# Unsupervised Learning of Efficient Geometry-Aware Neural Articulated Representations

**Atsuhiro Noguchi**[1] **Xiao Sun**[2] **Stephen Lin**[2] **Tatsuya Harada**[1,3]

[1]The University of Tokyo [2]Microsoft Research Asia [3]RIKEN

## Abstract

We propose an unsupervised method for 3D geometry-aware representation learning of articulated objects. Though photorealistic images of articulated objects can be rendered with explicit pose control through existing 3D neural representations, these methods require ground truth 3D pose and foreground masks for training, which are expensive to obtain. We obviate this need by learning the representations with GAN training. From random poses and latent vectors, the generator is trained to produce realistic images of articulated objects by adversarial training. To avoid a large computational cost for GAN training, we propose an efficient neural representation for articulated objects based on tri-planes and then present a GAN-based framework for its unsupervised training. Experiments demonstrate the efficiency of our method and show that GAN-based training enables learning of controllable 3D representations without supervision.

***Keywords*** image synthesis, articulated objects, neural radiance fields, unsupervised learning

## 1 Introduction

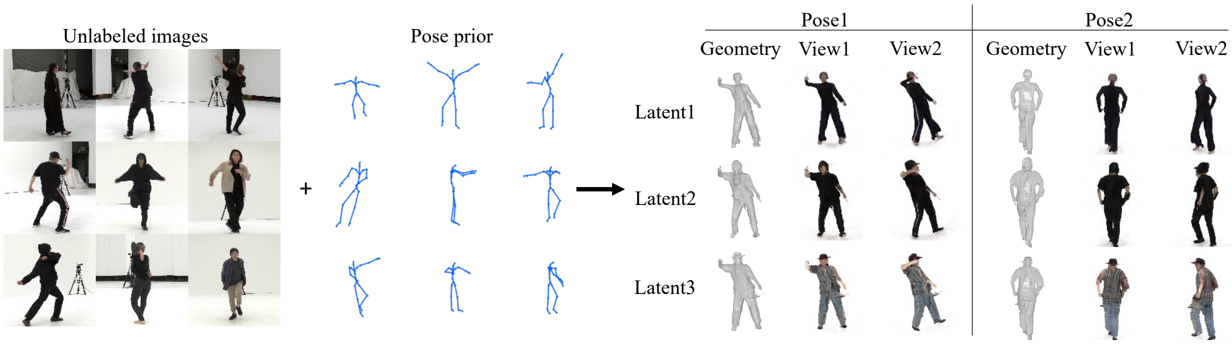

Figure 1: Our ENARF-GAN is a geometry-aware, 3D-consistent image generation model that allows independent control of viewpoint, object pose, and appearance information. It is learned from unlabeled images and a prior distribution on object pose.

3D models that allow free control over the pose and appearance of articulated objects are of importance in various applications including computer games, media content creation, and augmented/virtual reality. In early work, articulated objects were typically represented by explicit models such as skinned meshes. More recently, the success of learned implicit representations such as neural radiance fields (NeRF) Mildenhall et al. [2020] for rendering static 3D scenes has led to extensions for modeling dynamic scenes and articulated objects. Much of this attention has focused on photorealistic rendering of humans, from novel viewpoints and with controllable pose, by learning from images and videos.

Existing methods for learning explicitly pose-controllable articulated representations, however, require much supervision such as from multi-view synchronized videos with 3D pose/mesh annotation and a mask for each frame. Preparing such data involves tremendous annotation costs, thus reducing annotation is of great importance. In this paper, we propose a novel unsupervised learning framework for 3D pose-aware generative models of articulated objects, which are learned from unlabeled images of objects sharing the same structure and a distribution of object poses.

We exploit recent advances in 3D-aware generative adversarial networks (GANs) Schwarz et al. [2020], Chan et al. [2021], Niemeyer and Geiger [2021], Gu et al. [2022], Chan et al. [2022] for unsupervised learning of the articulated representations. They learn 3D-aware image generation models from images without supervision such as viewpoints or 3D shape. The generator is based on NeRF Mildenhall et al. [2020] and is optimized with a GAN objective by generating images from randomly sampled viewpoints and latent vectors from a prior distribution defined before training. By minimizing the distribution distance between generated and training images via GAN training, the generator learns to generate 3D-consistent images without any supervision. We employ the idea to learn representations for articulated objects without supervision. By defining a pose prior distribution for the target object, and optimizing the GAN objective function on randomly generated images from random poses and latent variables, it becomes possible to learn a generative model with free control of poses. We demonstrate this approach by modeling the pose prior as a skeletal distribution Noguchi et al. [2021], Su et al. [2021], while noting that other models like meshes Peng et al. [2021a,b] may bring potential performance benefits.

However, direct application of existing neural articulated representations to GANs is not computationally practical. While NeRF can produce high quality images, its processing is expensive because it requires network inference for every point in space. In recent years, many methods have been proposed to overcome this problem. Some methods Niemeyer and Geiger [2021], Gu et al. [2022] reduce computational cost by volume rendering at low resolution followed by 2D CNN based upsampling. Although this technique achieves high-resolution images with real-time inference speed, it is not geometry-aware (i.e., the surface mesh cannot be extracted). Recently, a method which we call Efficient NeRF Chan et al. [2022] overcomes the problem. The method is based on an efficient tri-plane based neural representation and GAN training on it. Thanks to the computationally efficient geometry-aware representation, it can produce relatively high-resolution ($128 \times 128$) images with volumetric rendering. To learn a 3D geometry-aware articulated representation without supervision, we extend the tri-plane representation to articulated objects. An overview of the method is visualized in Figure 1.

The contributions of this work are as follows:

- We propose a novel efficient neural representation for articulated objects based on an efficient tri-plane representation.
- We propose an efficient implementation of deformation fields using tri-planes for dynamic scene training, achieving 4 times faster rendering than NARF Noguchi et al. [2021] with comparable or better performance.
- We propose a novel GAN framework to learn articulated representations without using any 3D pose or mask annotation for each image. The controllable 3D representation can be learned from real unlabeled images.

## 2 Related Work

**Articulated 3D Representations.** The traditional approach for modeling pose-controllable representations of articulated objects is by skinned mesh Jacobson et al. [2014], James and Twigg [2005], Lewis et al. [2000], where each vertex of the mesh is deformed according to the skeletal pose. Several parametric skinned mesh models have been developed specifically for humans and animals Loper et al. [2015], Hesse et al. [2019], Pavlakos et al. [2019], Osman et al. [2020], Zuffi et al. [2017]. For humans, the skinned multi-person linear model (SMPL) Loper et al. [2015] is commonly used. However, these representations can only handle tight surfaces with no clothing and cannot handle non-rigid or topology-changing objects such as clothing or hair. Recently, implicit 3D shape representations have achieved state-of-the-art performance in pose-conditional shape reconstruction. These methods learn neural occupancy/indicator functions Chen and Zhang [2019], Mescheder et al. [2019] or signed distance functions Park et al. [2019] of articulated objects. These methods include parts-based Deng et al. [2020], Bozic et al. [2021] and skinning-based methods Tiwari et al. [2021], Chen et al. [2021]. However, all of these models require ground truth 3D pose and/or shape of objects for training. Very recently, methods have been proposed to reproduce the 3D shape and motion of objects from video data without using 3D shape and pose annotation Yang et al. [2021, 2022], Noguchi et al. [2022]. However, they either do not allow free control of poses or are limited to optimizing for a single object. An SMPL mesh-based generative model Grigorev et al. [2021] can learn a pose controllable generative model for humans. However, the rendering process is completely in 2D, and thus it is not geometry-aware.

**Implicit 3D representations.** Implicit 3D representations are memory efficient, continuous, and topology free. They have achieved the state-of-the art in learning 3D shape Chen and Zhang [2019], Mescheder et al. [2019], Park et al. [2019], static Sitzmann et al. [2019], Mildenhall et al. [2020], Barron et al. [2021] and dynamic scenes Pumarola et al. [2021], Li et al. [2021a], Park et al. [2021], articulated objects Peng et al. [2021a], Deng et al. [2020], Bozic et al. [2021], Tiwari et al. [2021], Chen et al. [2021], Peng et al. [2021b], Noguchi et al. [2021], Su et al. [2021], Alldieck et al. [2021], Xu et al. [2021], and image synthesis Schwarz et al. [2020], Chan et al. [2021]. Although early works rely on ground truth 3D geometry for training Chen and Zhang [2019], Mescheder et al. [2019], Park et al. [2019], developments in differentiable rendering have enabled learning of networks from only photometric reconstruction losses Sitzmann et al. [2019], Yariv et al. [2020], Mildenhall et al. [2020]. In particular, neural radiance fields (NeRF) Mildenhall et al. [2020] applied volumetric rendering on implicit color and density fields, achieving photorealistic novel view synthesis of complex static scenes using multi-view posed images. Dynamic NeRF Pumarola et al. [2021], Li et al. [2021a], Park et al. [2021] extends NeRF to dynamic scenes, but these methods just reenact the motion in the scene and cannot repose objects based on their structure. Recently, articulated representations based on NeRF have been proposed Peng et al. [2021a], Noguchi et al. [2021], Su et al. [2021], Peng et al. [2021b], Xu et al. [2021, 2022]. These methods can render images conditioned on pose configurations. However, all of them require ground truth skeletal poses or an SMPL mesh for training, which makes them unsuitable for in-the-wild images.

Another NeRF improvement is the reduction of computational complexity: NeRF requires forward computation of MLPs to compute color and density for every point in 3D. Thus, the cost of rendering is very high. Fast NeRF algorithms Garbin et al. [2021], Yu et al. [2021] reduce the computational complexity of neural networks by creating caches or using explicit representations. However, these methods can only be trained on a single static scene. Very recently, a hybrid explicit and implicit representation was proposed Chan et al. [2022]. In this representation, the feature field is constructed using a memory-efficient explicit representation called a tri-plane, and color and density are decoded using a lightweight MLP. This method can render images at low cost and is well suited for image generation models.

In this work, we propose an unsupervised learning framework for articulated objects. We extend the idea of tri-planes to articulated objects for efficient training.

**Generative 3D-aware image synthesis** Advances in generative adversarial networks (GANs) Goodfellow et al. [2014] have made it possible to generate high-resolution, photorealistic images Karras et al. [2019, 2020, 2021]. In recent years, many 3D-aware image generation models have been proposed by combining GANs with 3D generators that use meshes Szabó et al. [2019], voxels Wu et al. [2016], Zhu et al. [2018], Gadelha et al. [2017], Henzler et al. [2019], Nguyen-Phuoc et al. [2019, 2020], depth Noguchi and Harada [2020], or implicit representations Schwarz et al. [2020], Chan et al. [2021], Niemeyer and Geiger [2021], Gu et al. [2022], Chan et al. [2022]. These methods can learn 3D-aware generators without 3D supervision. Among these, image generation methods using implicit functions, thanks to their continuous and topology-free properties, have been successful in producing 3D-consistent and high quality images. However, fully implicit models Schwarz et al. [2020], Chan et al. [2021] are computationally expensive, making the training of GANs inefficient. Therefore, several innovations have been proposed to reduce the rendering cost of generators. Neural rendering-based methods Niemeyer and Geiger [2021], Gu et al. [2022] reduce computation by performing volumetric rendering at low-resolution and upsampling the rendered feature images using a 2D CNN. Though this enables generation of high-resolution images at a faster rate, 2D upsampling does not consider 3D consistency and cannot generate detailed 3D geometry. Very recently, a hybrid of explicit and implicit methods Chan et al. [2022] has been developed for 3d geometry-aware image generation. Instead of using a coordinate based implicit representation, this method uses tri-planes, which are explicit 3D feature representations, to reduce the number of forward computations of the network and to achieve volumetric rendering at high resolution.

The existing research is specialized to scenes of static objects or objects that exist independently of each other, and do not allow free control of the skeletal pose of the generated object. Therefore, we propose a novel GAN framework for articulated objects.

## 3 Method

Recent advances in implicit neural rendering Su et al. [2021], Noguchi et al. [2021] have made it possible to generate 3D-aware pose-controllable images of articulated objects from images with accurate 3D pose and foreground mask annotations. However, training such models from only in-the-wild images remains challenging since accurate 3D pose annotations are generally difficult to obtain for them. In the following, we first briefly review Neural Articulated Radiance Field (NARF) Noguchi et al. [2021], then propose an adversarial based framework, named ENARF-GAN, to efficiently train the NARF model without any paired pose-image and foreground mask annotations.

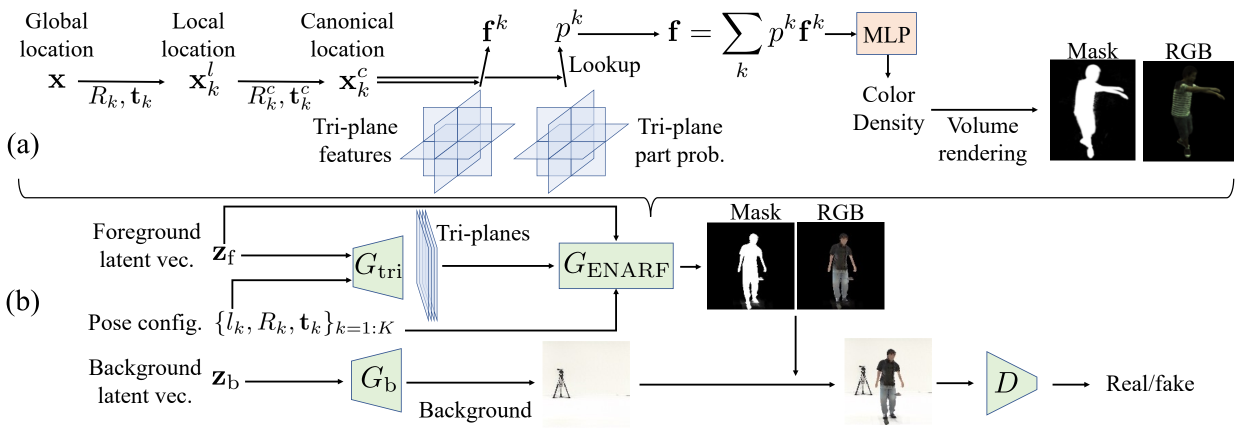

Figure 2: Overview of (a) Efficient NARF (ENARF) and (b) GAN training.

### 3.1 Neural Articulated Radiance Field Revisited

NARF is an implicit 3D representation for articulated objects. It takes a kinematic 3D pose configuration of an articulated object $o = \{l_k, R_k, \mathbf{t}_k\}_{k=1:K}$ as input and predicts the color and the density of any 3D location $\mathbf{x}$, where $l_k$ is the length of the $k^{\text{th}}$ part, and $R_k$ and $\mathbf{t}_k$ are its rotation and translation matrices, respectively. Given the pose configuration $o$, NARF first transforms a global 3D position $\mathbf{x}$ into several local coordinate systems defined by the rigid parts of the articulated object. Specifically, the transformed local location $\mathbf{x}_k^l$ for the $k^{\text{th}}$ part is computed as $\mathbf{x}_k^l = (R^k)^{-1}(\mathbf{x} - \mathbf{t}^k)$ for $k \in \{1, ..., K\}$.

NARF first trains an extra lightweight selector $S$ in the local space to decide which part a global 3D location $\mathbf{x}$ belongs to. Specifically, it outputs the probability $p^k$ of $\mathbf{x}$ belonging to the $k^{th}$ part. Then NARF computes color $c$ and density $\sigma$ at the location $\mathbf{x}$ from a concatenation of local locations masked by the corresponding part probability $p^k$.

$$c, \sigma = G(\mathrm{Cat}(\{\gamma(\mathbf{x}_k^l) * p^k\}_{k=1:K})), \tag{1}$$

where $\gamma$ is a positional encoding Mildenhall et al. [2020], Cat is the concatenation operation, and $G$ is implemented with a few MLP layers. The RGB color $\mathbf{C}$ and foreground mask value $\mathbf{M}$ for each pixel are generated by volumentric rendering Mildenhall et al. [2020]. The network is trained with a reconstruction loss between the generated and ground truth color $\mathbf{C}$ and mask $\mathbf{M}$. Please refer to the original NARF paper Noguchi et al. [2021] for more details.

### 3.2 Unsupervised Learning by Adversarial Training

In this work, we propose a method for efficient and unsupervised training of the NARF model from unposed image collections. Without loss of generality, we consider humans as the articulated objects here. To this end, we first define a human pose distribution $\mathcal{O}$. For one training iteration, our NARF based generator $G$ takes a latent vector $\mathbf{z}$ and a sampled human pose instance $o$ from $\mathcal{O}$ as input and predicts a synthesized image $\mathbf{C}$. Following standard adversarial training of GANs, a discriminator $\mathcal{D}$ is used to distinguish the synthesized image $\mathbf{C}$ from real ones $\tilde{\mathbf{C}}$. Formally, the training objectives of the generator $\mathcal{L}_{\text{adv}}^G$ and discriminator $\mathcal{L}_{\text{adv}}^D$ are defined as follows,

$$\mathcal{L}_{\text{adv}}^G = -\mathbb{E}\left[\log(D(G(\mathbf{z}, o)))\right], \mathcal{L}_{\text{adv}}^D = -\mathbb{E}\left[\log(D(\tilde{\mathbf{C}})) + \log(1 - D(G(\mathbf{z}, o)))\right]. \tag{2}$$

An overview of this method is illustrated in Figure 2 (b).

However, this training would be computationally expensive. The rendering cost of NARF is heavy because forward computation is performed for many 3D locations in the viewed space. Even though, in supervised training, the time and memory cost of computing the reconstruction loss could be reduced by evaluating it over just a small proportion of the pixels Noguchi et al. [2021], the adversarial loss in Equation 2 requires generation of the full image for evaluation. As a result, the amount of computation becomes impractical.

In the following, we propose a series of changes in feature computation and the selector to address this issue. Note that these changes not only enable implementation of the adversarial training, but also greatly improve the efficiency of the original NARF.

### 3.3 Efficiency Improvements on NARF

Recently, Chan et al. Chan et al. [2022] proposed a hybrid explicit-implicit 3D-aware network that uses a memory-efficient tri-plane representation to explicitly store features on axis-aligned planes. With this representation, the efficiency of feature extraction for a 3D location is greatly improved. Instead of forwarding all the sampled 3D points through the network, the intermediate features of arbitrary 3D points can be obtained via simple lookups on the tri-planes. The tri-plane representation can be more efficiently generated with Convolutional Neural Networks (CNNs) instead of MLPs. The intermediate features of 3D points are then transformed into the color and density using a lighweight decoder MLP. The decoder significantly increases the non-linearity between features at different positions, thus greatly enhancing the expressiveness of the model.

Here, we adapt the tri-plane representation to NARF for more efficient training. Similar to Chan et al. [2022], we first divide the original NARF network $G$ into an intermediate feature generator (the first linear layer of $G$) $W$ and a decoder network $G_{\text{dec}}$. Then, Equation 1 is rewritten as follows.

$$c, \sigma = G_{\text{dec}}(\mathbf{f}) \text{ , where } \mathbf{f} = W(\text{Cat}(\{\gamma(\mathbf{x}_k^l) * p^k | k \in \{1, ..., K\}\})), \tag{3}$$

where $\mathbf{f}$ is an intermediate feature vector of input 3D location $\mathbf{x}$, and $\mathbf{x}_k^l$ is the position of $\mathbf{x}$ in the local coordinate system of the $k^{\text{th}}$ part.

Re-implementing the feature generator $W$ to produce the tri-plane representation is not straightforward because its input, a weighted concatenation of $\mathbf{x}_k^l$, does not form a valid location in a specific 3D space. We make two important changes to address this issue. First, we decompose $W$ into $K$ sub-networks $\{W_k\}$, one for each part, where each sub-network takes the corresponding local position $\mathbf{x}_k^l$ as input and outputs an intermediate feature for the $k^{\text{th}}$ part. Then, the intermediate feature in Equation 3 can be equivalently rewritten as follows.

$$\mathbf{f} = \sum_{k=1}^{K} p^k * \mathbf{f}^k \text{ , where } \mathbf{f}^k = W_k(\gamma(\mathbf{x}_k^l)), \tag{4}$$

where $\mathbf{f}^k$ is a feature generated in the local coordinate system of the $k^{\text{th}}$ part. Now, $W_k$ can be directly re-implemented by tri-planes.

However, the computational complexity of this implementation is still proportional to $K$. In order to train a single tri-plane for all parts, the second change is to further transform the local coordinates $\mathbf{x}_k^l$ into a canonical space defined by a canonical pose $o^c$, similar to Animatable NeRF Peng et al. [2021b].

$$\mathbf{x}_k^c = R_k^c \mathbf{x}_k^l + \mathbf{t}_k^c, \tag{5}$$

where $R_k$ and $\mathbf{t}_k$ are respectively the rotation and translation of the $k^{\text{th}}$ part of the input pose, and $R_k^c$ and $\mathbf{t}_k^c$ are those of the canonical pose. Intuitively, $\mathbf{x}_k^c$ is the corresponding point location of $\mathbf{x}$ transformed into the canonical space when $\mathbf{x}$ is considered to belong to the $k^{\text{th}}$ part. Finally, the tri-plane features $\{F_{xy}, F_{xz}, F_{yz}\}$ are learned in the canonical space. The feature extraction for location $\mathbf{x}$ is achieved by retrieving the 32-dimensional feature vector $\mathbf{f}^k$ on the tri-plane in the canonical space for all parts, then taking a weighted sum of those features as in Equation 4,

$$\mathbf{f} = \sum_{k=1}^{K} p^k * \mathbf{f}^k \text{ , where } \mathbf{f}^k = F_{xy}(\mathbf{x}_k^c) + F_{xz}(\mathbf{x}_k^c) + F_{yz}(\mathbf{x}_k^c). \tag{6}$$

$F_{**}(\mathbf{a})$ is a retrieved feature vector from each axis-aligned plane at location $\mathbf{a}$.

We estimate the RGB color $c$ and density $\sigma$ from $\mathbf{f}$ using a lightweight decoder network $G_{\text{dec}}$ consisting of two FC layers with a hidden dimension of 64 and output dimension of 4. We apply volume rendering Mildenhall et al. [2020] on the color and density to output an RGB image $\mathbf{C}$ and a foreground mask $\mathbf{M}$.

Although we efficiently parameterize the intermediate features, the probability $p^k$ needs to be computed for every 3D point and every part. In the original NARF, the selector $S$ estimates the probabilities with lightweight MLPs. However, the number of forward passes of the MLPs is proportional to the number of parts $K$ and the cube of the resolution, which is computationally infeasible. Therefore, we propose an efficient selector network using tri-planes. Since $p^k$ is used to mask out features of irrelevant parts, this probability can be a rough approximation of the shape of each part. Thus the tri-plane representation is expressive enough to model the probability. We use $K$ separate 1-channel tri-planes to represent $P^k$, the part probability projected to each axis-aligned plane. We retrieve the three probability values $(p_{xy}^k, p_{xz}^k, p_{yz}^k) = (P_{xy}^k(\mathbf{x}_k^c), P_{xz}^k(\mathbf{x}_k^c), P_{yz}^k(\mathbf{x}_k^c))$ of the $k^{\text{th}}$ part by querying the 3D location in the canonical space $\mathbf{x}_k^c$. The probability $p^k$ that $\mathbf{x}_k^c$ belongs to the $k^{\text{th}}$ part is approximated as $p^k = p_{xy}^k p_{xz}^k p_{yz}^k$.

In this way, the features $F$ and part probabilities $P$ are modeled efficiently with a single tri-plane representation. The tri-plane is represented by a $(32 + K) \times 3$ channel image. The first 96 channels represent the tri-plane features in the canonical space. The remaining $3K$ channels represent the three axis-aligned probability maps for each of the $K$ parts. We call this approach Efficient NARF, or ENARF.

### 3.4 Dynamic Scene Overfitting

Before incorporating it into GAN training, we describe dynamic scene overfitting with the proposed tri-plane based representation to show its effectiveness. For training, we use the ground truth 3D pose and foreground mask of each frame. We optimize the reconstruction loss with respect to RGB and foreground mask.

$$\mathcal{L}_{\text{DSO}} = \sum_{r \in \mathcal{R}} \left( ||\mathbf{C} - \hat{\mathbf{C}}||_2^2 + ||\mathbf{M} - \hat{\mathbf{M}}||_2^2 \right), \tag{7}$$

where $\mathcal{R}$ is the set of rays in each batch, and $\hat{\mathbf{C}}$ and $\hat{\mathbf{M}}$ are the ground truth RGB color and foreground mask, respectively.

If the object shape is strictly determined by the poses that comprise the kinematic motion, we can use the same tri-plane features for the entire sequence and directly optimize them. However, real world objects have time or pose dependent non-rigid deformation such as clothes and facial expression change in a single sequence. Therefore, the tri-plane features should change depending on time and pose. Recently, many methods for time dependent NeRF, also known as Dynamic NeRF, have been proposed to handle time dependent deformation. Among them, we use a technique based on deformation fields Pumarola et al. [2021], Park et al. [2021]. Deformation field based methods Pumarola et al. [2021], Park et al. [2021] learn a mapping network from observation space to canonical space, and learn the NeRF in the canonical frame. Since learning the deformation field with an MLP is expensive, we also approximate it with tri-planes. We approximate the deformation in 3D space by independent 2D deformations in each tri-plane. First, a StyleGAN2 Karras et al. [2020] generator takes positionally encoded time $t$ and a rotation matrix of each part $R_k$ and generates 6-channel images which represent the relative 2D deformation from the canonical space of each tri-plane feature. To save computation, instead of applying the deformation to the input 3D location $\mathbf{x}$, the number of which is proportional to $K$ and the cube of the resolution, we warp the entire tri-plane features based on the generated deformation fields beforehand, and lookup the feature at $\mathbf{x}$. We use constant tri-plane probabilities $P$ for all frames since the object shares the same coarse part shape throughout the entire sequence. The remaining networks are the same. We refer to this method as D-ENARF.

### 3.5 GAN

To condition the generator on latent vectors, we utilize a StyleGAN2 Karras et al. [2020] based generator to produce tri-plane features. Instead of generating deformation fields, we generate the tri-planes directly by the generator. We condition each layer of the proposed ENARF by the latent vector with a modulated convolution Karras et al. [2020]. Since the proposed tri-plane based generator can only represent the foreground object, we use an additional StyleGAN2 based generator for the background.

We randomly sample latent vectors for our tri-plane generator and background generator: $\mathbf{z} = (\mathbf{z}_{\text{tri}}, \mathbf{z}_{\text{ENARF}}, \mathbf{z}_b) \sim \mathcal{N}(0, I)$, where $\mathbf{z}_{\text{tri}}$, $\mathbf{z}_{\text{ENARF}}$, and $\mathbf{z}_b$ are latent vectors for the tri-plane generator, ENARF, and background generator, respectively. The tri-plane generator $G_{\text{tri}}$ generates tri-plane feature $F$ and part probability $P$ from randomly sampled $\mathbf{z}_{\text{tri}}$ and bone length $\{l_k\}_{k=1:K}$. $G_{\text{tri}}$ takes $l_k$ as inputs to account for the diversity of bone lengths.

$$F, P = G_{\text{tri}}(\mathbf{z}_{\text{tri}}, \{l_k\}_{k=1:K}) \tag{8}$$

The ENARF based foreground generator $G_{\text{ENARF}}$ generates the foreground RGB image $\mathbf{C}_f$ and mask $\mathbf{M}_f$ from the generated tri-planes, and the background generator $G_{\text{b}}$ generates background RGB image $\mathbf{C}_b$.

$$\mathbf{C}_f, \mathbf{M}_f = G_{\text{ENARF}}(\mathbf{z}_{\text{ENARF}}, \{l_k, R_k, \mathbf{t}_k\}_{k=1:K}), \mathbf{C}_b = G_{\text{b}}(\mathbf{z}_b) \tag{9}$$

The final output RGB image is $\mathbf{C} = \mathbf{C}_f + \mathbf{C}_b * (1 - \mathbf{M}_f)$, which is a composite of $\mathbf{C}_f$ and $\mathbf{C}_b$.

To handle the diversity of bone lengths, we replace Equation 5 with one normalized by the length of the bone: $\mathbf{x}_k^c = \frac{l_k^c}{l_k} R_k^c \mathbf{x}_k + \mathbf{t}_k^c$, where $l_k^c$ is the bone length of the $k^{\text{th}}$ part in the canonical space.

We optimize these generator networks with GAN training. We use a bone loss in addition to an adversarial loss on images, R1 regularization on the discriminator Mescheder et al. [2018], and L2 regularization on the tri-planes. The bone loss ensures that an object is generated in the foreground. Based on the input pose of the object, a skeletal image $B$ is created, where pixels with skeletons are 1 and others are 0, and the generated mask $M$ at pixels with skeletons is

Table 1: Quantitative comparison on dynamic scenes.

| | Cost | | | Novel view | | | Novel pose | | |
|---|---|---|---|---|---|---|---|---|---|
| | #Memory | #FLOPS | Time(s) | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| Animatable NeRF Peng et al. [2021b] | - | - | **0.42** | 28.28 | 0.9484 | 0.05818 | 29.09 | 0.9507 | 0.05706 |
| NARF Noguchi et al. [2021]. | 283.9GB | 15.9T | 2.17 | 30.62 | 0.9625 | 0.05228 | 29.51 | **0.959** | 0.05208 |
| ENARF | 27.0GB | 71.7G | 0.47 | 31.94 | 0.9655 | 0.04792 | 29.66 | 0.953 | 0.05702 |
| D-ENARF | 27.6GB | 354G | 0.49 | **32.93** | **0.9713** | **0.03718** | **30.06** | 0.9396 | **0.05205** |
| ENARF w/o selector | **23.0GB** | **70.2G** | 0.43 | 29.16 | 0.9493 | 0.07234 | 27.9 | 0.9377 | 0.08316 |
| ENARF w/ MLP selector | 83.2GB | 337G | 1.13 | 32.27 | 0.9684 | 0.04633 | 29.74 | 0.9573 | 0.05228 |

made close to 1.

$$\mathcal{L}_{\text{bone}} = \frac{\sum_{r\in\mathcal{R}}(1-M)^2 B}{\sum_{r\in\mathcal{R}} B}. \tag{10}$$

Additional details are provided in the appendix. The final loss is the linear combination of these losses.

## 4 Experiments

Our experimental results are presented in two parts. First, the proposed Efficient NARF (ENARF) is compared to the state-of-the-art methods NARF Noguchi et al. [2021] and Animatable NeRF Peng et al. [2021b] in terms of both efficiency and effectiveness in Section 4.1. Ablation studies on the deformation modeling and design choices for the selector are also discussed. Second, in Section 4.2, we present our results of using adversarial training on ENARF, namely, ENARF-GAN and compare it with baselines based on VAE Noguchi et al. [2021] and StyleNARF Gu et al. [2022], both qualitatively and quantitatively. Ablation studies are conducted to evaluate the effectiveness of the pose prior selection and the generalization ability of the proposed framework.

### 4.1 Training on a Dynamic Scene

Following the training setting in Animatable NeRF Peng et al. [2021b], we train our ENARF model on synchronized multi-view videos of a single moving articulated object. The ZJU mocap dataset Peng et al. [2021a] consisting of three subjects (313, 315, 386) is used for training. We use the same pre-processed data provided by the official implementation of Animatable NeRF. All images are resized to $512 \times 512$. We use 4 views and the first 80% of the frames for training, and the remaining views or frames for testing. In this setting, the ground truth camera and articulated object poses, as well as the ground truth foreground mask, are given for each frame. More implementation details can be found in the appendix.

First, we compare our method with the state-of-the-art supervised methods NARF Noguchi et al. [2021] and Animatable NeRF Peng et al. [2021b]. Note that our method and NARF Noguchi et al. [2021] only need ground truth kinematic pose parameters (joint angles and bone lengths) as inputs, while Animatable NeRF needs the ground truth SMPL mesh parameters for both training and inference. In addition, Animatable NeRF requires additional training on novel poses to render novel-pose images, which is not necessary for our model.

Table 1 shows the quantitative results. To compare the efficiency between models, we examine the GPU memory, FLOPS, and the running time used to render an entire image of resolution $512 \times 512$ on a single A100 GPU as evaluation metrics. To compare the quality of synthesized images under novel view and novel pose settings, PSNR, SSIM Wang et al. [2004], and LPIPS Zhang et al. [2018] are used as evaluation metrics. Table 1 shows that the proposed Efficient NARF achieves competitive or even better performance compared to existing methods with far fewer FLOPS and 4.6 times the speed of the original NARF. Although the runtime of ENARF is a bit slower than Animatable NeRF (0.05s), its performance is far superior under both novel view and novel pose settings. In addition, it does not need extra training on novel poses. Our dynamic model D-ENARF further improves the performance of ENARF with little increased overhead in inference time, and outperforms the state-of-the-arts Animatable NeRF and NARF by a large margin. Qualitative results for novel view and pose synthesis are shown in Figure 3. ENARF produces much sharper images than NARF due to more efficient feature extraction. D-ENARF, which utilizes deformation fields, further improves the render quality. In summary, the proposed D-ENARF method achieves better performance in both image quality and computational efficiency.

**Ablation Study** To evaluate the effectiveness of the tri-plane based selector, we compare our method against models using an MLP selector or without a selector. A quantitative comparison is provided in Table 1, and a qualitative comparison is provided in the appendix. Although an MLP based selector improves the metrics a bit, it results in a significant increase in testing time. In contrast, the model without a selector is unable to learn clean/sharp part shapes

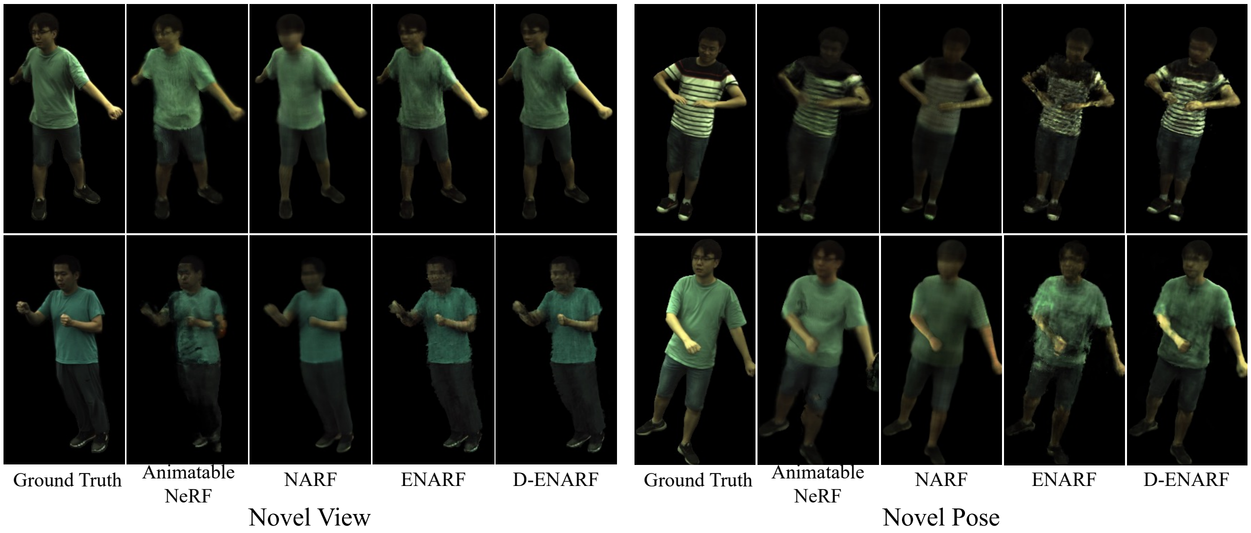

Figure 3: Qualitative comparison of novel view and pose synthesis.

Table 2: Quantitative comparison on generative models. * indicates that the methods are modified from the cited papers.

| | FID↓ | FG-FID↓ | Depth↓ | PCKh@0.5 ↓ |
|---|---|---|---|---|
| ENARF-VAE Noguchi et al. [2021]*. | - | 63.0 | **3.2** | **0.886** |
| StyleNARF Gu et al. [2022]*. | **20.8** | - | 16.5 | 0.867 |
| ENARF-GAN | 22.6 | **21.3** | 8.8 | 0.855 |
| ENARF-GAN CMU pri. | 24.2 | 25.6 | 12.8 | 0.853 |
| ENARF-GAN rand. pri. | 25.9 | 37.0 | 13.8 | 0.798 |
| w/ trunc. $\psi = 0.4$ | 26.5 | 24.1 | 8.0 | 0.881 |

and textures, because a 3D location will inappropriately be affected by all parts without a selector. These results indicate that our tri-plane based selector is efficient and effective.

### 4.2 Unsupervised Learning with GAN

In this section, we train the proposed efficient NARF using GAN objectives without any image-pose pairs or mask annotations.

**Comparison with Baselines** Since this is the first work to learn an articulated representation without image-pose supervision or mesh shape priors, no exact competitor exists. We thus compare our method against two baselines. The first is a supervised baseline called ENARF-VAE Noguchi et al. [2021], where a ResNet50 He et al. [2016] based encoder estimates the latent vectors $\mathbf{z}$ from images, and the efficient NARF based decoder decodes the original images from estimated latent vectors $\mathbf{z}$ and ground truth pose configurations $o$. These networks are trained with the reconstruction loss defined in Equation 7 using images *without background*. The second model is called StyleNARF, which is a combination of the original NARF and the state-of-the-art high resolution 3D-aware image generation model called StyleNeRF Gu et al. [2022]. To reduce the computational cost, the original NARF first generates low resolution features using volumetric rendering. Subsequently, a 2D CNN based network upsamples them into final images. Additional details are provided in the appendix. Please note that ENARF-VAE cannot handle the background, and StyleNARF loses 3D consistency and thus cannot generate high resolution geometry.

We use the SURREAL dataset Varol et al. [2017] for comparison. It is a synthetic human image dataset with a resolution of 128 × 128. Dataset details are given in the appendix. For the pose prior, we use the ground truth pose distribution of the training dataset, where we randomly sample poses from the entire dataset. Please note that we do not use image-pose pairs for these unsupervised methods.

Quantitative results are shown in Table 2. We measure image quality with the Fréchet Inception Distance (FID) Heusel et al. [2017] metric. To better evaluate the quality of foreground, we use an extra metric called FG-FID that replaces the background with black color, using the generated or ground truth mask. We measure depth plausibility by comparing the real and generated depth map. Although there is no ground truth depth for the generated images, the depth generated from a pose would have a similar depth to the real depth that arises from the same pose. We compare the

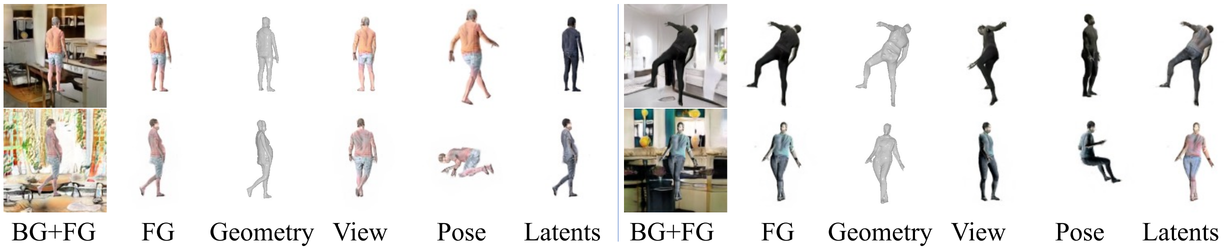

Figure 4: Learned geometry and disentangled representations on the SURREAL dataset by ENARF-GAN. For each of the generated results, the leftmost three columns show the generated images with background, foreground, and corresponding geometry. The rightmost three images show the results of changing only the viewpoint, object pose, and latent variables for the same results, respectively.

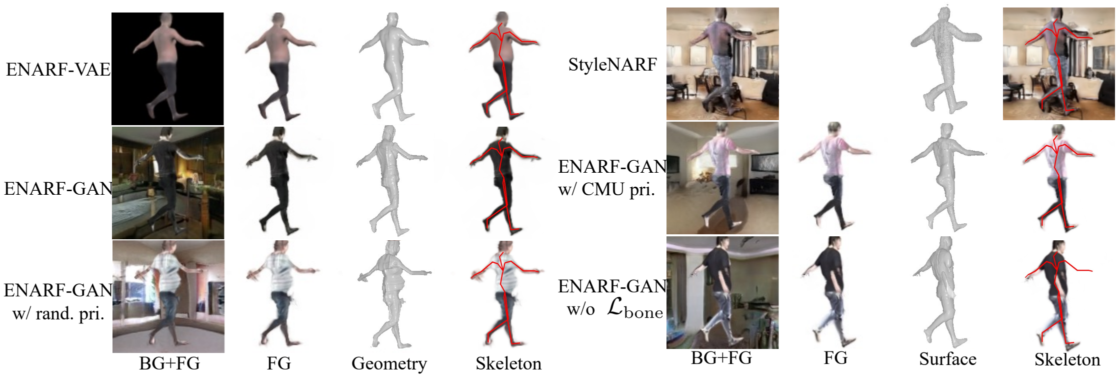

Figure 5: Qualitative comparison on generative models.

L2 norm between the inverse depth generated from poses sampled from the dataset and the real inverse depth of them. Finally, we measure the correspondence between pose and appearance. We apply an off-the-shelf 2D human keypoint estimator Contributors [2020] on generated images and compute the Percentage of Correct Keypoints (PCK), which is commonly used for evaluating 2D pose estimators, between the input poses and the estimated poses. We report the averaged PCKh@0.5 metric Andriluka et al. [2014] for all keypoints. Qualitative results are shown in Figure 5. ENARF-VAE produces the most plausible depth/geometry and learns the most accurate pose conditioning among the three. However, its FID is worse and the images lack photorealism. While styleNARF achieves the best FID among the three, it learns geometry at a low resolution, so it cannot explicitly render only the foreground or generate accurate geometry of the generated images. In contrast, our method performs volumetric rendering at the output resolution, and the generated geometry perfectly matches the generated foreground image.

**Using Different Pose Distribution** Obtaining a ground truth pose distribution of the training images is not feasible for in-the-wild images or new categories. Thus, we train our model with a pose distribution different from the training images. Here, we consider two pose prior distributions. The first uses poses from CMU Panoptic Joo et al. [2015] as a prior, which we call the CMU prior. During training, we randomly sample poses from the entire dataset. In addition, to show that our method works without collecting actual human motion capture data, we also create a much simpler pose prior. We fit a multi-variate Gaussian distribution on each joint angle of the CMU Panoptic dataset, and we use randomly sampled poses from the distribution for training. Each Gaussian distribution only defines the rotation angle of each part, which can be easily constructed for novel objects. We call this the random prior.

Quantitative and qualitative results are shown in Table 2 and Figure 5. We can confirm that even when using the CMU prior, we can learn pose-controllable 3D representations with just a slight sacrifice in image quality. When using the random prior, the plausibility of the generated images and the quality of the generated geometry are worse. This may be because the distribution of the random prior is so far from the distribution of poses in the dataset that the learned space of latent vectors too often falls outside the distribution of the actual data. Therefore, we used the truncation trick Karras et al. [2019] to restrict the diversity of the latent space, and the results are shown in the bottom row of Table 2. By using the truncation trick, even with a simple prior, we can eliminate latent variables outside the distribution and improve the quality of the generated images and geometry. Further experimental results on truncation are given in the appendix.

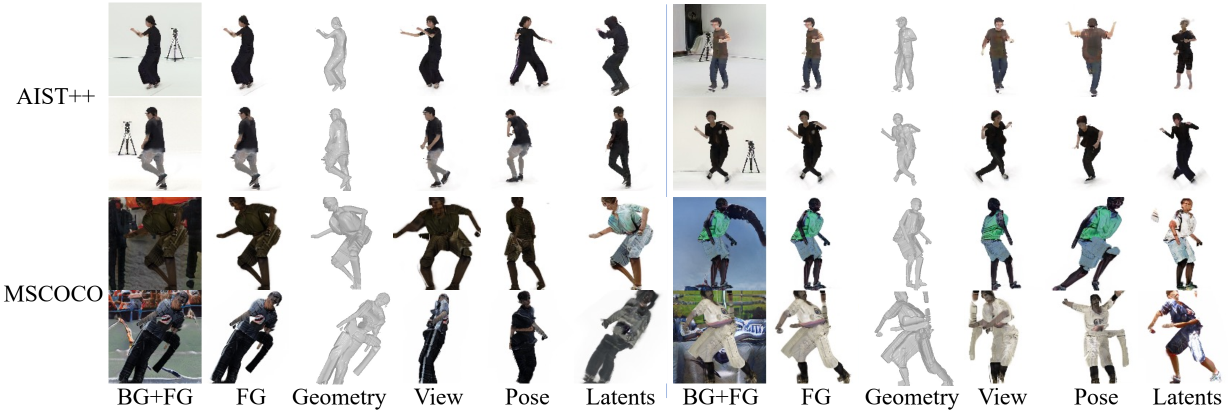

Figure 6: Qualitative results on AIST++ and MSCOCO.

**Additional Results on Real Images** To show the generalization ability of the proposed framework, we train our model on two real image datasets, namely AIST++ Li et al. [2021b] and MSCOCO Lin et al. [2014]. AIST++ is a dataset of dancing persons with relatively simple background. We use the ground truth pose distribution for training. MSCOCO is a large scale in-the-wild image dataset. We choose images capturing roughly the whole human body and crop them around the persons. Since 3D pose annotations are not available for MSCOCO, we use poses in CMU Panoptic as the pose prior. Please note that we do not use any image-pose or mask supervisions for training. Qualitative results are shown in Figure 6. Experimental results with AIST++, which has a simple background, show that it is possible to generate detailed geometry and images with independent control of viewpoint, pose, and appearance. For MSCOCO, two successful and two unsuccessful results are shown in Figure 6. MSCOCO is a very challenging dataset because of the complex background, the lack of clear separation between foreground objects and background, and the many occlusions. Although our model does not always produce plausible results, it is possible to generate geometry and control each element independently. As an initial attempt, the results are promising.

## 5 Conclusion

In this work, we propose a novel unsupervised learning framework for 3D geometry-aware articulated representations. We showed that our framework is able to learn representations with controllable viewpoint and pose. We first propose a computationally efficient neural 3D representation for articulated objects by adapting the tri-plane representation to NARF, then show it can be trained with GAN objectives without using ground truth image-pose pairs or mask supervision. However, the resolution and the quality of the generated images are still limited compared to recent NeRF-based GAN methods. Future work includes incorporating the neural rendering techniques proposed in those methods to generate photorealistic high quality images while preserving the 3D consistency.

## 6 Acknowledgement

This work was supported by D-CORE Grant from Microsoft Research Asia and partially supported by JST AIP Acceleration Research JPMJCR20U3, Moonshot R&D Grant Number JPMJPS2011, JSPS KAKENHI Grant Number JP19H01115, and JP20H05556 and Basic Research Grant (Super AI) of Institute for AI and Beyond of the University of Tokyo. We would like to thank Haruo Fujiwara, Lin Gu, and Yuki Kawana for helpful discussions.

## References

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.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. GRAF: Generative radiance fields for 3D-aware image synthesis. In *NeurIPS*, 2020.

Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *CVPR*, 2021.

Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021.

Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. StyleNeRF: A style-based 3D aware generator for high-resolution image synthesis. In *ICLR*, 2022.

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022.

Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *ICCV*, 2021.

Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-NeRF: Articulated neural radiance fields for learning human shape, appearance, and pose. In *NeurIPS*, 2021.

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*, 2021a.

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 *ICCV*, 2021b.

Alec Jacobson, Zhigang Deng, Ladislav Kavan, and John P Lewis. Skinning: Real-time shape deformation (full text not available). In *ACM SIGGRAPH 2014 Courses*. 2014.

Doug L James and Christopher D Twigg. Skinning mesh animations. *ACM Transactions on Graphics (TOG)*, 24(3): 399–407, 2005.

John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000.

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)*, 34(6):1–16, 2015.

Nikolas Hesse, Sergi Pujades, Michael J Black, Michael Arens, Ulrich G Hofmann, and A Sebastian Schroeder. Learning and tracking the 3D body shape of freely moving infants from rgb-d sequences. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2540–2551, 2019.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019.

Ahmed AA Osman, Timo Bolkart, and Michael J Black. Star: Sparse trained articulated human body regressor. In *ECCV*. Springer, 2020.

Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *CVPR*, 2017.

Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, 2019.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.

Boyang Deng, John P Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. NASA neural articulated shape approximation. In *ECCV*, 2020.

Aljaz Bozic, Pablo Palafox, Michael Zollhofer, Justus Thies, Angela Dai, and Matthias Nießner. Neural deformation graphs for globally-consistent non-rigid reconstruction. In *CVPR*, 2021.

Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Neural-GIF: Neural generalized implicit functions for animating people in clothing. In *ICCV*, 2021.

Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. SNARF: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *ICCV*, 2021.

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 *NeurIPS*, 2021.

Gengshan Yang, Minh Vo, Neverova Natalia, Deva Ramanan, Vedaldi Andrea, and Joo Hanbyul. BANMo: Building animatable 3D neural models from many casual videos. In *CVPR*, 2022.

Atsuhiro Noguchi, Umar Iqbal, Jonathan Tremblay, Tatsuya Harada, and Orazio Gallo. Watch it move: Unsupervised discovery of 3D joints for re-posing of articulated objects. In *CVPR*, 2022.

Artur Grigorev, Karim Iskakov, Anastasia Ianina, Renat Bashirov, Ilya Zakharkin, Alexander Vakhitov, and Victor Lempitsky. Stylepeople: A generative model of fullbody human avatars. In *CVPR*, 2021.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *NeurIPS*, 2019.

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021.

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *CVPR*, 2021.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *CVPR*, 2021a.

Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, 2021.

Thiemo Alldieck, Hongyi Xu, and Cristian Sminchisescu. imghum: Implicit generative models of 3d human shape and articulated pose. In *ICCV*, 2021.

Hongyi Xu, Thiemo Alldieck, and Cristian Sminchisescu. H-NeRF: Neural radiance fields for rendering and temporal reconstruction of humans in motion. In *NeurIPS*, 2021.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020.

Tianhan Xu, Yasuhiro Fujita, and Eiichi Matsumoto. Surface-aligned neural radiance fields for controllable 3d human synthesis. In *CVPR*, 2022.

Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *ICCV*, 2021.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. PlenOctrees for real-time rendering of neural radiance fields. In *ICCV*, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *NeurIPS*, 2021.

Attila Szabó, Givi Meishvili, and Paolo Favaro. Unsupervised generative 3D shape learning from natural images. *arXiv preprint arXiv:1910.00287*, 2019.

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NeurIPS*, 2016.

Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3D representations. In *NeurIPS*, 2018.

Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2D views of multiple objects. In *2017 International Conference on 3D Vision (3DV)*, 2017.

Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3D shape from adversarial rendering. In *ICCV*, 2019.

Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019.

Thu H Nguyen-Phuoc, Christian Richardt, Long Mai, Yongliang Yang, and Niloy Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *NeurIPS*, 2020.

Atsuhiro Noguchi and Tatsuya Harada. RGBD-GAN: Unsupervised 3D representation learning from natural image datasets via rgbd image synthesis. In *ICLR*, 2020.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? 2018.

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*, 13(4):600–612, 2004.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

MMPose Contributors. Openmmlab pose estimation toolbox and benchmark. `https://github.com/open-mmlab/mmpose`, 2020.

Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.

Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *ICCV*, 2015.

Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *ICCV*, 2021b.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

Adam Bielski and Paolo Favaro. Emergence of object segmentation in perturbed generative models. In *NeurIPS*, 2019.

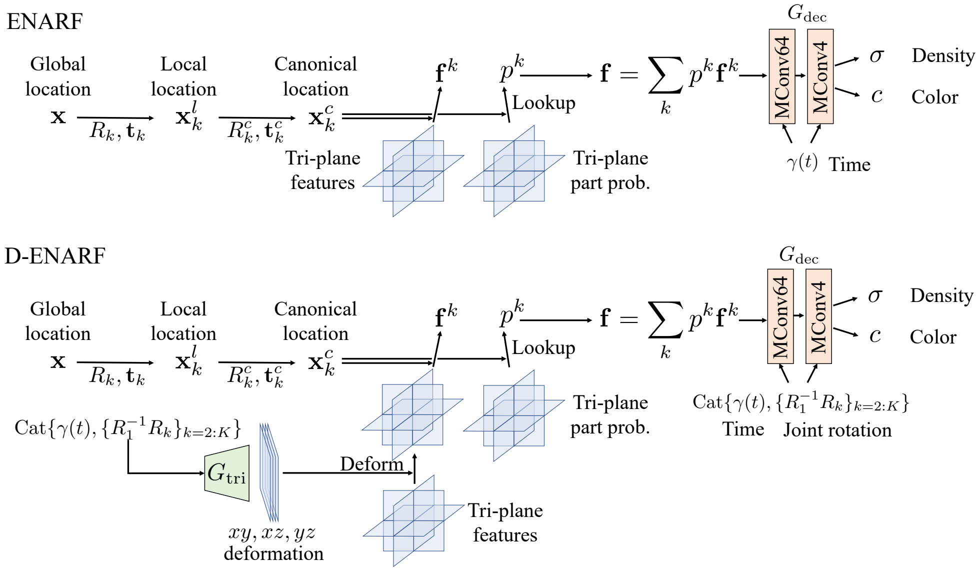

Figure 7: Network details of ENARF and D-ENARF. MConv denotes Modulated Convolution Karras et al. [2020].

# A ENARF Implementation details

## A.1 Model Details

Figure 7 illustrates the learning pipeline of our ENAFR and D-ENAFR models. The decoder network $G_{\text{dec}}$ is implemented with a modulated convolution as in Karras et al. [2020]. For the ENARF model, $G_{\text{dec}}$ is conditioned on the time input $t$ with positional encoding $\gamma(*)$. Following Mildenhall et al. [2020], we use 10 frequencies in the positional encoding. For the D-ENARF model, rotation matrices are additionally used as input. To efficiently learn the deformation field in our D-ENARF model, additional tri-plane features are learned with a StyleGAN2 Karras et al. [2020] based generator. Each feature represents the relative transformation from the canonical frame in pixels. The learned deformation field is used to deform the canonical features to handle the non-rigid deformation for each time and pose.

We used the coarse-to-fine sampling strategy as in Mildenhall et al. [2020] to sample the rendering points on camera rays. Instead of using two separate models to predict points at coarse and fine stages respectively Mildenhall et al. [2020], we use a single model to predict sampled points at both stages. Specifically, for each ray, 48 and 64 points are sampled at the coarse and fine stages, respectively.

## A.2 Efficient Implementation

For efficiency, we introduce a weak shape prior for the part occupancy probability $p^k$ in Section 3.3 of the main paper. Specifically, if a 3D location $\mathbf{x}_k^c$ in the canonical space is outside of a cube with one side of $2a$ located at the center of the part $\mathbf{p}_k^c$, we set $p^k$ to 0, namely, $p^k \leftarrow 0$ if $\max(|\mathbf{x}_k^c - \mathbf{p}_k^c|) > a$. We set $a$ to $\frac{1}{3}$ meter for all parts. This weak shape prior is used for all NARF-based methods. Since we do not have to compute the intermediate feature $\mathbf{f}_k$ (in Equation 6 in the main paper) for the points with part probability $p_k = 0$, the overall computational cost for feature generation is significantly reduced. We implement this efficiently by (1) gathering the valid ($p_k > 0$) canonical positions $\mathbf{x}_k^c$ and compute intermediate features $\mathbf{f}_k$ for them, (2) multiplying by $p^k$, and (3) summing up the feature for each part $p_k * \mathbf{f}_k$ with a scatter_add operation.

## A.3 Training Details

We use the Adam Kingma and Ba [2015] optimizer with an equalized learning rate Karras et al. [2018] of 0.001. The learning rate decay rate is set to 0.99995. The ray batch sizes are set to 4096 for ENARF and 512 for NARF. The ENARF model is trained for 100,000 iterations and the NARF model is trained for 200,000 iterations with a batch size 16. The training takes 15 hours on a single A100 GPU for ENARF and 24 hours for NARF.

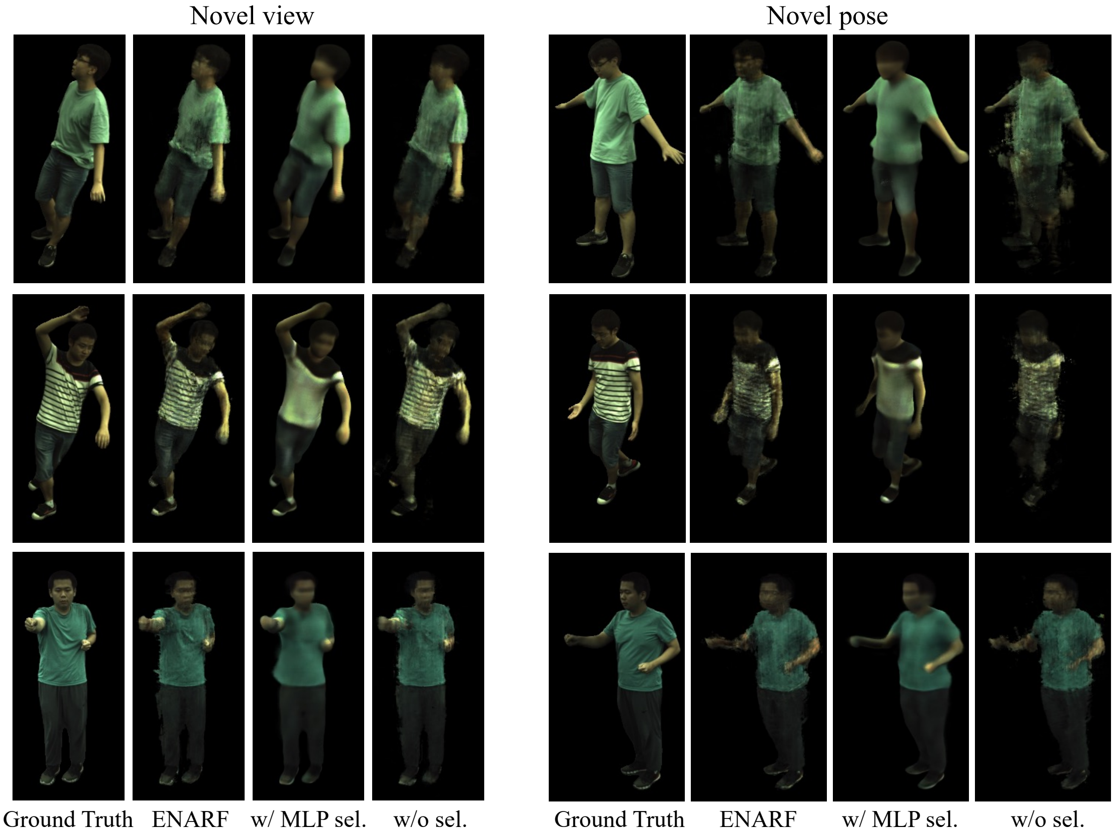

Figure 8: Ablation study on selector.

## B Ablation Study on Selector (Section 4.1)

To show the effectiveness of the tri-plane based selector, we compared our model with a model without the selector and a model with an MLP based selector. In the model without a selector, we simply set $p^k = \frac{1}{K}$. In the MLP based selector, we used a two-layer MLP with a hidden layer dimension of 10 for each part, as in NARF Noguchi et al. [2021]. The quantitative and qualitative comparisons are provided in Table 1 in the main paper and Figure 8, respectively. The model without a selector cannot generate clear and sharp images because the feature of any location to compute the density and color is evenly contributed by all parts. The MLP based selector helps learn independent parts. However, the generated images look blurry compared to ours. Moreover, it requires much more GPU memory and FLOPS for training and testing. In summary, the proposed tri-plane based selector is superior in terms of both effectiveness and efficiency.

## C Ablation Study on View Dependency

Following Efficient NeRF Chan et al. [2022], our model does not take the view direction as input, i.e. the color is not view dependent. We note that this implementation is inconsistent with the original NeRF Mildenhall et al. [2020] model that takes the view direction as an input. Here, we do an ablation study on the view dependent input in our model. Specifically, the positional encoding is added to the view direction $\gamma(\mathbf{d})$ and additionally used as the input of $G_{\text{dec}}$. Experimental results indicate that additional view direction input leads to darker images and degrades the performance. We thus do not use view direction as input of our models by default. Quantitative and qualitative comparisons are provided in Table 3 and Figure 9, respectively.

Table 3: Quantitative comparison on dynamic scenes.

| | Novel view | | | Novel pose | | |
|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | PSNR↑ | SSIM↑ | LPIPS↓ |
| ENARF | $\underline{31.94}$ | 0.9655 | 0.04792 | 29.66 | $\underline{0.953}$ | 0.05702 |
| D-ENARF | **32.93** | **0.9713** | **0.03718** | **30.06** | 0.9396 | $\underline{0.05205}$ |
| ENARF w/ view | 29.98 | 0.9603 | 0.05129 | 28.42 | 0.9497 | 0.06005 |
| D-ENARF w/ view | 31.05 | $\underline{0.9668}$ | $\underline{0.04162}$ | 29.3 | **0.9563** | **0.04711** |

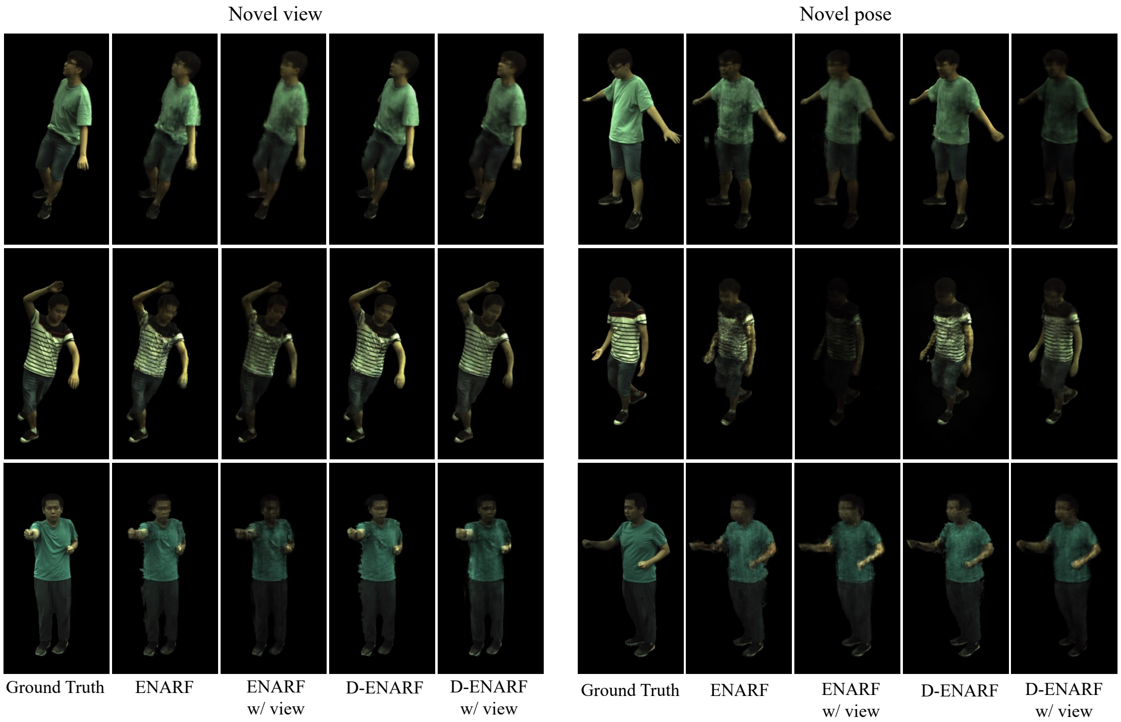

Figure 9: Ablation study on view direction.

## D Implementation Details of ENAFR-GAN

### D.1 Bone Region Loss $\mathcal{L}_{\text{bone}}$ (Section 3.5)

One obvious positional constraint for the foreground object is that it should be generated to cover at least the regions of bones defined by the input pose configuration. This motivates us to propose a bone region loss $\mathcal{L}_{\text{bone}}$ on the foreground mask $M$ to facilitate model training. First, we create a skeletal image $B$ from an input pose configuration. Examples of $B$ are visualized in Figure 10. The skeletal image $B$ is an image in which each joint and its parent joint are linked by a straight line of 1-pixel width. The bone region loss $\mathcal{L}_{\text{bone}}$ is then defined to penalize any overlaps between the background region $(1 - M)$ and the bone regions.

$$\mathcal{L}_{\text{bone}} = \frac{\sum(1-M)^2 B}{\sum B}. \tag{11}$$

We show the comparison results of training with or without $\mathcal{L}_{\text{bone}}$ in Table 4 and Figure 11. Although the foreground image quality is comparable, Figure 11 shows that the generated images are not well aligned with the input pose without $\mathcal{L}_{\text{bone}}$. In Table 4, we can see that PCKh@0.5 metric for the wrist keypoint becomes worse without $\mathcal{L}_{\text{bone}}$.

### D.2 Training Details of ENARF-GAN

We set the dimension of $\mathbf{z}_{\text{tri}}$ and $\mathbf{z}_b$ to 512, and $\mathbf{z}_{\text{ENARF}}$ to 256. We use the Adam optimizer with an equalized learning rate Karras et al. [2018] of 0.0004. We set the batch size to 12 and train the model for 300,000 iterations. The training

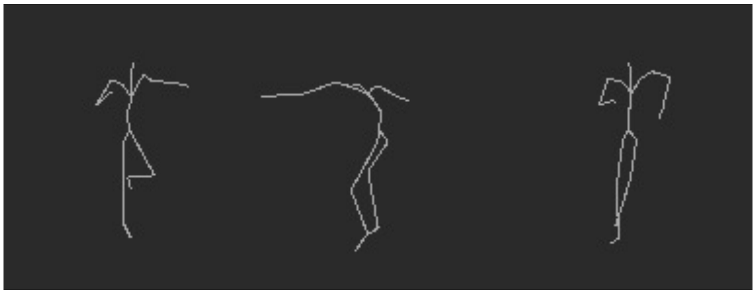

Figure 10: Examples of bone images.

Table 4: Quantitative comparison on generative models. * indicates that the methods are modified from the cited papers.

|  | FID↓ | FG-FID↓ | Depth↓ | PCKh@0.5 mean ↑ | PCKh@0.5 wrist ↑ |
|---|---|---|---|---|---|
| ENARF-GAN | 22.6 | 21.3 | 8.881 | 0.8555 | 0.6753 |
| ENARF-GAN w/o $\mathcal{L}_{\text{bone}}$ | 24.3 | 21.0 | 14.81 | 0.8587 | 0.6302 |

takes 4 days on a single A100 GPU. In the testing phase, a 128×128 image is rendered in 80ms (about 12 fps) using a single A100 GPU.

### D.3 Shifting Regularization

To prevent the background generator from synthesizing the foreground object, we apply shifting regularization Bielski and Favaro [2019] for the background generator. We randomly shift and crop the background images before overlaying foreground images on them. If the background generator synthesizes the foreground object, the shifted images can be easily detected by the discriminator $D$, which encourages the background generator to focus only on the background. We generate $128 \times 256$ background images and randomly crop $128 \times 128$ images.

## E Pose Consistency Metric

We propose a pose consistency metric to evaluate the consistency between the input pose and the pose of the generated image. Specifically, we use an off-the-shelf 2D human pose estimator Contributors [2020] pretrained on the MPII human pose dataset Andriluka et al. [2014] to detect 2D pose in the generated image and compare the detected pose with the input pose under the metric of PCKh@0.5 Andriluka et al. [2014]. We use ankle, knee, hip, neck, wrist, elbow, and shoulder joints for evaluation.

## F Truncation Trick (Section 4.2)

The truncation trick Karras et al. [2019] can improve the quality of the images by limiting the diversity of the generated images. Figure 12 shows the results of generating images with the truncation $\psi$ for the tri-plane generator $G_{\text{tri}}$ set to 1.0, 0.7, and 0.4. When truncation $\psi$ is set to 1.0, multiple legs and arms will be generated. Smaller $\psi$ helps generate more plausible appearance and shapes of the object.

## G StyleNARF (Section 4.2)

StyleNARF is a combination of NARF Noguchi et al. [2021] and StyleNeRF Gu et al. [2022]. To reduce the computational complexity, it first generates low-resolution features with NARF and then upsamples the features to a higher resolution with a CNN-based generator. Similar to ENARF, StyleNARF can only generate foreground objects, so a StyleGAN2 based background generator $G_{\text{b}}$ is used as in ENARF-GAN. First, we sample latent vectors from a normal distribution, $\mathbf{z} = (\mathbf{z}_{\text{NARF}}, \mathbf{z}_{\text{b}}, \mathbf{z}_{\text{up}}) \sim \mathcal{N}(0, I)$, where $\mathbf{z}_{\text{NARF}}$ is a latent vector for NARF, $\mathbf{z}_{\text{b}}$ is a latent vector for background, and $\mathbf{z}_{\text{up}}$ is a latent vector for the upsampler. Then the NARF model $G_{\text{NARF}}$ generates low-resolution foreground feature $\mathbf{F}_f$ and mask $\mathbf{M}_f$, and $G_{\text{b}}$ generates background feature $\mathbf{F}_b$.

$$\mathbf{F}_f, \mathbf{M}_f = G_{\text{NARF}}(\mathbf{z}_{\text{ENARF}}, \{l_k, R_k, \mathbf{t}_k\}_{k=1:K}), \mathbf{F}_b = G_{\text{b}}(\mathbf{z}_b). \tag{12}$$

The foreground and background are combined using the generated foreground mask $\mathbf{M}_f$ at low resolution.

$$\mathbf{F} = \mathbf{F}_f + \mathbf{F}_b * (1 - \mathbf{M}_f). \tag{13}$$

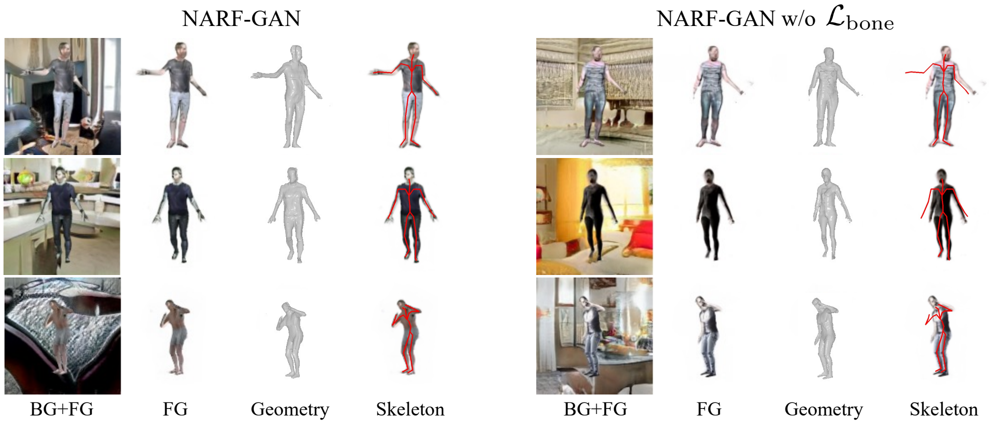

Figure 11: Ablation study on $\mathcal{L}_{\text{bone}}$. From left to right, generated images, foreground images, geometry, and foreground images with input skeletons are visualized.

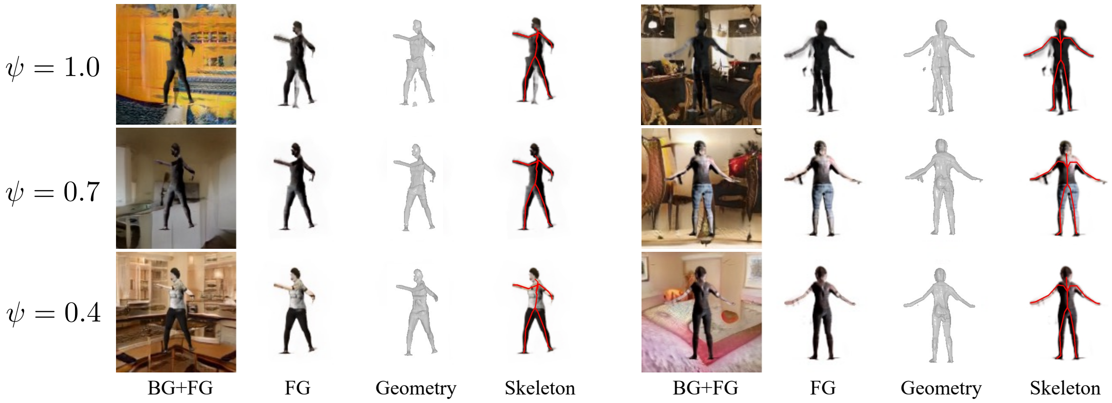

Figure 12: Truncation trick.

The upsampler $G_{\text{up}}$ upsamples the feature $\mathbf{F}$ based on the latent vector $\mathbf{z}_{\text{up}}$ and generates the final output $\mathbf{C}$.

$$\mathbf{C} = G_{\text{up}}(\mathbf{F}, \mathbf{z}_{\text{up}}). \tag{14}$$

All layers are implemented with Modulated Convolution Karras et al. [2020].

# H Datasets

We use images at resolution 128 × 128 for GAN training.

## H.1 SURREAL

We crop the first frame of all videos to 180 × 180 and resize them to 128 × 128 so that the pelvis joint is centered. 68033 images are obtained.

## H.2 AIST++

The images are cropped to 600 × 600 so that the pelvis joint is centered, and then resized to 128 × 128. We sample 3000 frames for each subject, resulting in 90000 images in total.

### H.3 MSCOCO

First, we select the persons whose entire body is almost visible according to the 2D keypoint annotations in MSCOCO. Each selected person is cropped by a square rectangle that tightly encloses the person and it is resized to $128 \times 128$. The number of collected samples is 38727.

## I Pose Distribution

Two pose distributions are used in our experiments. One is the CMU pose distribution, which consists of 390k poses collected in the CMU Panoptic dataset. We follow the pose pre-processing steps in the SURREAL dataset, where the distance between the person and the camera is randomly distributed in a normal distribution of mean 7 meters and variance 1 meter. The pose rotation around the z-axis is uniformly distributed between 0 and $2\pi$.

Another pose distribution used in our experiments is a random pose distribution. First, a multivariate normal distribution is fitted to the angle distribution of each joint under the 390k pose samples of the CMU Panoptic dataset. Then, poses are randomly sampled from the learned multivariate normal distribution and randomly rotated so that the pelvis joint is directly above the medial points of the right and left plantar feet.