.

	Ours	$+\mathcal{L}_{\text{merge}}$	$-\mathcal{L}_{\Gamma}$	$-\mathcal{L}_{\mathcal{E}}$	$-\mathcal{L}_{\mathrm{t}}$	$-\mathcal{L}_{\text{separation}}$	$-\Delta d$
Novel view LPIPS↓	0.063	0.062	0.062	0.062	**0.061**	0.062	0.065
Novel view SSIM↑	0.958	0.958	0.958	**0.959**	**0.959**	**0.959**	0.954
Novel pose LPIPS↓	**0.065**	**0.065**	0.069	0.069	**0.065**	0.067	0.077
Novel pose SSIM↑	**0.954**	0.953	0.951	0.952	**0.954**	0.952	0.946
Joint MPJPE(mm)↓	**8.49**	10.25	9.35	12.13	8.70	9.72	22.14

5. Discussion and Limitations

Our method can discover the structure of unseen categories, but it needs a foreground mask, which may not be available for new classes. While off-the-shelf instance segmentation approaches can be used, as we do for the experiments on humans [4] and dogs [5], this limits the practical applicability of our method. In addition, our method requires multi-view videos that capture the object from all sides. The results in the paper use five cameras around the object for robots, six cameras for humans, and eight cameras for the dog. The four viewpoints available in the Human3.6M dataset [15] do not provide sufficient coverage for our method. An additional constraint is that we can only repose parts that move relative to each other in the training sequence—we cannot infer what we cannot see. Our solution does not tackle the problem of defining plausible motion ranges around the joints and focuses on spherical joints, leaving different types of joints, such as sliding joints, for future work. We also observed that the randomness of the structure initialization can sometimes affect the structure discovery (Section 3.5.1). We leave it to future work to find a more elegant solution than simply re-initializing it when this happens.

Societal impact. Our approach lends itself to similar dishonest uses as deepfakes, *e.g.*, a person could be "rendered to perform" illegal, incriminating, or indecent activities.

6. Conclusions

We presented a method that discovers the structure of an articulated object from arbitrary categories, by watching it move in a multi-view video. It can then render the object from novel views and even directly manipulate its pose. Our method works for arbitrary articulated objects, as we show using robots with varying structures.

Acknowledgments

This work was partially supported by JST AIP Acceleration Research JPMJCR20U3, Moonshot R&D Grant Number JPMJPS2011, JSPS KAKENHI Grant Number JP19H01115 and Basic Research Grant (Super AI) of Institute for AI and Beyond of the University of Tokyo.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srini-

vasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[2] Aljaz Bozic, Pablo Palafox, Michael Zollhofer, Justus Thies, Angela Dai, and Matthias Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[3] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[4] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 8

[5] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 8

[6] Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[7] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Unsupervised part discovery from contrastive reconstruction. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[8] Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. NASA neural articulated shape approximation. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[9] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational U-Net for conditional appearance and shape generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[10] Joao Fayad, Chris Russell, and Lourdes Agapito. Automated articulated structure and 3D shape recovery from point correspondences. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. 3

[11] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3D shape. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[12] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[13] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning (ICML)*, 2020. 4

[14] Wei-Chih Hung, Varun Jampani, Sifei Liu, Pavlo Molchanov, Ming-Hsuan Yang, and Jan Kautz. SCOPS: Self-supervised co-part segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2014. 1, 8

[16] Umar Iqbal, Pavlo Molchanov, Thomas Breuel, Juergen Gall, and Jan Kautz. Hand pose estimation via 2.5D latent heatmap regression. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 3

[17] Umar Iqbal, Pavlo Molchanov, and Jan Kautz. Weakly-supervised 3D human pose learning via multi-view images in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3

[18] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 3

[19] Tomas Jakab, Ankush Gupta, Hakan Bilen, and Andrea Vedaldi. Self-supervised learning of interpretable keypoints from unlabelled videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[20] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3

[21] Sinead Kearney, Wenbin Li, Martin Parsons, Kwang In Kim, and Darren Cosker. RGBD-Dog: Predicting canine pose from RGBD sensors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6, 7

[22] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3D human pose using multi-view geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[23] Jogendra Nath Kundu, Mugalodi Rakesh, Varun Jampani, Rahul Mysore Venkatesh, and R Venkatesh Babu. Appearance consensus driven self-supervised human mesh recovery. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 3, 7

[24] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural human performer: Learning generalizable radiance fields for human performance rendering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[25] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Single-view robot pose and joint angle estimation via render and compare. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3

[26] Timothy E Lee, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Oliver Kroemer, Dieter Fox, and Stan Birchfield. Camera-to-robot pose estimation from a single image. In *International Conference on Robotics and Automation (ICRA)*, 2020. 1, 3

[27] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. HybrIK: A hybrid analytical-neural inverse kinematics solution for 3D human pose and shape estima-

tion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3

[28] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[29] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. BARF: Bundle-adjusting neural radiance fields. *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[30] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 2021. 2

[31] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping GAN: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 2

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (ToG)*, 2015. 7

[33] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Bjorn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 8

[34] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2

[35] 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 *European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 4

[36] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 4

[37] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3DPO: Canonical 3D pose networks for non-rigid structure from motion. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[38] Michael Oechsle, Songyou Peng, and Andreas Geiger. UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[39] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. HyperNeRF: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv:2106.13228*, 2021. 1, 2

[40] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[41] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 6, 7

[42] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, , and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[43] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[44] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning monocular 3D human pose estimation from multi-view images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3

[45] Darius Rückert, Linus Franke, and Marc Stamminger. ADOP: Approximate differentiable one-pixel point rendering. *arXiv:2110.06635*, 2021. 1

[46] Sara Sabour, Andrea Tagliasacchi, Soroosh Yazdani, Geoffrey Hinton, and David J Fleet. Unsupervised part representation by flow capsules. In *International Conference on Machine Learning (ICML)*, 2021. 3

[47] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Learning to regress 3D face shape and expression from an image without 3D supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[48] Luca Schmidtke, Athanasios Vlontzos, Simon Ellershaw, Anna Lukens, Tomoki Arichi, and Bernhard Kainz. Unsupervised human pose estimation through transforming shape templates. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3, 7

[49] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-NeRF: Articulated neural radiance fields for learning human shape, appearance, and pose. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[50] Xiao Sun, Bin Xiao, Shuang Liang, and Yichen Wei. Integral human pose regression. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 3

[51] Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3D keypoints via end-to-end geometric reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[52] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 3, 8

[53] Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Neural-GIF: Neural generalized implicit functions for animating people in clothing. In *IEEE Conference*

on Computer Vision and Pattern Recognition (CVPR), 2021. 2, 8

[54] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a deforming scene from monocular video. *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[55] Bastian Wandt, Marco Rudolph, Petrissa Zell, Helge Rhodin, and Bodo Rosenhahn. CanonPose: Self-supervised monocular 3D human pose estimation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3

[56] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv:2105.05994*, 2021. 1, 2

[57] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2, 4, 5

[58] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing (TIP)*, 2004. 7

[59] Jingyu Yan and Marc Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2008. 3

[60] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Ce Liu, and Deva Ramanan. ViSER: Video-specific surface embeddings for articulated 3D shape reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 3

[61] Gengshan Yang, Minh Vo, Neverova Natalia, Deva Ramanan, Vedaldi Andrea, and Joo Hanbyul. BANMo: Building animatable 3D neural models from many casual videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[62] Lingbo Yang, Pan Wang, Chang Liu, Zhanning Gao, Peiran Ren, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Xiansheng Hua, and Wen Gao. Towards fine-grained human pose transfer with detail replenishing network. In *IEEE Transactions on Image Processing (TIP)*, 2020. 2

[63] Yuan Yao, Yasamin Jafarian, and Hyun Soo Park. MONET: Multiview semi-supervised keypoint detection via epipolar divergence. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 3

[64] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. Pose-guided human animation from a single image in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[65] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[66] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 7

[67] Yuting Zhang, Yijie Guo, Yixin Jin, Yijun Luo, Zhiyuan He, and Honglak Lee. Unsupervised discovery of object landmarks as structural representations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3, 8

[68] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo Magnification: Learning view synthesis using multiplane images. In *ACM Transactions on Graphics (SIGGRAPH)*, 2018. 1

[69] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[70] Christian Zimmermann and Thomas Brox. Learning to estimate 3D hand pose from single RGB images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 3