Fast-SNARF: A Fast Deformer for Articulated Neural Fields

Xu Chen*[1,2] Tianjian Jiang*[1] Jie Song[1] Max Rietmann[3] Andreas Geiger[2,4] Michael J. Black[2] Otmar Hilliges[1]

[1]ETH Zürich [2]Max Planck Institute for Intelligent Systems, Tübingen [3]NVIDIA [4]University of Tübingen

Abstract—Neural fields have revolutionized the area of 3D reconstruction and novel view synthesis of *rigid* scenes. A key challenge in making such methods applicable to *articulated* objects, such as the human body, is to model the deformation of 3D locations between the rest pose (a canonical space) and the deformed space. We propose a new articulation module for neural fields, Fast-SNARF, which finds accurate correspondences between canonical space and posed space via iterative root finding. Fast-SNARF is a drop-in replacement in functionality to our previous work, SNARF, while significantly improving its computational efficiency. We contribute several algorithmic and implementation improvements over SNARF, yielding a speed-up of $150\times$. These improvements include voxel-based correspondence search, pre-computing the linear blend skinning function, and an efficient software implementation with CUDA kernels. Fast-SNARF enables efficient and simultaneous optimization of shape and skinning weights given deformed observations without correspondences (e.g. 3D meshes). Because learning of deformation maps is a crucial component in many 3D human avatar methods and since Fast-SNARF provides a computationally efficient solution, we believe that this work represents a significant step towards the practical creation of 3D virtual humans.

1 Introduction

3D avatars are an important building block for many emerging applications in the metaverse, AR/VR and beyond. To this end, an algorithm to reconstruct and animate non-rigid articulated objects, such as humans, accurately and quickly is required. This challenging task requires modeling the 3D shape and deformation of the human body – a complex, articulated, non-rigid object. We consider the scenario in which training data, in the form of 3D scans, is available and we model the deformable shape of the human in a canonical space that can be transformed into the posed observation space. For such techniques to be widely applicable, it is paramount that algorithms can learn from 3D scans of people in arbitrary poses without requiring pre-computed correspondence between the input scans. Therefore, inferring the transformation that 3D locations undergo between the posed observation space and some canonical space is the key challenge to attain a model that can be animated.

Static shape modeling has recently seen much progress with the advent of neural fields [39, 42, 43, 49]. Such representations are promising due to their ability to represent complex geometries of arbitrary topology, at arbitrary resolution, by leveraging multi layer perceptrons (MLPs) to encode spatial quantities of interest (e.g. occupancy probabilities) in 3D space. Recent work [43] has further achieved fast reconstruction and real-time view synthesis of rigid scenes with high quality. However, to enable fast *non-rigid* reconstruction and realistic *animation* of articulated objects, a robust and fast articulation module is needed.

Articulation of neural fields is typically modeled via deformation of 3D space, which warps neural fields from a rest pose (canonical space) into any target pose (posed space), leveraging dense deformation fields. Several techniques have been proposed to construct such deformation fields. Building upon traditional mesh-based linear blend skinning (LBS) [26], several works [27, 41, 50, 59, 64] learn dense skinning weight fields in *posed space* and then derive the deformation fields via LBS. While inheriting the smooth deformation properties of LBS, the resulting skinning weight fields cannot generalize to unseen poses, because they are *pose-dependent*, and changes in pose lead to drastic changes to the spatial layout of the deformation field. These changes have not been observed at training time for unseen poses. Another line of work approximates the mapping as piece-wise rigid transformations [16, 45], which suffers from discontinuous artifacts at joints. The mapping could also be approximated based on a skinned base mesh [24], which can lead to inaccuracies due to the mismatch between the

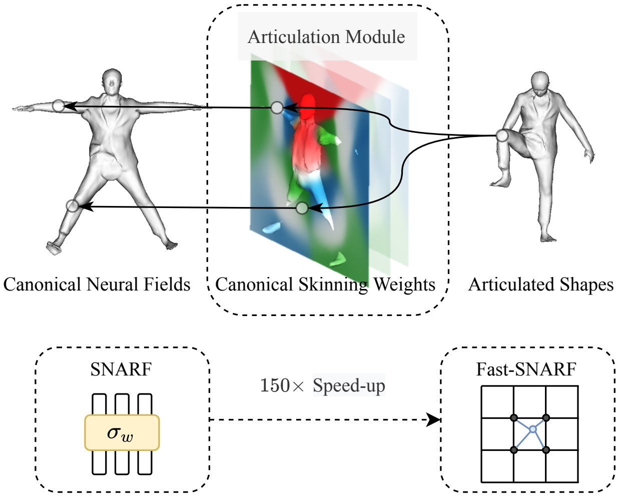


Fig. 1: **Fast-SNARF for Articulated Neural Fields.** Fast-SNARF finds accurate correspondences between canonical space and posed space while being 150× faster than our previous method SNARF [12]. Fast-SNARF enables optimizing shape and skinning weights given deformed observations without correspondences (e.g. 3D meshes).

*Xu Chen and Tianjian Jiang contributed equally.

base mesh and the actual, observed, shape. Additionally, this approach takes the skinning weights from the nearest neighboring point on the base mesh. This leads to ambiguities when the mesh is in self-contact, producing erroneous nearest-neighbor associations.

Our recent work, SNARF [12], overcomes these problems by design in that it learns a skinning weight field in canonical space that is *pose-independent*. This formulation produces natural deformations due to the smooth deformation properties of LBS and generalizes to unseen poses because of the pose-independent canonical skinning weights. Furthermore, in contrast to previous methods [27, 41, 50, 59, 64], pose-independent skinning weights can be learned unsupervised, i.e. without the need for ground-truth skinning weights or other forms of annotations.

However, a major limitation of SNARF is the algorithm's computational inefficiency. While learning a canonical skinning weight field enables generalization, the deformation from posed to canonical space is defined implicitly, and hence can only be determined numerically via iterative root finding. The efficiency of the operations at each root finding iteration plays a critical role in the speed of the overall articulation module. Therefore, computationally expensive operations in SNARF, such as computing LBS and evaluating the skinning weight field, parameterized by an MLP, lead to prohibitively slow speed – learning an animatable avatar from 3D meshes takes 8 hours on high-end GPUs.

In this paper, we propose Fast-SNARF, an articulation module that is fast yet preserves the accuracy and robustness of SNARF. We achieve this by significantly reducing the computation at each root finding iteration in the articulation module. First, we use a compact voxel grid to represent the skinning weight field instead of an MLP. The voxel-based representation can replace MLPs without loss of fidelity because the skinning weight field is naturally smooth, and is pose-independent in our formulation. In addition, exploiting the linearity of LBS, we factor out LBS computations into a pre-computation stage without loss of accuracy. As a result, the costly MLP evaluations and LBS calculations in SNARF are replaced by a single tri-linear interpolation step, which is lightweight and fast. Together with a custom CUDA kernel implementation, Fast-SNARF can deform points with a speed-up of 150x w.r.t. SNARF (from 800ms to 5ms) without loss of accuracy.

In our experiments, we follow the setting of SNARF and learn an animatable avatar, including its shape and skinning weights, from 3D scans in various poses, represented by a pose-conditioned occupancy field parameterized by an MLP. The overall inference and training speed, including both articulation and evaluation of the canonical shape MLP, is increased by $30\times$ and $15\times$ respectively. Note that the speed bottleneck is shifted from articulation (in SNARF) to evaluating the canonical shape MLP (in Fast-SNARF). Fast-SNARF is also faster than other articulation modules and is significantly more accurate, as we show empirically. While we focus on learning occupancy networks, Fast-SNARF can support other neural fields in the same manner that SNARF and its variants have been utilized [28, 31, 66, 73].

We hope Fast-SNARF will accelerate research on articulated 3D shape representations and we release the code on our project webpage [1] to facilitate future research.

Relation to SNARF [12]: This paper is an extension of SNARF [12], a conference paper published at ICCV'21, which models articulation of neural fields. This paper addresses the main limitation of SNARF, i.e. its computational inefficiency via a series of algorithmic and implementation improvements described in Section 4. We provide a speed and accuracy comparison of Fast-SNARF with SNARF and other baseline methods, and thorough ablation studies in Section 4.2.

2 Related Work

2.1 Rigid Neural Fields

Neural fields have emerged as a powerful tool to model complex rigid shapes with arbitrary topology in high fidelity by leveraging the expressiveness of neural networks. These neural networks regress the distance to the surface [49], occupancy probability [39], color [47] or radiance [42] of 3D points. Conditioning on local information such as 2D image features or 3D point cloud features produces more detailed reconstructions than using global features [13, 20, 52, 57, 58]. Such representations can be trained with direct 3D supervision, e.g. ground truth occupancy or distance to the surface, or can be trained indirectly with raw 3D points clouds [3, 7, 19, 59, 69] or 2D images [42, 44, 61, 70].

Fast Rigid Neural Fields: One major limitation of neural field representations is their slow training and inference speeds, mainly due to the fact that multiple evaluations of deep neural networks are necessary to generate images and each of these evaluations is time-consuming. Several approaches have recently been proposed to improve the training [9, 33, 43, 60, 62, 63] and inference speed [18, 22, 54, 71]. The core idea is to leverage explicit representations [52], such as voxel grids or hash tables, to store features for a sparse set of points in space. The dense field can then be obtained by interpolating sparse features and by decoding the features using neural networks. Instead of point locations, these networks take features as input, which are more informative, enabling the network to be shallow and hence more computationally efficient. However, the underlying explicit representations have a fixed spatial layout that limits these methods to rigid shapes.

Our proposed articulation module can deform rigid neural fields to enable non-rigid animation at inference time and enable learning from deformed observations during training. Importantly, our module runs at a comparable speed to recent fast rigid neural field representations (e.g. [43]) and is thus complementary to advancements made in accelerating neural fields.

2.2 Articulation of Neural Fields

Recently, several articulation algorithms for neural fields have been proposed. These methods serve as a foundation for many tasks such as generative modeling of articulated objects or humans [4, 11, 15, 23, 46, 72], and reconstructing

[1] https://github.com/xuchen-ethz/fast-snarf

animatable avatars from scans [12, 16, 32, 40, 41, 59, 64], depth [17, 48, 65], videos [10, 28, 30, 31, 34, 45, 51, 53, 66, 67, 73] or a single image [21, 24, 68]. Compared to mesh-based representations, articulated neural fields naturally model varied topology in various body poses, and can be applied on articulated objects of different categories or humans in different clothing styles without requiring customized templates. In the following, we discuss existing approaches to articulating neural fields.

Part Based Models: One option is to model articulated shapes as a composition of multiple parts [16, 40, 45]. Rigidly transforming these parts according to the input "bone" transformations produces deformed shapes. While preserving the global structure after articulation, the continuity of surface deformations is violated, causing artifacts at the intersections of parts. Moreover, inferring the correct part assignment from raw data is challenging and typically requires ground-truth supervision.

Backward Skinning: Another line of work [27, 41, 59, 64] learns skinning weight fields in deformed space and then derives the backward warping field using LBS to map points in deformed space to canonical points. Such methods are straightforward to implement but inherently suffer from poor generalization to unseen poses. Backward deformation fields are defined in deformed space and, hence, inherently deform with the pose. Existing methods train pose-conditioned MLPs to predict such deformation fields. These deformations can be complex and the mapping from human pose to deformations is very high-dimensional. Small changes in the input pose can produce large changes in output skinning weights, making it difficult to generate deformations that have not been seen during training. Learning such pose-dependent skinning weight fields is also challenging, thus existing methods often rely on strong supervision via ground-truth skinning weights. Moreover, due to the varying spatial configuration, such pose-dependent skinning weights cannot be modeled using acceleration data structures such as explicit voxel grids.

Forward Skinning: Learning the skinning weights in canonical space instead of deformed space is a natural way to resolve the generalization issue. However, deriving the mapping from deformed to canonical points with canonical skinning weights is not straightforward, because the skinning weights of the deformed query points are unknown. Thus, SNARF [12] attains this mapping using an iterative root finding formulation, which finds the canonical points that are forward-skinned to the deformed query location. This formulation enables the articulation of neural fields into arbitrary poses, even those unseen during training. The pose-independent canonical skinning weights as well as the canonical shape can be jointly learned unsupervised without the need for ground-truth skinning weights or shape, avoiding the manual effort needed to define skinning weights and implicitly solving the challenging problem of canonicalizing posed observations. Moreover, multiple canonical correspondences can be found using such methods, which is important to handle self-contact. This forward skinning formulation has already found widespread use in many tasks, such as generative modeling [11], or personalized avatar reconstruction from scans [32], depth [17], or images [28, 31, 66, 73].

However, one major limitation of this formulation is its slow speed due to the expensive computation at each root finding iteration. The original SNARF model relies on an MLP to parameterize the skinning weight field. At each root finding iteration, SNARF requires evaluation of the MLP to compute LBS weights, which is time-consuming. This limitation is further amplified when combining forward skinning with rendering algorithms that require many queries along many rays (cf. [14]). To reduce computation time, existing methods [28, 66] use an explicit mesh to tighten the search space of root finding. However, these methods introduce the overhead of mesh extraction and still require days of training time to learn avatars from images.

We address this problem by using a voxel-based parameterization of the skinning weight field and by factoring out the LBS computation into a pre-computation stage. Since Fast-SNARF does not require mesh extraction in the training loop, it is more versatile and much faster to train than methods that rely on meshes (e.g. [28]) (minutes vs. days). Our method also enables learning the skinning weights.

3 Differentiable Forward Skinning

In this section, we briefly summarize the differentiable forward skinning approach proposed in SNARF [12]. We then discuss Fast-SNARF in Section 4.

General Pipeline: Figure 2 illustrates the general pipeline for modeling articulated neural fields. Given a query point in posed space, an articulation module first finds its correspondences in canonical space according to the input body pose. Then the canonical shape properties are evaluated at the correspondence locations. When multiple correspondences exist, multiple values of these properties are predicted and aggregated into one value as the final output.

Canonical Neural Fields: Canonical shape properties can be modeled using any coordinate-based representation, e.g. occupancy fields [39] or radiance fields [42]. For convenience, we follow SNARF and use occupancy fields as an example. The occupancy field in SNARF [12] is defined as

$$f_{\sigma_f} : \mathbb{R}^3 \times \mathbb{R}^{n_p} \to [0, 1], \tag{1}$$

$$\mathbf{x}, \mathbf{p} \mapsto o. \tag{2}$$

Here f_{σ_f} is the occupancy field that predicts the occupancy probability o for any canonical point $\mathbf{x}$. The parameters of the occupancy field are denoted as σ_f. It can be optionally conditioned on the articulated pose $\mathbf{p}$ to model pose-dependent local deformations such as clothing wrinkles.

Neural Blend Skinning: In SNARF, the articulation is modeled using LBS. To apply LBS to continuous neural fields, a skinning weight field in canonical space is defined as:

$$\mathbf{w}_{\sigma_w} : \mathbb{R}^3 \to \mathbb{R}^{n_b}, \tag{3}$$

where σ_w are the parameters and n_b denotes the number of bones. In SNARF, this field is parameterized as an MLP. However, any other coordinate-based representation can be used instead. Given the skinning weights $\mathbf{w}$ of a 3D point $\mathbf{x}$ and the bone transformations $\boldsymbol{B} = \{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_{n_b}\}$

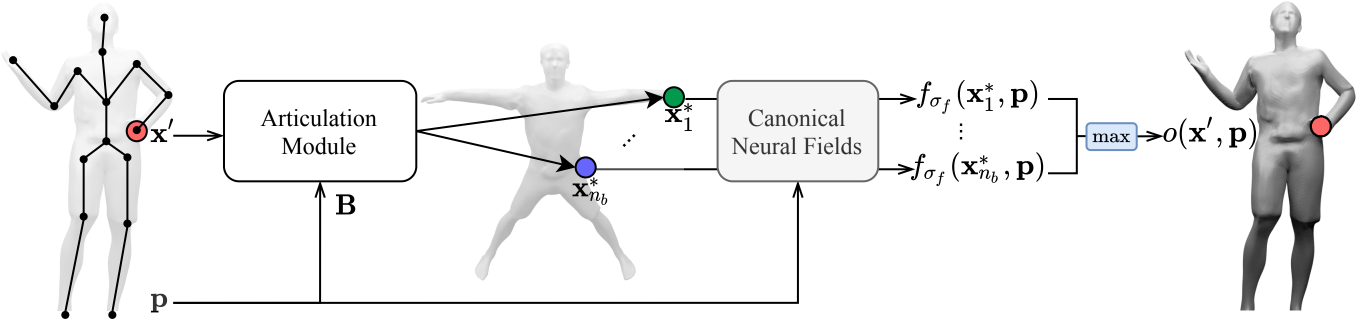


Fig. 2: **General Framework for Articulated Neural Field Representations.** Given a query point, $\mathbf{x}'$, in deformed space and the input pose (represented as joint angles $\mathbf{p}$ and 6D transformations $\mathbf{B}$), an articulation module first finds its canonical correspondences $\mathbf{x}^*$. The canonical shape representation f_{σ_f} then outputs the occupancy probabilities or densities at $\{\mathbf{x}^*\}$ which are finally aggregated to yield the occupancy probability or density of the query point $\mathbf{x}'$.

($\boldsymbol{B}_i \in SE(3)$) that correspond to a particular body pose $\mathbf{p}$, the 6D transformation $\mathbf{T}(\mathbf{x}) \in \mathbb{R}^{3\times 4}$ of a canonical point is determined by the following convex combination:

$$\mathbf{T}(\mathbf{x}) = \sum_{i=1}^{n_b} w_{\sigma_w,i}(\mathbf{x}) \cdot \boldsymbol{B}_i. \tag{4}$$

The deformed point corresponding to the canonical point is then computed as

$$\mathbf{x}' = \mathbf{d}_{\sigma_w}(\mathbf{x}, \boldsymbol{B}) = \mathbf{T}(\mathbf{x}) \cdot \mathbf{x}. \tag{5}$$

Correspondence Search: The canonical skinning weight field and Eq. (5) define the mapping from canonical points to deformed ones, i.e. $\mathbf{x} \to \mathbf{x}'$. However, generating posed shapes requires the inverse mapping, i.e. $\mathbf{x}' \to \mathbf{x}$, which is defined implicitly as the root of the following equation:

$$\mathbf{d}_{\sigma_w}(\mathbf{x}, \boldsymbol{B}) - \mathbf{x}' = \mathbf{0}. \tag{6}$$

The roots of this equation cannot be analytically solved in closed form. Instead, the solution can be attained numerically via standard Newton or quasi-Netwon optimization methods, which iteratively find a location $\mathbf{x}$ that satisfies Eq. (6) (see Fig. 3):

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\mathbf{J}^k)^{-1} \cdot (\mathbf{d}_{\sigma_w}(\mathbf{x}^k, \boldsymbol{B}) - \mathbf{x}'). \tag{7}$$

Here $\mathbf{J}$ is the Jacobian matrix of $\mathbf{d}_{\sigma_w}(\mathbf{x}^k, \boldsymbol{B}) - \mathbf{x}'$. To avoid computing the Jacobian at each iteration, Broyden's method [6] and low-rank approximation $\tilde{\mathbf{J}}$ of $\mathbf{J}^{-1}$ is used.

Handling Multiple Correspondences: Multiple roots, denoted by the set $\{\mathbf{x}_i^*\}$, might exist due to self-contact where multiple canonical correspondences of one deformed point exist (see green and blue points in Fig. 2). Multiple roots are found by initializing the optimization-based root finding procedure with different starting locations and exploiting the local convergence of the optimizer. The initial states $\{\mathbf{x}_i^0\}$ are obtained by transforming the deformed point $\mathbf{x}'$ rigidly to the canonical space for each of the n_b bones, and the initial Jacobian matrices $\{\mathbf{J}_i^0\}$ are the spatial gradients of the skinning weight field at the corresponding initial states:

$$\mathbf{x}_i^0 = \boldsymbol{B}_i^{-1} \cdot \mathbf{x}' \quad \mathbf{J}_i^0 = \left.\frac{\partial \mathbf{d}_{\sigma_w}(\mathbf{x}, \boldsymbol{B})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_i^0} \tag{8}$$

The final set of correspondences is determined by their convergence:

$$\mathcal{X}^* = \{\mathbf{x}_i^* \mid \|\mathbf{d}_{\sigma_w}(\mathbf{x}_i^*, \boldsymbol{B}) - \mathbf{x}'\|_2 < \epsilon\}, \tag{9}$$

where ϵ is the convergence threshold.

Aggregating Multiple Correspondences: Taking the maximum of multiple implicit functions is a standard operator for composing multiple independent implicit shapes into a single one [55, 56]. In our case, each correspondence can be considered as belonging to an independent shape (body part). Thus, the maximum over the occupancy probabilities of all canonical correspondences gives the final occupancy prediction:

$$o'(\mathbf{x}', \mathbf{p}) = \max_{\mathbf{x}^* \in \mathcal{X}^*} \{f_{\sigma_f}(\mathbf{x}^*, \mathbf{p})\}. \tag{10}$$

Intuitively, a deformed point is unoccupied only if all of its canonical correspondences are unoccupied.

Losses: The canonical neural fields and the skinning weights can be learned jointly from observations in the deformed space. SNARF assumes direct 3D supervision and uses the binary cross entropy loss $\mathcal{L}_{\mathrm{BCE}}(o(\mathbf{x}', \mathbf{p}), o_{gt}(\mathbf{x}'))$ between the predicted and ground-truth occupancy for any deformed point. In addition, two auxiliary losses are applied during the first epoch to bootstrap training. SNARF randomly samples points along the bones that connect joints in canonical space and encourages their occupancy probabilities to be one. Moreover, SNARF encourages the skinning weights of all joints to be 1 for their parent bones. These two bootstrapping losses are derived from the skeleton only, which is easy to create in practice and can be shared across different subjects, unlike shape and skinning weights which vary across subjects wearing different clothing. In particular, for human subjects, the skeleton of a minimally clothed SMPL model is sufficient to learn the shape and skinning weights of different clothed humans, without any manual effort needed to re-define the skeleton.

Gradients: To learn the skinning weights $\mathbf{w}_{\sigma_w}$ using a loss applied on the predicted occupancy probability in posed space $\mathcal{L}(o(\mathbf{x}', \mathbf{p}))$, the gradient of $\mathcal{L}$ w.r.t. σ_w is required. Applying the chain rule, the gradient $\frac{\partial \mathcal{L}}{\partial \sigma_w}$ is given by

$$\frac{\partial \mathcal{L}}{\partial \sigma_w} = \frac{\partial \mathcal{L}}{\partial o} \cdot \frac{\partial o}{\partial f_{\sigma_f}} \cdot \frac{\partial f_{\sigma_f}(\mathbf{x}^*)}{\partial \mathbf{x}^*} \cdot \frac{\partial \mathbf{x}^*}{\partial \sigma_w}, \tag{11}$$

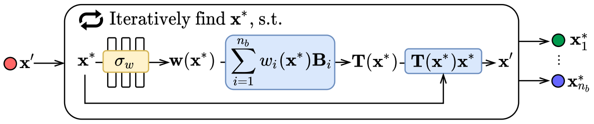

Iteratively find $\mathbf{x}^*$, s.t.
$\mathbf{x}'$
$\mathbf{x}^*$
σ_w
$\mathbf{w}(\mathbf{x}^*)$
$\sum_{i=1}^{n_b} w_i(\mathbf{x}^*)\mathbf{B}_i$
$\mathbf{T}(\mathbf{x}^*)$
$\mathbf{T}(\mathbf{x}^*)\mathbf{x}^*$
$\mathbf{x}'$
$\mathbf{x}_1^*$
$\mathbf{x}_{n_b}^*$

Fig. 3: **MLP-based Forward Skinning (SNARF).** Given a point in deformed space $\mathbf{x}'$, SNARF finds its canonical correspondences $\mathbf{x}^*$ that satisfy the forward skinning equation (5) via root finding. Multiple correspondences can be reliably found by initializing the root finding algorithm with multiple starting points derived from the bone transformations.

where $\mathbf{x}^*$ is the root as defined in Eq. (9)

The last term cannot be obtained using standard auto-differentiation because $\mathbf{x}^*$ is determined by the iterative correspondence search using σ_w. This iterative procedure is not trivially differentiable. To overcome this problem, implicit differentiation is used to derive the following analytical form of the last term:

$$\frac{\partial \mathbf{x}^*}{\partial \sigma_w} = -\left(\frac{\partial \mathbf{d}_{\sigma_w}(\mathbf{x}^*, \boldsymbol{B})}{\partial \mathbf{x}^*}\right)^{-1} \cdot \frac{\partial \mathbf{d}_{\sigma_w}(\mathbf{x}^*, \boldsymbol{B})}{\partial \sigma_w}. \tag{12}$$

Substituting Eq. (12) into Eq. (11) yields the gradient term $\frac{\partial \mathcal{L}}{\partial \sigma_w}$ which then allows skinning weights to be learned with standard back-propagation.

4 Fast Differentiable Forward Skinning

While the formulation mentioned above can articulate neural fields with good quality and generalization ability, the original SNARF algorithm is computationally expensive, which limits its wider application. As a reference, determining the correspondences of 200k points takes 800ms on an NVIDIA Quadro RTX 6000 GPU. In the following, we describe how Fast-SNARF overcomes this issue, reducing the computation time from 800ms to 5ms (Table 2).

4.1 Voxel-based Correspondence Search

The core of our fast method involves factoring out costly computations at each root finding iteration in SNARF, including MLP evaluations and LBS calculations, by putting these into a pre-computation stage as illustrated in Algorithm 1. Figure 5 graphically illustrates the cost of each key algorithm block in Fast-SNARF.

Voxel-based Skinning Field: The main speed bottleneck of SNARF lies in computing Eq. (7) at each iteration of Broyden's method. Computing Eq. (7) is time-consuming because it involves querying skinning weights, which are parameterized via an MLP in SNARF, and then computing LBS. We notice that the skinning weight field does not contain high-frequency details as illustrated in Fig. 7. Therefore, we re-parameterize the skinning weight field $\mathbf{w}$ with a low-resolution voxel grid $\{\mathbf{w}_v\}$ with skinning weights $\mathbf{w}_v$ defined for each grid point $\mathbf{x}_v$. The skinning weights of any, non-grid aligned point in space are then obtained via tri-linear interpolation. We find that a resolution of $64 \times 64 \times 16$ is sufficient to describe the skinning weights in all experiments. Note that we use lower resolution along the z-axis due to the "flatness" of the human body along this dimension in canonical space.

We choose a voxel representation over other neural field representations for computational efficiency. Querying a value from a voxel grid requires fewer read operations (8 for tri-linear interpolation) compared to multi-resolution hash tables [43] ($8 \times L$ for a L-level table) and tri-planes [8, 52] (4×3) and does not require running additional MLPs. While voxel-based representations are less memory efficient, this is not critical in our case since skinning weights are naturally smooth and can be well represented by a low-resolution grid.

Pre-computing LBS: Computing linear blend skinning (Eq. (14)) at each root finding iteration also impacts speed. We notice redundancy in the computation and hence propose a pre-computation scheme to improve efficiency. Similar to Eq. (14), the linearly blended skinning transformation of a canonical point with voxel-based skinning weights is given as

$$\begin{aligned}\mathbf{T}(\mathbf{x}) &= \sum_{i=1}^{n_\mathrm{b}} w_i(\mathbf{x}) \cdot \boldsymbol{B}_i \\ &= \sum_{i=1}^{n_\mathrm{b}} \mathrm{trilerp}(w_{v_0,i}, \ldots, w_{v_7,i}, \mathbf{x}) \cdot \boldsymbol{B}_i,\end{aligned} \tag{13}$$

where $v_0, \ldots v_7$ are the 8 neighbouring grid points of $\mathbf{x}$. Since tri-linear interpolation is a linear operation w.r.t. the values at grid points, the equation above can be rewritten as

$$\begin{aligned}\mathbf{T}(\mathbf{x}) &= \sum_{i=1}^{n_\mathrm{b}} \mathbf{A}(\mathbf{x}) \cdot [w_{v_0,i} \ldots w_{v_7,i}]^T \cdot \boldsymbol{B}_i \\ &= \mathbf{A}(\mathbf{x}) \cdot \sum_{i=1}^{n_\mathrm{b}} [w_{v_0,i} \ldots w_{v_7,i}]^T \cdot \boldsymbol{B}_i \\ &= \mathbf{A}(\mathbf{x}) \cdot [\sum_{i=1}^{n_\mathrm{b}} w_{v_0,i}\boldsymbol{B}_i \ldots \sum_{i=1}^{n_\mathrm{b}} w_{v_7,i}\boldsymbol{B}_i]^T \\ &= \mathbf{A}(\mathbf{x}) \cdot [\mathbf{T}_{v_0} \ldots \mathbf{T}_{v_7}]^T\end{aligned} \tag{14}$$

where $\mathbf{T}_v$ are the linearly blended transformations of neighbouring grid points. Our explicit voxel-based skinning weights representation $\{\mathbf{w}_v\}$ allows us to compute the transformations for all grid points $\{\mathbf{T}_v\}$ given current body poses:

$$\mathbf{T}_v = \sum_{i=1}^{n_\mathrm{b}} w_{v,i} \cdot \boldsymbol{B}_i. \tag{15}$$

Then, during root finding, the required transformation at any canonical point $\mathbf{T}(\mathbf{x})$ can be determined by tri-linearly interpolating neighbouring transformations in $\{\mathbf{T}_v\}$. Thus, LBS only needs to be run for a small set of grid points instead of all query points in the root-finding procedure.

Custom CUDA Kernel: Broyden's method is iterative and involves many small operations that have to be computed per query point, such as arithmetic operations on small matrices and reading values from the voxel grid. We note that these operations can be computed in an independent manner. This motivates us to implement this module with a custom CUDA kernel instead of using native functions in

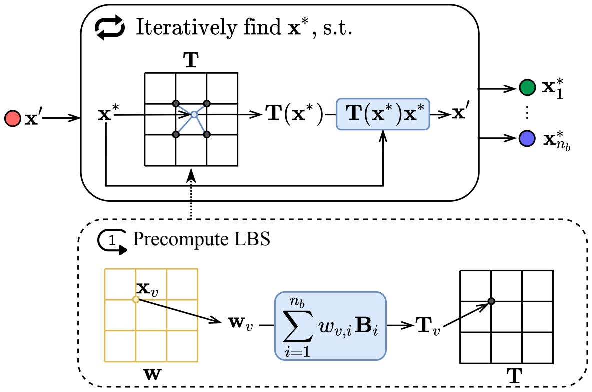


Fig. 4: **Voxel-based Forward Skinning (Fast-SNARF).** In comparison with SNARF (cf. Fig. 3), Fast-SNARF uses a voxel-based representation to speed up the iterative correspondence search. The skinning weight field is represented as a voxel grid. For each pose, we first pre-compute LBS for each grid point, yielding a transformation field. For each query deformed point $\mathbf{x}'$, Fast-SNARF finds its canonical correspondences $\mathbf{x}^*$ which satisfy $\mathbf{T}(\mathbf{x}^*) \cdot \mathbf{x}^* = \mathbf{x}'$.

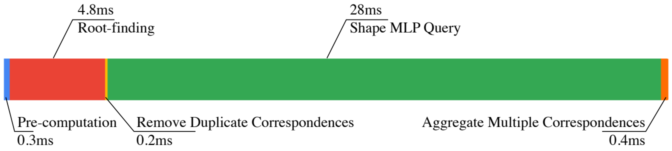


Fig. 5: **Runtime Profile.** We show the computation time of major algorithm blocks in Fast-SNARF.

Algorithm 1 Correspondence Search

```
Inputs:
$\{(\mathbf{x}', \mathbf{x}^0, \tilde{\mathbf{J}}^0)\}$ query points and initialization
$\boldsymbol{B}$ bone transformations
$\mathbf{w}_{\sigma_w}$ skinning weights MLP

Variant 1: MLP-based Search (SNARF)
for $\mathbf{x}', \mathbf{x}^0, \tilde{\mathbf{J}}^0 \in \{(\mathbf{x}', \mathbf{x}^0, \tilde{\mathbf{J}}^0)\}$ in parallel do
    for $k \leftarrow 0, n$ do
        $\underline{w_1, ..., w_{n_b} \leftarrow \mathbf{w}_{\sigma_w}(\mathbf{x}_j^k)}$          costly operations
        $\underline{\mathbf{T} \leftarrow \sum_{i=1}^{n_b} w_i(\mathbf{x}^k) \cdot \boldsymbol{B}_i}$          inside root finding
        $\mathbf{x}^{k+1}, \tilde{\mathbf{J}}^{k+1} \leftarrow \text{broyden}(\mathbf{x}^k, \tilde{\mathbf{J}}^k, \mathbf{T}, \mathbf{x}')$          ▷ Eq. (6)
    end for
end for
return: $\{\mathbf{x}^n\}$

Variant 2: Voxel-based Search (Fast-SNARF)
for each $\mathbf{x}_v \in \{\mathbf{x}_v\}$ in parallel do
    $\underline{w_1, ..., w_{n_b} \leftarrow \mathbf{w}_{\sigma_w}(\mathbf{x}_v)}$          pre-computation
    $\underline{\mathbf{T}_v \leftarrow \sum_{i=1}^{n_b} w_i \cdot \boldsymbol{B}_i}$
end for
for $\mathbf{x}', \mathbf{x}^0, \tilde{\mathbf{J}}^0 \in \{(\mathbf{x}', \mathbf{x}^0, \tilde{\mathbf{J}}^0)\}$ in parallel do
    for $k \leftarrow 0, n$ do
        $\underline{\mathbf{T} \leftarrow \text{trilerp}(\mathbf{x}^k, \{\mathbf{T}_v\})}$          lightweight operation
        $\mathbf{x}^{k+1}, \tilde{\mathbf{J}}^{k+1} \leftarrow \text{broyden}(\mathbf{x}^k, \tilde{\mathbf{J}}^k, \mathbf{T}, \mathbf{x}')$          ▷ Eq. (6)
    end for
end for
return: $\{\mathbf{x}^n\}$
```

standard deep learning frameworks. The handwritten kernel, parallelized over query points, fuses the entire method into a single kernel that keeps working variables in registers, avoiding unnecessary time and memory costs from launching native kernels and synchronizing intermediate results. The input to our CUDA kernel for iterative root finding is the pre-computed voxel grid of transformations $\{\mathbf{T}_v\}$, the bone transformations $\mathbf{B}$ as well as query points $\mathbf{x}'$. The kernel first computes the multiple initialization states (Eq. (8)). Then, at each root finding iteration, the kernel tri-linearly interpolates $\{\mathbf{T}_v\}$ and transforms the points (Eq. (5)), and applies Broyden's update (Eq. (7)). After each iteration k, we filter diverged and converged points $\mathbf{x}^k$ by checking whether $\left\|\mathbf{d}_{\sigma_w}(\mathbf{x}^k, \boldsymbol{B}) - \mathbf{x}'\right\|_2$ is larger than the divergence threshold or smaller than the convergence threshold, further reducing the number of required computations.

Remove Duplicate Correspondences: A further important speed optimization pertains to the treatment of multiple correspondences found by the root finding algorithm. The set of valid correspondences contains duplicates because different initial states can converge to the same solution. To avoid unnecessary evaluation of the canonical neural fields for these duplicates, we detect duplicate solutions by their relative distances in canonical space and discard them.

4.2 Skinning Weights Optimization

Analogous to SNARF, in theory, Fast-SNARF supports learning skinning weights with the analytical gradients in Eq. (12). However, there are two practical challenges.

Approximated Gradient: A first problem lies in that Eq. (12) involves computing derivatives and the matrix inversion $\left(\frac{\partial \mathbf{d}_{\sigma_w}(\mathbf{x}^*, \boldsymbol{B})}{\partial \mathbf{x}^*}\right)^{-1}$, which is time-consuming, impeding our goal of fast training. To address this, we note that this term is identical to the inverse of the Jacobian $\mathbf{J}$ in the last iteration of root finding (Eq. (7)):

$$\left(\frac{\partial \mathbf{d}_{\sigma_w}(\mathbf{x}^*, \boldsymbol{B})}{\partial \mathbf{x}^*}\right)^{-1} = \underbrace{\left(\frac{\partial \mathbf{d}_{\sigma_w}(\mathbf{x}^*, \boldsymbol{B}) - \mathbf{x}'}{\partial \mathbf{x}^*}\right)^{-1}}_{\mathbf{J}} \tag{16}$$

because the deformed point $\mathbf{x}'$ is a given input and is independent of the canonical correspondence $\mathbf{x}^*$. The inverse of the Jacobian $\mathbf{J}$ is approximated in Broyden's method as $\tilde{\mathbf{J}}$. Thus, we use $\tilde{\mathbf{J}}$ directly:

$$\frac{\partial \mathbf{x}^*}{\partial \sigma_w} = -\tilde{\mathbf{J}} \cdot \frac{\partial \mathbf{d}_{\sigma_w}(\mathbf{x}^*, \boldsymbol{B})}{\partial \sigma_w}. \tag{17}$$

Distilling Smooth Skinning Fields: A second problem is that the voxel-based parameterization does not have the global smoothness bias of MLPs, thus optimizing voxels directly would result in a noisy skinning weight field. To obtain smooth skinning weights while using voxel-based correspondence search, a common approach is to apply a total variational regularizer. However, we experimentally found that this regularization does not lead to the desired smoothness of the skinning weights and negatively affects the accuracy of the generated shapes. We thus propose a new approach by using an MLP to parameterize the skinning weight field during training but continuously distill the MLP to a voxel-based skinning weight field at each training iteration. The skinning weight field is thus smooth by design due to the intermediate use of an MLP. At each training iteration, we compute the skinning weights voxel grid on the fly by evaluating the MLP at grid points $\{\mathbf{x}_v\}$, and then use our fast voxel-based correspondence search. In this scheme the parameters of the MLP are optimized during training, not the voxels directly which are only used to store the weights. The conversion from MLP to voxels does introduce additional computation during training, but the overhead is minor since the voxel grid is low resolution. The inference speed is not influenced at all because the MLP is used during training only. This yields on-par accuracy with SNARF as we inherit the inductive smoothness bias of the MLP-based skinning weight model.

5 Learning Human Avatars from 3D Scans

We can use our articulation module to learn animatable avatars from 3D scans. Given a set of 3D meshes in various body poses, our method learns the human shape in canonical space as an occupancy field alongside the canonical skinning weight field which is needed for animation. We use the same training losses as SNARF [12] (see Section 3).

5.1 Minimally Clothed Humans

We first evaluate the speed and accuracy of our method and baselines on minimally clothed humans.

5.1.1 Dataset

We follow the same evaluation protocol as NASA [16] and SNARF [12]. More specifically, we use the DFaust [5] subset of AMASS [38] for training and evaluating our model on SMPL [35] meshes of people in minimal clothing. This dataset covers 10 subjects of varying body shapes. For each subject, we use 10 sequences, from which we randomly select one sequence for validation, using the rest for training. For each frame in a sequence, 20K points are sampled, among which, half are sampled uniformly in space and half are sampled in near-surface regions by first applying Poisson disk sampling on the mesh surface, followed by adding isotropic Gaussian noise with $\sigma = 0.01$ to the sampled point locations. In addition to the "within distribution" evaluation on DFaust, we test "out of distribution" performance on another subset of AMASS, namely PosePrior [1]. This subset contains challenging, extreme poses, not present in DFaust. This "out of distribution" setting simulates the real application scenario, where the reconstructed avatars are driven with arbitrary poses from MoCap systems or user control to generate new animations. The poses in AMASS are obtained by fitting the SMPL model to a sparse set of keypoints tracked by a MoCap system [36].

5.1.2 Baselines

We consider SNARF as our main baseline. In addition, we consider the following additional baselines. For SNARF, "Back-LBS" and "Pose-ONet" we use the same training losses and hyperparameters as in Fast-SNARF.

Pose-Conditioned Occupancy Networks (Pose-ONet): This baseline extends Occupancy Networks [39] by directly concatenating the pose input to the occupancy network.

Backward Skinning (Back-LBS): This baseline implements the concept of backward skinning similar to [27]. A network takes a deformed point and pose condition as input and outputs the skinning weights of the deformed point. The deformed point is then warped back to canonical space via LBS and the canonical correspondence is fed into the canonical shape network to query occupancy.

NASA: NASA [16] models articulated human bodies as a composition of multiple parts, each of which transforms rigidly and deforms according to the pose. Note that in contrast to us, NASA requires ground-truth skinning weights for surface points as supervision. We use the official NASA implementation provided by the authors.

5.1.3 Results and Discussion

Within Distribution Accuracy: Overall, all methods perform well in this relatively simple setting, as shown in Table 1. Our method achieves on-par or better accuracy compared to SNARF and provides an improvement over other baselines. Our method produces bodies with smooth surfaces and correct poses as shown in Fig. 6. In contrast, NASA suffers from discontinuous artifacts near joints. Back-LBS and Pose-ONet suffer from missing body parts.

Out of Distribution (OOD) Accuracy: In this setting, we test the trained models on a different dataset, PosePrior [1], to assess the performance in more realistic settings, where poses can be far from those in the training set. Unseen poses cause drastic performance degradation to the baseline methods as shown in Table 1. In contrast, similar to SNARF, our method degrades gracefully despite test poses being drastically different from training poses and very challenging. As can be seen in Fig. 6, our method generates natural shapes for the given poses while NASA fails to generate correctives at bone intersections for unseen poses, leading to noticeable artifacts. Pose-ONet fails to generate meaningful shapes and Back-LBS produces distorted bodies due to incorrect skinning weights.

Speed Comparison: We report the training and inference speed of all methods on a single NVIDIA Quadro RTX 6000 GPU. In this setting, with MLP-based canonical shape, Fast-SNARF can be trained within 25 minutes and produces accurate shapes in any pose. Baseline methods that reach similar speed, i.e. Pose-ONet, and Back-LBS, do not produce satisfactory results (see Fig. 6). Compared to the original SNARF, our improvements, detailed in Section 4, lead to a speed-up of $150\times$ for the articulation module without loss of accuracy, as shown in Table 1. Fast-SNARF also dramatically boosts the training speed (25 minutes vs. 8 hours). Compared to NASA, Fast-SNARF evaluates the canonical shape MLP only for true correspondences, while NASA always

	Within Distribution		Out of Distribution		Inference Speed			Training Time
	IoU bbox	IoU surf	IoU bbox	IoU surf	Articulation	Shape	Total	
Pose-ONet*	79.34%	58.61%	49.21%	28.69%	0ms	28.95ms	29.88ms	16min
Backward-LBS*	81.68%	87.44%	66.93%	68.93%	12.39ms	27.67ms	40.60ms	31min
NASA	96.14%	86.98%	83.16%	60.21%	-	-	582ms	4h
SNARF	97.31%	90.38%	93.97%	80.65%	806.67ms	186.82ms	994.01ms	8h
Fast-SNARF	**97.41%**	**90.52%**	**94.20%**	**81.25%**	5.27ms	27.78ms	34.70ms	25min

TABLE 1: **Quantitative Results on Minimally Clothed Humans.** The mean IoU of uniformly sampled points in space (IoU bbox) and points near the surface (IoU surface), as well as the inference and training time are reported. Our method achieves similar accuracy as SNARF (previous state-of-the-art) while being much faster. Our method outperforms all other baselines in terms of accuracy. Improvements are more pronounced for points near the surface, and for poses outside the training distribution. Also our method is faster than all baselines except Pose-ONet. Note that Pose-ONet and Backward-LBS (above the separation line, marked with *) produce distorted shapes, as shown in Fig. 6.

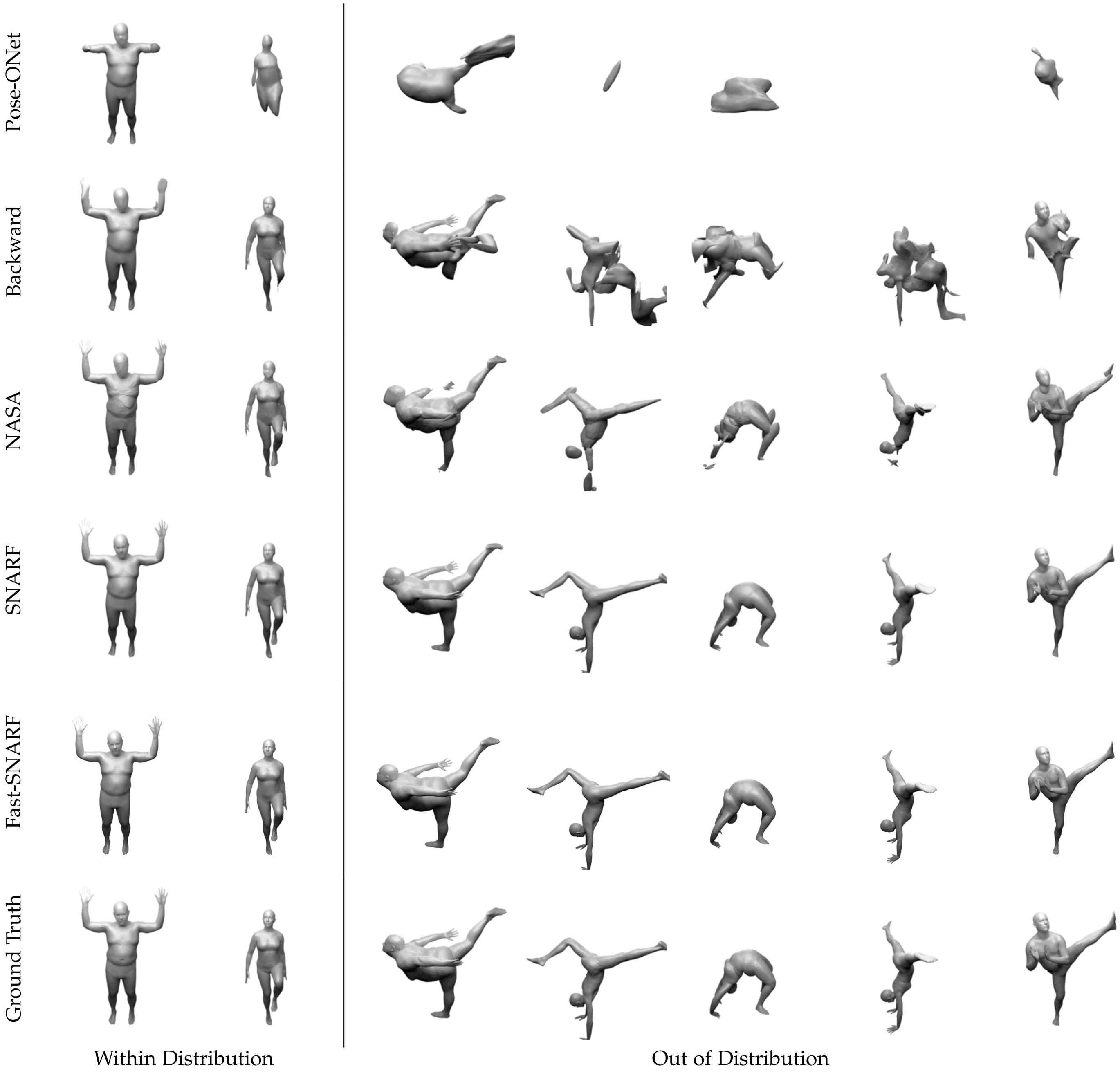


Fig. 6: **Qualitative Results on Minimally Clothed Humans.** Our method and SNARF produce results similar to the ground-truth with correct pose and plausible local details, both for poses within the training distribution and more extreme (OOD) poses. In contrast, the baseline methods suffer from various artifacts including incorrect poses (Pose-ONet), degenerate shapes (Pose-ONet and Backward), and discontinuities near joints (NASA), which become more severe for unseen poses.

generates many candidate correspondences, one for each bone, and needs to evaluate the canonical shape MLP for all candidates, leading to slow inference (582ms vs. 35ms) and training (4 hours vs. 25 minutes).

Ablation - MLP Distillation: SNARF optimizes an MLP-based skinning field, resulting in smooth skinning weights but slow training and inference. In Fast-SNARF, we adopt an MLP distillation strategy: we optimize an MLP-based skinning weight field for smoothness, but convert it on the fly to a low-resolution voxel grid at each training iteration, to enable voxel-based correspondence search. In this way, Fast-SNARF learns a similarly smooth skinning field as shown in Fig. 7, yet is much faster than SNARF (see Table 2).

We also compare this MLP distillation strategy with a naive strategy in which we directly optimize the skinning weights at each grid point with an additional total variation loss on the skinning weights voxel grid. As shown in Table 2, directly optimizing skinning weights voxel (w/o MLP distillation) leads to inferior results. This accuracy degradation is due to noisy skinning weights as shown in Fig. 7. The noisy skinning weights not only affect the accuracy of posed shapes but also result in artifacts on the canonical shapes, e.g. on the right hand in Fig. 7. In contrast, our strategy distills smooth skinning weights voxels from the MLP while introducing only a slight overhead during training (25 minutes vs. 23 minutes). Note that all three models learn incomplete skinning weights for hands. Despite that the skeleton includes two joints for each hand, namely the wrist (green) and palm (white), these models assign each hand to a single joint, either palm or wrist, arbitrarily. This is due to the limited hand movement in the training data.

Ablation - Voxel Grid Resolution: We study the effect of different resolutions of the skinning weight voxel grid. The results are shown in Table 2. In general, higher resolutions lead to higher accuracy but longer training and inference time. A resolution of $32 \times 32 \times 8$ or $64 \times 64 \times 16$ yields a good balance between accuracy and speed. A grid of lower resolution $16 \times 16 \times 4$ cannot fully represent the skinning weight field and leads to a noticeable accuracy degradation (by 2.8%). On the other hand, further increasing the resolution to $128 \times 128 \times 32$ produces diminishing returns, i.e. only 0.3% IoU improvement, because the skinning weight field is naturally smooth and does not contain high-frequency details. Also, higher resolution significantly slows down the training and inference speed by more than 2 times because 1) more points need to be evaluated when converting the MLP to voxels during training and 2) the high-resolution voxel grid does no longer fit into the GPU's shared memory and impacts read speeds significantly.

Configurations	Accuracy	Inference	Training
Baseline SNARF	80.7%	807ms + 187ms	8h
+ Voxel-based search	-	61ms + 187ms	-
+ Pre-compute LBS	-	40ms + 187ms	-
+ CUDA kernel	-	5.3ms + 187ms	-
+ Filter corres.	-	5.3ms + 28ms	-
Fast-SNARF	81.2%	5.3ms + 28ms	25 min
w/o MLP distillation	78.2%	5.3ms + 28ms	23 min
$16 \times 16 \times 4$	78.3%	3.6ms + 28ms	23min
$32 \times 32 \times 8$	81.1%	4.6ms + 28ms	24min
$64 \times 64 \times 16$	81.2%	5.3ms + 28ms	25min
$128 \times 128 \times 32$	81.5%	16ms + 28ms	52min

TABLE 2: **Quantitative Ablation Study.** We report accuracy (the mean IoU of points near the surface in out of distribution setting), inference speed (articulation speed + shape query speed) and training time of several ablative baselines.

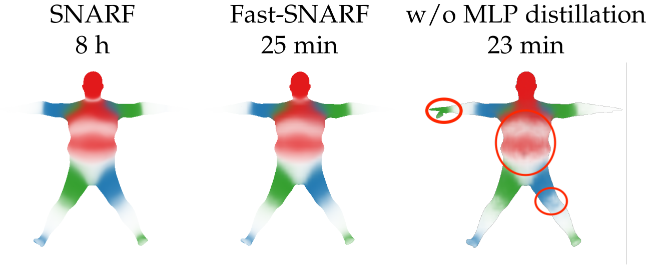


Fig. 7: **Skinning Weight Learning Strategies.** We show skinning weights learned with three different strategies as well as the corresponding training times. See text.

5.2 Clothed Avatar from Scans

Dataset: We use the registered meshes from CAPE [37] and their SMPL parameters to train our model. We use 8 subjects with different clothing types for evaluation. We train a model for each subject and clothing condition.

Baselines: Clothed humans are more challenging to model than minimally clothed humans due to the clothing details and non-linear deformations. Since most baselines from Section 5.1 already suffer from implausible shapes and artifacts, we exclude them in this evaluation. Instead, we keep SNARF as our major baseline, and also include a new baseline denoted as "SMPL NN". This baseline assumes that a skinned base mesh is given, such as SMPL [35]. Given a pose, such a method first deforms the SMPL model to the target pose using mesh-based LBS. Then for each query point in deformed space, its corresponding skinning weights are defined as the skinning weights of its nearest vertex on the deformed SMPL mesh. Finally, with the skinning weights, the query point can be transformed back to the canonical space base on inverse LBS.

Results: The results are shown in Fig. 8. Our method can generate realistic clothed humans in various poses including details on the face and clothing (e.g. the collar on the left sample). The clothing also deforms naturally with the body poses (e.g. the collar on the left sample and the lapel on the right sample). While SNARF produces results of similar quality, training our method only requires a fraction of SNARF's training time (80 minutes vs. 20 hours). Compared with the SMPL NN baseline, our results contain much more detail because our method derives accurate correspondences between the deformed space and canonical space. SMPL NN suffers from overly smooth shapes due to inaccurate correspondences when the actual shape and the skinned base mesh do not match well, e.g. around the lapel.

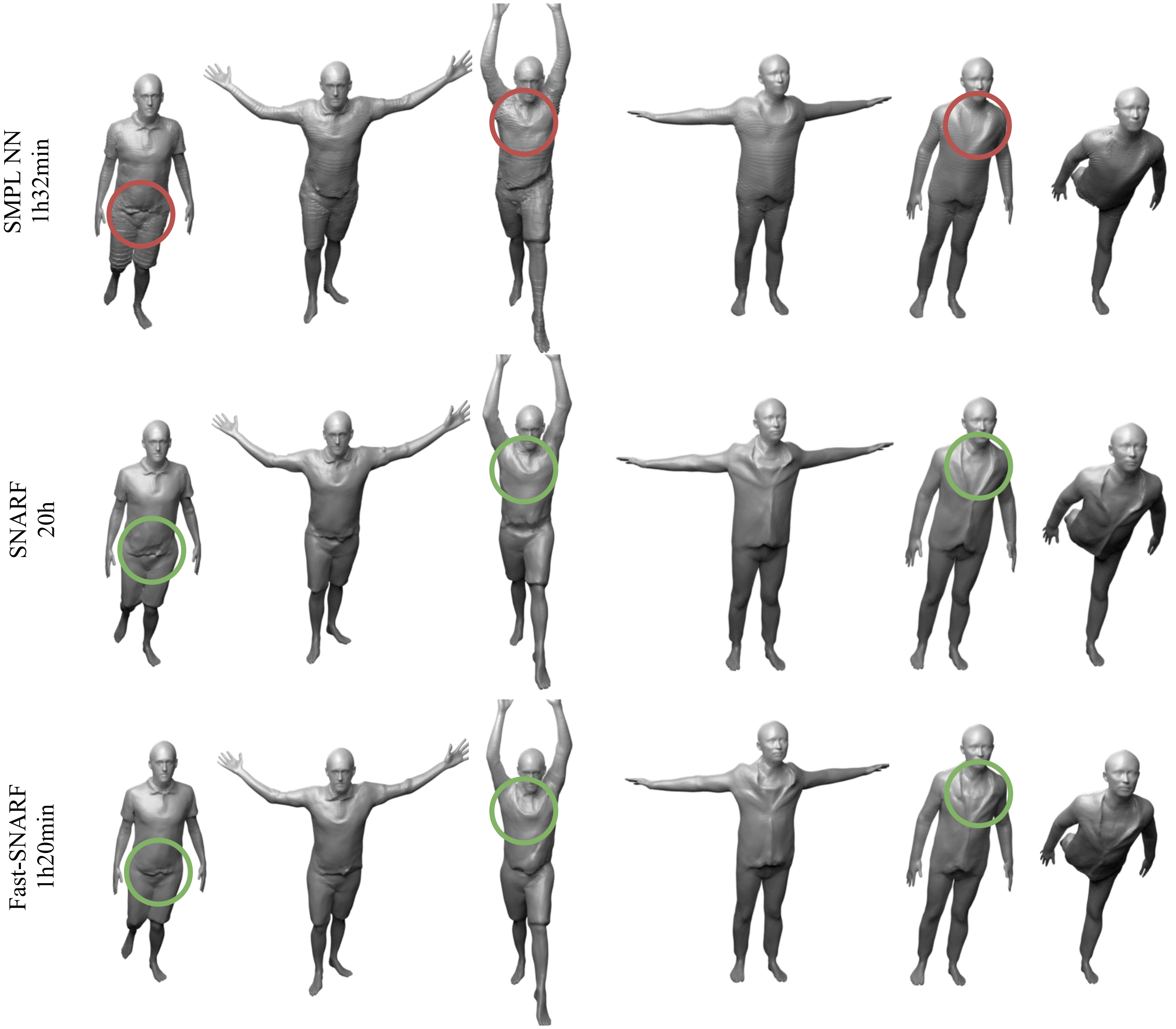


Fig. 8: **Qualitative Results on Clothed Humans [37]**. Our method and SNARF both learn realistic clothing shape and deformations. In contrast, the baseline method, using a skinned base mesh, produces fewer details due to the inaccurate deformation when the base mesh mismatches the actual shape (highlighted in red circles).

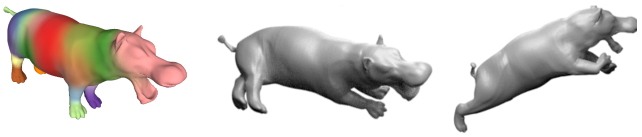


Fig. 9: **Qualitative Results on Animal.** Our method learns meaningful skinning weights and shapes for an animal (left), enabling realistic animation in unseen poses (right).

6 Learning 3D Animals

Our method also supports other articulated objects beyond humans. Here, we apply our method to a quadruped. The training data contains static meshes obtained by randomly posing a parametric animal model, SMAL [74]. Although the animal has more degrees of freedom than a human (33 joints vs. 24 joints), our method successfully learns meaningful skinning weights and shapes, and generates realistic animation in unseen poses, as shown in Fig. 9.

7 Learning Human Avatars from Images

Compared to the original SNARF, our new articulation module is even more versatile. Fast-SNARF can support other neural fields in the same manner that SNARF and its variants have been utilized. In addition, our voxel-based skinning weights representation allows skinning weights to be explicitly defined or initialized. This functionality is particularly useful in scenarios where observations are insufficient to determine skinning weights reliably. Furthermore, the speed of Fast-SNARF allows integration with volumetric renderers, enabling new applications that involve image observations. Here, we demonstrate an example in which we learn textured animatable avatars from monocular videos by taking these advantages. This example has been covered as an ablative baseline in our concurrent work InstantAvatar [29].

Canonical Representation: To model appearance, we use

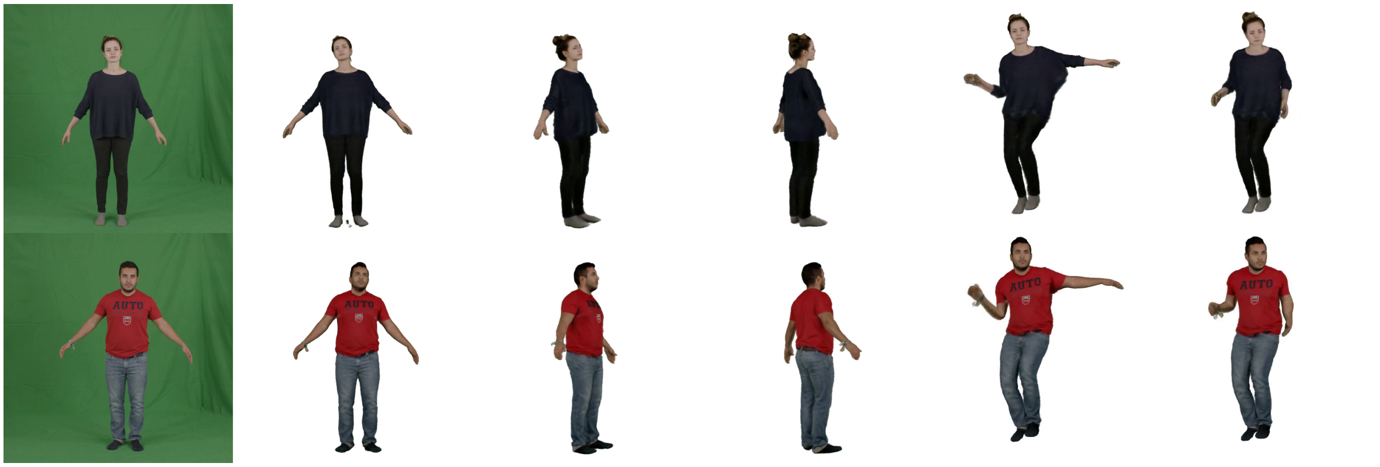


Fig. 10: **Qualitative Results on PeopleSnapshot [2].** Trained with a monocular video of a moving person and accurate poses, our method can generate images of the person from novel viewpoints (col. 3-4) and in novel body poses (col. 5-6).

neural radiance fields (NeRFs) [42] as our canonical representation. NeRF predicts the opacity σ and color c for any 3D point $\mathbf{x}$ in canonical space

$$\mathbf{f}_{\sigma_f} : \mathbb{R}^3 \to \mathbb{R}^+, \mathbb{R}^3 \tag{18}$$

$$\mathbf{x} \mapsto \sigma, c. \tag{19}$$

We note that our deformation module runs much faster than a standard MLP (5ms vs. 27ms). Therefore, relying on the original NeRF formulation would introduce significant overhead due to its reliance on MLPs to represent the scene. To circumvent this bottleneck we use instant-NGP [43] as the backbone. Instant-NGP stores explicit multi-layer spatial features as a hash table and uses only a shallow MLP to fuse features of different coarseness. Hence the method is much faster than deep MLP-based methods. Note that we have to omit pose-dependent local deformations because instant-NGP does not allow additional pose conditioning, unlike MLPs.

Volumetric Rendering of Articulated Shapes: For rendering, we follow NeRF [42] and first compute the opacity and color of multiple sample points along a ray cast from a pixel and then integrate the opacity and color using volumetric rendering. Rendering articulated NeRFs is similar. The main difference is that the sample points are first mapped to the canonical space via our articulation module before computing the opacity and color.

Losses: We train our model by minimizing the robust Huber loss [25] ρ between the predicted color of the pixels C and the corresponding ground-truth color C_{gt}:

$$\mathcal{L}_{\text{rgb}} = \rho(\|C - C_{gt}\|). \tag{20}$$

In addition, we use the human mask (which is obtained via chroma key) and apply a loss on the rendered 2D alpha values, in order to reduce floating noise artifacts in the empty space which are typical for NeRF representations:

$$\mathcal{L}_{\text{alpha}} = \|\alpha - \alpha_{gt}\|_1. \tag{21}$$

Note that in this setting we use preset skinning weights derived from the SMPL [35] model. This is because we use the PeopleSnapshot [2] dataset which does not contain sufficient pose variations to learn skinning weights.

Dataset: We use the PeopleSnapshot dataset proposed by Alldieck et al. [2], which contains videos of humans rotating in front of a camera with limited pose variation.

Results: Our results are shown in Fig. 10. Our method faithfully reconstructs the 3D appearance of both humans including very fine details such as the logo on the T-shirt, wrinkles on the pants, and facial features. With the reconstructed model, we can then synthesize novel views of the virtual human. In addition, we can generate images in novel poses. The images in novel poses are realistic except that the hand regions are noisy. This is because the palms are mostly invisible in the training videos. It is a limitation that our method cannot infer unobserved regions.

Due to our fast articulation module Fast-SNARF and the backbone instant-NGP [43], the reconstruction process is fast. With accurate poses as input, the training process takes only 3 minutes to reach satisfactory results. Rendering an image in any pose from any view takes on average 1 s.

8 Conclusion

We propose Fast-SNARF, a fast, robust, and universal articulation module for neural field representations. Fast-SNARF is built upon the idea of differentiable forward skinning from SNARF [12], but is orders of magnitude faster than SNARF thanks to a series of algorithmic and implementation improvements. These include voxel-based correspondence search, LBS pre-computation, a custom CUDA kernel implementation for root finding, duplicate correspondences removal, approximated implicit gradients, and online MLP-to-voxel conversion. The resulting algorithm can find correspondences as accurately as SNARF while being 150× faster. This leads to significant speed-up in various real-world applications of forward skinning algorithms. Using Fast-SNARF we are able to learn animatable human avatars from scans 15× faster than SNARF, and in contrast to SNARF, the speed bottleneck is now the canonical shape query instead of the articulation module. We believe Fast-SNARF's speed and accuracy will open new applications and accelerate research on non-rigid 3D reconstruction.

Acknowledgements: Xu Chen was supported by the Max Planck ETH Center for Learning Systems. Andreas Geiger was supported by the DFG EXC number 2064/1 - project number 390727645. We thank Yao Feng, Jinlong Yang, Yuliang Xiu and Alex Zicong Fan for their feedback.

Disclosure: MJB has received research gift funds from Adobe, Intel, Nvidia, Meta/Facebook, and Amazon. MJB has financial interests in Amazon, Datagen Technologies, and Meshcapade GmbH. MJB's research was performed solely at, and funded solely by, the Max Planck.

REFERENCES

[1] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[2] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3D people models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] Alexander W. Bergman, Petr Kellnhofer, Wang Yifan, Eric R. Chan, David B. Lindell, and Gordon Wetzstein. Generative neural articulated radiance fields. *Arxiv*, 2022.

[5] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Charles G Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 1965.

[7] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[8] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[9] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[10] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, Xu Jia, and Huchuan Lu. Animatable neural radiance fields from monocular rgb videos. *arXiv.org*, 2021.

[11] Xu Chen, Tianjian Jiang, Jie Song, Jinlong Yang, Michael J Black, Andreas Geiger, and Otmar Hilliges. gdna: Towards generative detailed neural avatars. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[12] Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. SNARF: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[13] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3D shape reconstruction and completion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[14] Enric Corona, Tomas Hodan, Minh Vo, Francesc Moreno-Noguer, Chris Sweeney, Richard Newcombe, and Lingni Ma. Lisa: Learning implicit shape and appearance of hands. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. SMPLicit: Topology-aware generative model for clothed people. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[16] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. In *European Conference on Computer Vision (ECCV)*, 2020.

[17] Zijian Dong, Chen Guo, Jie Song, Xu Chen, Andreas Geiger, and Otmar Hilliges. PINA: Learning a personalized implicit neural avatar from a single RGB-D video sequence. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[19] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proc. of the International Conf. on Machine learning (ICML)*, 2020.

[20] Tong He, John Collomosse, Hailin Jin, and Stefano Soatto. Geo-pifu: Geometry and pixel aligned implicit functions for single-view human reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[21] Tong He, Yuanlu Xu, Shunsuke Saito, Stefano Soatto, and Tony Tung. ARCH++: Animation-ready clothed human reconstruction revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[22] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[23] Fangzhou Hong, Mingyuan Zhang, Liang Pan, Zhongang Cai, Lei Yang, and Ziwei Liu. Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *ACM Trans. Gr.*, 2022.

[24] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[25] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[26] Doug L. James and Christopher D. Twigg. Skinning mesh animations. *ACM Trans. on Graphics*, 24(3):399, 2005.

[27] Timothy Jeruzalski, David IW Levin, Alec Jacobson, Paul Lalonde, Mohammad Norouzi, and Andrea Tagliasacchi. Nilbs: Neural inverse linear blend skinning. *arXiv.org*, 2004.05980, 2020.

[28] Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. Selfrecon: Self reconstruction your digital avatar from monocular video. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[29] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. Instantavatar: Learning avatars from monocular video in 60 seconds. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[30] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. Neuman: Neural human radiance field from a single video. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[31] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhoefer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[32] Siyou Lin, Hongwen Zhang, Zerong Zheng, Ruizhi Shao, and Yebin Liu. Learning implicit templates for point-based clothed human modeling. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[33] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[34] 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 Trans. on Graphics*, 2021.

[35] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. on Graphics*, 2015.

[36] Matthew M. Loper, Naureen Mahmood, and Michael J. Black. MoSh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 33(6):220:1–220:13, Nov. 2014.

[37] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. Learning to dress 3D people in generative clothing. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[38] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[39] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[40] Marko Mihajlovic, Shunsuke Saito, Aayush Bansal, Michael Zollhoefer, and Siyu Tang. COAP: Compositional articulated occupancy of people. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[41] Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. LEAP: Learning articulated occupancy of people. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[42] 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 *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[43] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. on Graphics*, 41, 2022.

[44] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3D representations without 3D supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[45] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Neural articulated radiance field. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[46] Atsuhiro Noguchi, Xiao Sun, Stephen Lin, and Tatsuya Harada. Unsupervised learning of efficient geometry-aware neural articulated representations. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[47] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[48] Pablo Palafox, Aljaž Božič, Justus Thies, Matthias Nießner, and Angela Dai. Neural parametric models for 3D deformable shapes. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[49] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[50] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Animatable neural radiance fields for human body modeling. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[51] 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 *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[52] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[53] Sida Peng, Shangzhan Zhang, Zhen Xu, Chen Geng, Boyi Jiang, Hujun Bao, and Xiaowei Zhou. Animatable neural implict surfaces for creating avatars from videos. *arXiv preprint arXiv:2203.08133*, 2022.

[54] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[55] Antonio Ricci. A constructive geometry for computer

graphics. *The Computer Journal*, 16(2), 1973.

[56] Malcolm Sabin. The use of potential surfaces for numerical geometry. *British Aircraft Corporation, Weybridge, UK, Technical Report No. VTO/MS*, 153, 1968.

[57] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[58] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3D human digitization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[59] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J. Black. SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[60] Sara Fridovich-Keil and Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[61] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[62] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[63] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[64] Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Neural-GIF: Neural generalized implicit functions for animating people in clothing. In *International Conference on Computer Vision (ICCV)*, October 2021.

[65] Shaofei Wang, Marko Mihajlovic, Qianli Ma, Andreas Geiger, and Siyu Tang. MetaAvatar: Learning animatable clothed human models from few depth images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[66] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdfs. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[67] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[68] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J Black. ICON: Implicit Clothed humans Obtained from Normals. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[69] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *ICCV*, 2019.

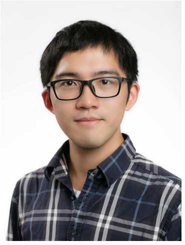
[70] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

[72] Jianfeng Zhang, Zihang Jiang, Dingdong Yang, Hongyi Xu, Yichun Shi, Guoxian Song, Zhongcong Xu, Xinchao Wang, and Jiashi Feng. Avatargen: A 3d generative model for animatable human avatars. *Arxiv*, 2022.

[73] Yufeng Zheng, Victoria Fernández Abrevaya, Xu Chen, Marcel C Bühler, Michael J Black, and Otmar Hilliges. IMAvatar: Implicit morphable head avatars from videos. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

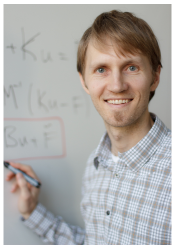
[74] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.



Xu Chen is currently a Ph.D. student at ETH Zurich and Max Planck Institute for Intelligent Systems, supervised by Prof. Otmar Hilliges, Prof. Andreas Geiger, and Prof. Michael J. Black. His main research interest lies in computer vision and graphics, especially in capturing and modeling 3D humans. He obtained a master's degree in Robotics, Systems, and Control at ETH Zurich in 2018 and a bachelor's degree in Electrical Engineering and Information Technology at the University of Stuttgart in 2015.

Tianjian Jiang is currently a research assistant at the Advanced Interactive Technology lab at ETH Zurich. He obtained his master's degree in Computer Science at ETH Zurich in 2021.

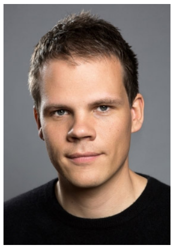
Dr. Jie Song is a postdoctoral researcher at ETH Zurich. He earned his Ph.D. in Computer Science from ETH Zurich in 2020, and his research interests encompass computer vision, machine learning, and human-computer interaction. Throughout his academic journey, he has collaborated with prominent research partners such as Google, Meta, and FIFA. His overarching goal is to create intelligent systems capable of naturally understanding and interacting with humans.



Dr. Max Rietmann is a computational scientist who did his PhD and Postdoctoral work at USI Lugano and ETH Zurich on high-performance computing and finite-element methods with applications in Geophysics. He currently works at NVIDIA with a focus on accelerating scientific and industrial applications using GPUs with a focus on helping users get the most out of their hardware through end-to-end workflow analysis and optimization.

Andreas Geiger is a professor at the University of Tübingen. Prior to this, he was a visiting professor at ETH Zürich and a group leader at the Max Planck Institute for Intelligent Systems. He studied at KIT, EPFL and MIT, and received his PhD degree in 2013 from the Karlsruhe Institute of Technology (KIT). He is an ELLIS fellow and coordinates the ELLIS PhD and PostDoc program. His research interests are at the intersection of computer vision, machine learning and robotics, with a particular focus on 3D scene perception, deep representation learning, generative models and reconstruction of 3D geometry and materials.

Michael Black received his B.Sc. from the University of British Columbia (1985), his M.S. from Stanford (1989), and his Ph.D. from Yale University (1992). He has held positions at the University of Toronto, Xerox PARC, and Brown University. He is an Honorarprofessor at the University of Tuebingen and one of the founding directors at the Max Planck Institute for Intelligent Systems in Tübingen, Germany, where he leads the Perceiving Systems department. He was a Distinguished Amazon Scholar (2017-2021). His work has won several awards including the IEEE Computer Society Outstanding Paper Award (1991), Honorable Mention for the Marr Prize (1999 and 2005), the 2022 and 2010 Koenderink Prize, the 2013 Helmholtz Prize, and the 2020 Longuet-Higgins Prize. He is a member of the German National Academy of Sciences Leopoldina and a foreign member of the Royal Swedish Academy of Sciences. In 2013 he co-founded Body Labs Inc., which was acquired by Amazon in 2017. He is a co-founder and Chief Scientist of Meshcapade.



Otmar Hilliges is a Full Professor of computer science at ETH Zurich, where he leads the AIT lab and serves as head of the institute for intelligent interactive systems. His research lies at the intersection of machine learning, computer vision and human computer interaction (HCI). He focuses on machine perception of human activity. Specifically, Hilliges is interested in spatio-temporal understanding of how humans move in and interact with the physical world. He develops algorithms, methods and representations for human- and interaction-centric understanding of the world from videos, images and other sensor data. Prior to joining ETH, he was a Researcher at Microsoft Research Cambridge (2012-2013). His Diploma (equiv. MSc) in Computer Science is from Technische Universität München, Germany (2004) and his PhD in Computer Science from LMU München, Germany (2009). He spent two years as a postdoc at Microsoft Research Cambridge (2010-2012). He has published more than 100 peer-reviewed papers in the major venues on computer vision, HCI and computer graphics. 20+ patents have been filed in his name on a variety of subjects from surface reconstruction to AR/VR. Amongst other sources of funding, Otmar Hilliges is a recipient of the prestigious ERC starting grant and ERC consolidator grant.