LEAP: Learning Articulated Occupancy of People

Marko Mihajlovic[1], Yan Zhang[1], Michael J. Black[2], Siyu Tang[1]

[1] ETH Zürich, Switzerland

[2] Max Planck Institute for Intelligent Systems, Tübingen, Germany

neuralbodies.github.io/LEAP

Abstract

Substantial progress has been made on modeling rigid 3D objects using deep implicit representations. Yet, extending these methods to learn neural models of human shape is still in its infancy. Human bodies are complex and the key challenge is to learn a representation that generalizes such that it can express body shape deformations for unseen subjects in unseen, highly-articulated, poses. To address this challenge, we introduce LEAP (LEarning Articulated occupancy of People), a novel neural occupancy representation of the human body. Given a set of bone transformations (i.e. joint locations and rotations) and a query point in space, LEAP first maps the query point to a canonical space via learned linear blend skinning (LBS) functions and then efficiently queries the occupancy value via an occupancy network that models accurate identity- and pose-dependent deformations in the canonical space. Experiments show that our canonicalized occupancy estimation with the learned LBS functions greatly improves the generalization capability of the learned occupancy representation across various human shapes and poses, outperforming existing solutions in all settings.

1. Introduction

Parametric 3D human body models [40, 65] are often represented by polygonal meshes and have been widely used to estimate human pose and shape from images and videos [19, 31, 36], create training data for machine learning algorithms [24, 52] and synthesize realistic human bodies in 3D digital environments [72, 73]. However, the mesh-based representation often requires a fixed topology and lacks flexibility when combined with deep neural networks where back-propagation through the 3D geometry representation is desired.

Neural implicit representations [42, 48, 49] have been proposed recently to model rigid 3D objects. Such representations have several advantages. For instance, they
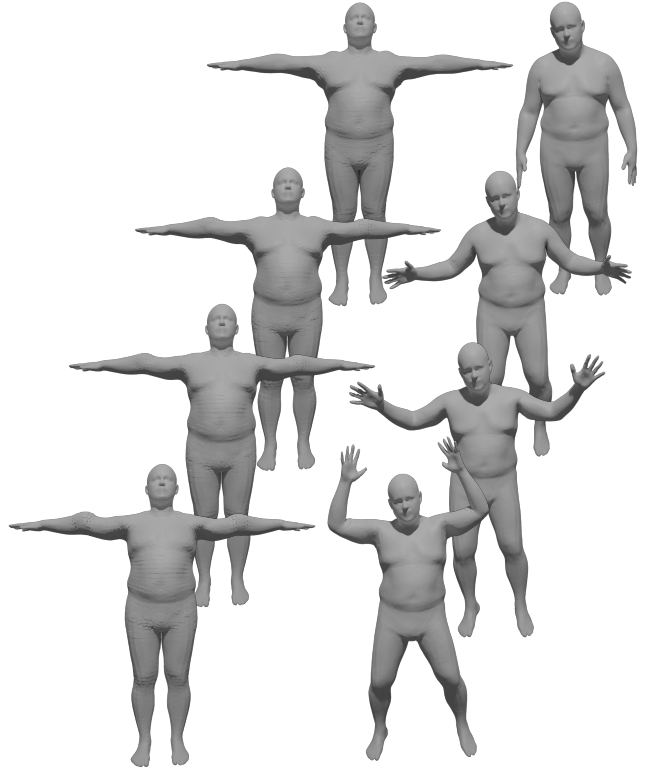

Figure 1. LEAP successfully represents unseen people in various challenging poses by learning the occupancy of people in a canonical space. Shape- and pose-dependent deformations are modeled through carefully designed neural network encoders. Pose-dependent deformations are best observed around the elbows in the canonical pose.

are continuous and do not require a fixed topology. The 3D geometry representation is differentiable, making interpenetration tests with the environment efficient. However, these methods perform well only on static scenes and objects, their generalization to deformable objects is limited, making them unsuitable for representing articulated 3D human bodies. One special case is NASA [15] which takes a set of bone transformations of a human body as in-

put and represents the 3D shape of the subject with neural occupancy networks. While demonstrating promising results, their occupancy representation only works for a single subject and does not generalize well across different body shapes. Therefore, the widespread use of their approach is limited due to the per-subject training.

In this work, we aim to learn articulated neural occupancy representations for various human body shapes and poses. We take inspiration from the traditional mesh-based parametric human body models [40, 65], where identity- and pose-dependent body deformations are modeled in a canonical space, and then Linear Blend Skinning (LBS) functions are applied to deform the body mesh from the canonical space to a posed space. Analogously, given a set of bone transformations that represent the joint locations and rotations of a human body in a posed space, we first map 3D query points from the posed space to the canonical space via learned inverse linear blend skinning (LBS) functions and then compute the occupancy values via an occupancy network that expresses differentiable 3D body deformations in the canonical space. We name it LEAP (LEarning Articulated occupancy of People).

The key idea of LEAP is to model accurate identity- and pose-dependent occupancy of human bodies in a canonical space (in analogy to the Shape Blend Shapes and Pose Blend Shapes in SMPL [40]). This circumvents the challenging tasks of learning occupancy functions in various posed spaces. Although conceptually simple, learning the canonicalized occupancy representation for a large variety of human shapes and poses is a highly non-trivial task.

The first challenge we encounter is that the conventional LBS weights are only defined on the body surface. In order to convert a query point from a posed space to the canonical space and perform the occupancy check, a valid skinning weight for *every* point in the posed spaces is required. To that end, we parameterize both forward and inverse LBS functions using neural networks and learn them from data. To account for the undefined skinning weights for the points that are not on the surface of a human body, we introduce a cycle-distance feature for every query point, which models the consistency between the forward and the inverse LBS operations on that point.

Second, a high fidelity human body model should be able to express accurate body shapes that vary across individuals and capture the subtle surface deformations when the body is posed differently. To that end, we propose novel encoding schemes for the bone transformations by exploiting prior knowledge about the kinematic structure and plausible shapes of a human body. Furthermore, inspired by the recent advances of learning pixel-aligned local features for 3D surface reconstruction [54, 55], for every query point, we use the learned LBS weights to construct a locally aware bone transformation encoding that captures accurate local shape deformations. As demonstrated in our experiments, the proposed local feature is an effective and expressive representation that captures detailed pose and shape-dependent deformations.

We demonstrate the efficacy of LEAP on the task of placing people in 3D scenes [72]. With the proposed occupancy representation, LEAP is able to effectively prevent person-person and person-scene interpenetration and outperforms the recent baseline [72].

Our *contributions* are summarized as follows: **1)** we introduce LEAP, a novel neural occupancy representation of people, which generalizes well across various body shapes and poses; **2)** we propose a canonicalized occupancy estimation framework and learn the forward and the inverse linear blend skinning weights for every point in space via deep neural networks; **3)** we conduct novel encoding schemes for the input bone transformations, which effectively model accurate identity- and pose-dependent shape deformations; **4)** experiments show that our method largely improves the generalization capability of the learned neural occupancy representation to unseen subjects and poses.

2. Related work

Articulated mesh representations. Traditional animatable characters are composed of a skeleton structure and a polygonal mesh that represents the surface/skin. This surface mesh is deformed by rigid part rotations and a skinning algorithm that produces smooth surface deformations [27]. A popular skinning algorithm is Linear Blend Skinning (LBS) which is simple and supported by most game engines. However, its flexibility is limited and it tends to produce unrealistic artifacts at joints [40, Fig. 2]. Thus, other alternatives have been proposed for more realistic deformations. They either improve the skinning algorithm [34, 38, 41, 64], learn body models from data [8, 9, 18, 22, 50], or develop more flexible models that learn additive vertex offsets (for identity, pose, and soft-tissue dynamics) in the canonical space [40, 47, 53].

While polygonal mesh representations offer several benefits such as convenient rendering and compatibility with animation pipelines, they are not well suited for inside/outside query tests or to detect collisions with other objects. A rich set of auxiliary data structures [32, 39, 57] have been proposed to accelerate search queries and facilitate these tasks. However, they need to index mesh triangles as a pre-processing step, which makes them less suitable for articulated meshes. Furthermore, the indexing step is inherently non-differentiable and its time complexity depends on the number of triangles [29], which further limits the applicability of the auxiliary data structures for learning pipelines that require differentiable inside/outside tests [23, 72, 73]. Contrary to these methods, LEAP supports straightforward and efficient differentiable

inside/outside tests without requiring auxiliary data structures.

Learning-based implicit representations. Unlike polygonal meshes, implicit representations support efficient and differentiable inside/outside queries. They are traditionally modeled either as linear combinations of analytic functions or as signed distance grids, which are flexible but memory expensive [58]. Even though the problem of the memory complexity for the grid-based methods is approached by [30, 46, 60, 70, 71], they have been outperformed by the recent learning-based continuous representations [2, 3, 10, 12, 21, 33, 42, 43, 45, 48, 49, 59, 66, 68]. Furthermore, to improve scalability and representation power, the idea of using local features has been explored in [7, 11, 44, 49, 54, 55, 66]. These learning-based approaches represent 3D geometry by using a neural network to predict either the closest distance from a query point to the surface or an occupancy value (i.e. inside or outside the 3D geometry). LEAP follows in the footsteps of these methods by representing a 3D surface as a neural network decision boundary while taking advantage of local features for improved representation power. However, unlike the aforementioned implicit representations that are designed for static shapes, LEAP is able to represent articulated objects.

Learning-based articulated representations. Recent work has also explored learning deformation fields for modeling articulated human bodies. LoopReg [4] has approached model-based registration by exploring the idea of mapping surface points to the canonical space and then using a distance transform of a mesh to project canonical points back to the posed space, while PTF [63] tackles this problem by learning a piecewise transformation field. ARCH [25] uses a deterministic inverse LBS that for a given query point retrieves the closest vertex and uses its associated skinning weights to transform the query point to the canonical space. NiLBS [28] proposes a neural inverse LBS network that requires per-subject training. NASA [15] is proposed to model articulated human body using a piecewise implicit representation. It takes as input a set of bone coordinate frames and represents the human shape with neural networks. Unlike these methods that are defined for human meshes with fixed-topology or require expensive per-subject training, LEAP uses deep neural networks to approximate the forward and the inverse LBS functions and generalizes well to unseen subjects. LEAP is closely related to NASA, with the following key differences **(i)** it shows improved representation power, outperforming NASA in all the settings; and **(ii)** LEAP is able to represent unseen people with a single neural network, eliminating the need for per-subject training. Concurrent with our work, SCANimate [56] uses a similar approach to learn subject-specific models of clothed people from raw scans.

Structure-aware representations. Prior work has explored pictorial structure [17, 67] and graph convolutional neural networks [6, 13, 37] to include structure-aware priors in their methods. A structured prediction layer (SPL) proposed in [1] encodes human joint dependencies by a hierarchical neural network design to model 3D human motion. HKMR [19] exploits a kinematics model to recover human meshes from 2D images, while [74] takes advantage of kinematic modeling to generate 3D joints. Inspired by these methods, we propose a forward kinematics model for a more powerful encoding of human structure solely from bone transformations to benefit occupancy learning of articulated objects. On the high-level, our formulation can be considered as inverse of the kinematics models proposed in HKMR and SPL that regress human body parameters from abstract feature vectors. Ours creates an efficient structural encoding from human body parameters.

Application: Placing people in 3D scenes. Recently, PSI [73] and PLACE [72] have been proposed to generate realistic human bodies in 3D scenes. However, these approaches *1)* require a high-quality scene mesh and the corresponding scene SDF to perform person-scene interpenetration tests and *2)* when multiple humans are generated in one scene, the results often exhibit unrealistic person-person interpenetrations. As presented in Sec. 6.4, these problems are addressed by representing human bodies with LEAP. As LEAP provides a differentiable volumetric occupancy representation of a human body, we propose an efficient point-based loss that minimizes the interpenetration between the human body and any other objects that are represented as point clouds.

3. Preliminaries

In this section, we start by reviewing the parametric human body model (SMPL [40]) and the widely used mesh deformation method: Linear Blend Skinning (LBS).

SMPL and its canonicalized shape correctives. SMPL body model [40] is an additive human body model that explicitly encodes identity- and pose-dependent deformations via additive mesh vertex offsets. The model is built from an artist-created mesh template $\bar{T} \in \mathbb{R}^{N \times 3}$ in the canonical pose by adding shape- and pose-dependent vertex offsets via *shape* $\mathsf{B}_S(\beta)$ and *pose* $\mathsf{B}_P(\theta)$ blend shape functions:

$$\bar{V} = \bar{T} + \mathsf{B}_S(\beta) + \mathsf{B}_P(\theta)\,, \tag{1}$$

where $\bar{V} \in \mathbb{R}^{N \times 3}$ are the modified canonical vertices. The linear blend shape function $\mathsf{B}_S(\beta; S)$ (2) is controlled by a vector of shape coefficients β and is parameterized by orthonormal principal components of shape displacements $\mathcal{S} \in \mathbb{R}^{N \times 3 \times |\beta|}$ that are learned from registered meshes.

$$\mathsf{B}_S(\beta; \mathcal{S}) = \sum\nolimits_{n=1}^{|\beta|} \beta_n \mathcal{S}_n \tag{2}$$

Similarly, the linear pose blend shape function $\mathsf{B}_P(\theta; P)$ (3) is parameterized by a learned pose blend shape matrix $\mathcal{P} = [\mathbf{P}_1, \ldots, \mathbf{P}_{9K}] \in \mathbb{R}^{N \times 3 \times 9K}(\mathbf{P}_n \in \mathbb{R}^{N \times 3})$ and is controlled by a per-joint rotation matrix $\theta = [r_0, r_1, \cdots, r_K]$, where K is the number of skeleton joints and $r_k \in \mathbb{R}^{3 \times 3}$ denotes the relative rotation matrix of part k with respect to its parent in the kinematic tree

$$\mathsf{B}_P(\theta; \mathcal{P}) = \sum_{n=1}^{9K}(vec(\theta)_n - vec(\theta^*)_n)\mathbf{P}_n. \quad (3)$$

Inspired by SMPL, LEAP captures the canonicalized occupancy of human bodies, where the shape correctives are modeled by deep neural networks and learned from data.

Regressing joints from body vertices. Joint locations $\mathbf{J} \in \mathbb{R}^{K \times 3}$ in SMPL are defined in the rest pose and depend on the body identity/shape parameter β. The relation between body shapes and joint locations is defined by a learned regression matrix $\mathcal{J} \in \mathbb{R}^{K \times N}$ that transforms rest body vertices into rest joint locations (4)

$$\mathbf{J} = \mathcal{J}(\bar{T} + \mathsf{B}_S(\beta; \mathcal{S})). \quad (4)$$

Regressing body vertices from joints. We observe that the regression of body joints from vertices (4) can be inverted and that we can directly regress body vertices from joint locations; if $K > |\beta|$ the problem is generally well constrained. For this, we first calculate the shape-dependent joint displacements $\mathbf{J}_\Delta \in \mathbb{R}^{K \times 3}$ by subtracting joints of the template mesh from the body joints (5) and then create a linear system of equations to express a relation between the joint displacements and shape coefficients (6)

$$\mathbf{J}_\Delta = \mathbf{J} - \mathcal{J}\bar{T} \quad (5)$$

$$\mathbf{J}_\Delta = \sum_n^{|\beta|} \mathcal{J}\mathcal{S}_n\beta_n. \quad (6)$$

This relation is useful to create an effective shape feature vector which will be demonstrated in Section. 4.1.1.

Linear Blend Skinning (LBS). Each modified vertex $\bar{V}_i$ (1) is deformed via a set of blend weights $\mathcal{W} \in \mathbb{R}^{N \times K}$ by a linear blend skinning function (9) that rotates vertices around joint locations $\mathbf{J}$:

$$G_k(\theta, \mathbf{J}) = \prod_{j \in A(k)} \left[\begin{array}{c|c} r_j & \mathbf{j}_j \\ \hline \vec{0} & 1 \end{array}\right] \quad (7)$$

$$B_k = G_k(\theta, \mathbf{J})G_k(\theta^*, \mathbf{J})^{-1} \quad (8)$$

$$V_i = \sum_{k=1}^{K} w_{k,i} B_k \bar{v}_i \quad (9)$$

where $w_{k,i}$ is an element of $\mathcal{W}$. Specifically, let $\mathcal{G} = \{G_k(\theta, \mathbf{J}) \in \mathbb{R}^{4 \times 4}\}_{k=1}^{K}$ be the set of K rigid bone transformation matrices that represent a 3D human body in a world coordinate (7). Then, $\mathcal{B} = \{B_k \in \mathbb{R}^{4 \times 4}\}_{k=1}^{K}$ is the set of local bone transformation matrices that convert the body

from the canonical space to a posed space (8), and $\mathbf{j}_j \in \mathbb{R}^3$ (an element of $\mathbf{J} \in \mathbb{R}^{K \times 3}$) represents jth joint location in the rest pose. $A(k)$ is the ordered set of ancestors of joint k.

Note that $\mathcal{W}$ is only defined for mesh vertices in SMPL. As presented in Section 4.2, LEAP proposes to parameterize the forward and the inverse LBS operations via neural networks in order to create generalized LBS weights that are defined for *every* point in 3D space.

4. LEAP: Learning occupancy of people

Overview. LEAP is an end-to-end differentiable occupancy function $f_\Theta(x|\mathcal{G}) : \mathbb{R}^3 \mapsto \mathbb{R}$ that predicts whether a query point $x \in \mathbb{R}^3$ is located inside the 3D human body represented by a set of K rigid bone transformation matrices $\mathcal{G}$ (7). The overview of our method is depicted in Fig. 2.

First, the bone transformation matrices $\mathcal{G}$ are taken by three feature encoders (Sec. 4.1) to produce a global feature vector z, which is then taken by a per-bone learnable linear projection module Π_{ω_k} to create a compact code $z_k \in \mathbb{R}^{12}$.

Second, the input transformations $\mathcal{G}$ are converted to the local bone transformations $\{B_k\}_{k=1}^{K}$ (8) that define per-bone transformations from the canonical to a posed space.

Third, an input query point $x \in \mathbb{R}^3$ is transformed to the canonical space via the inverse linear blend skinning network. Specifically, the inverse LBS network estimates the skinning weights $\hat{w}_x \in \mathbb{R}^K$ for the query point x (Sec. 4.2). Then, the corresponding point $\hat{\bar{x}}$ in the canonical space is obtained via the inverse LBS operation (10). Similarly, the weights $\hat{w}_x$ are also used to calculate the point feature vector z_x as a linear combination of the bone features z_k (11)

$$\hat{\bar{x}} = \left(\sum_{k=1}^{K} \hat{w}_x[k]B_k\right)^{-1} x, \quad (10)$$

$$z_x = \sum_{k=1}^{K} \hat{w}_x[k]z_k. \quad (11)$$

Fourth, the forward linear blend skinning network takes the estimated point $\hat{\bar{x}}$ in the canonical pose and predicts weights $\hat{w}_{\hat{\bar{x}}}$ that are used to estimate the input query point $\hat{x}$ via (12). This cycle (posed $\rightarrow$ canonical $\rightarrow$ posed space) defines an additional *cycle-distance* feature d_x (13) for the query point x

$$\hat{x} = \left(\sum_{k=1}^{K} \hat{w}_{\hat{\bar{x}}}[k]B_k\right) \hat{\bar{x}}, \quad (12)$$

$$d_x = \sum_{k=1}^{K} |\hat{w}_x[k] - \hat{w}_{\hat{\bar{x}}}[k]|. \quad (13)$$

Last, an occupancy multi-layer perceptron O_w (ONet) takes the canonicalized query point $\hat{\bar{x}}$, the local point code z_x and the cycle-distance feature d_x, and predicts whether the query point is inside the 3D human body:

$$\hat{o}_x = \begin{cases} 0, & \text{if } O_w(\hat{\bar{x}}|z_x, d_x) < 0.5 \\ 1, & \text{otherwise}. \end{cases} \quad (14)$$
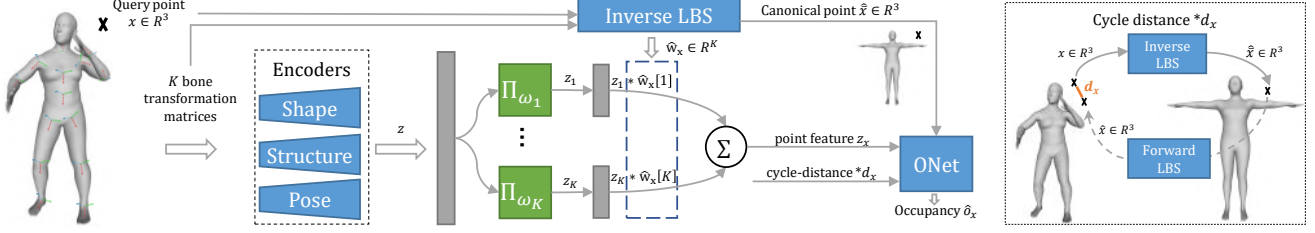
Figure 2. **Overview.** LEAP consists of three encoders that take K bone transformations $\mathcal{G}$ as input and create a global feature vector z that is further customized for each bone k through a per-bone learned projection module $\Pi_{\omega_k} : z \mapsto z_k$. Then, learned LBS weights $\hat{w}_x$ are used to estimate the position of the query point x in the canonical pose $\hat{\bar{x}}$ and to construct efficient local point features z_x, which are propagated together through an occupancy neural network with an additional cycle distance feature d_x. Blue blocks denote neural networks, green blocks are learnable linear layers, gray rectangles are feature vectors, and a black cross sign denotes query point $x \in \mathbb{R}^3$.

4.1. Encoders

We propose three encoders to leverage the prior knowledge about the kinematic structure (Sec. 4.1.2) and to encode shape-dependent (Sec. 4.1.1) and pose-dependent (Sec. 4.1.3) deformations

4.1.1 Shape encoder

As introduced in (Sec. 3), SMPL [40] is a statistical human body model that encodes prior knowledge about human shape variations. Therefore, we invert the SMPL model in a fully-differentiable and efficient way to create a shape prior from the input transformation matrices $\mathcal{G}$. Specifically, the input per-bone rigid transformation matrix (7) is decomposed to the joint location in the canonical pose $\mathbf{j}_k \in \mathbb{R}^3$ and the local bone transformation matrix B_k (8). The joint locations are then used to solve the linear system of equations (6) for the shape coefficients $\hat{\beta}$ and to further estimate the canonical mesh vertices $\hat{\bar{V}}$:

$$\hat{\bar{V}} = \bar{T} + \mathsf{B}_S(\hat{\beta}; \mathcal{S}) + \mathsf{B}_P(\theta; \mathcal{P}). \quad (15)$$

Similarly, the posed vertices $\hat{V}$, which are needed by the inverse LBS network, are estimated by applying the LBS function (9) on the canonical vertices $\hat{\bar{V}}$.

The mesh vertices $\hat{\bar{V}}$ and $\hat{V}$ are propagated through a PointNet [51] encoder to create the shape features for the canonicalized and posed human bodies, respectively.

Note that required operations for this process are *differentiable* and can be *efficiently* implemented by leveraging the model parameters of SMPL.

4.1.2 Structure encoder

Inspired by [1] and [19], we propose a structure encoder to effectively encode the kinematic structure of human bodies by explicitly modeling the joins dependencies.

The structured dependencies between joints are defined by a kinematic tree function $\tau(k)$ which, for the given bone k, returns the index of its parent. Following this definition,
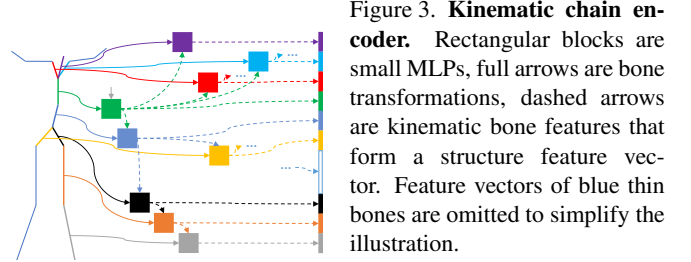

Figure 3. **Kinematic chain encoder.** Rectangular blocks are small MLPs, full arrows are bone transformations, dashed arrows are kinematic bone features that form a structure feature vector. Feature vectors of blue thin bones are omitted to simplify the illustration.

we propose a hierarchical neural network architecture (Figure 3) that consists of per-bone two-layer perceptrons m_{θ_k}. The input to m_{θ_k} consists of the joint location $\mathbf{j}_k$, bone length l_k and relative bone rotation matrix r_k of bone k with respect to its parent in the kinematic tree. Additionally, for the non-root bones, the corresponding m_{θ_k} also takes the feature of its parent bone. The output of each two-layer perceptron v_k^S (16) is then concatenated to form a structure feature v^S (17)

$$v_k^S = \begin{cases} m_{\theta_1}\left(vec(r_1) \oplus \mathbf{j}_1 \oplus l_1\right), & \text{if } k = 1 \\ m_{\theta_k}\left(vec(r_k) \oplus \mathbf{j}_k \oplus l_k \oplus v_{\tau(k)}^S\right), & \text{otherwise} \end{cases}$$
$$(16)$$

$$v^S = \oplus_{k=1}^K v_k^S, \quad (17)$$

where $\oplus$ is the feature concatenation operator.

4.1.3 Pose encoder

To capture pose-dependent deformations, we use the same projection module as NASA [15]. The root location $t_0 \in \mathbb{R}$ of the skeleton is projected to the local coordinate frame of each bone. These are then concatenated as one pose feature vector v^P (18)

$$v^P = \oplus_{k=1}^K B_k^{-1} t_0. \quad (18)$$

4.2. Learning linear blend skinning

Since our occupancy network (ONet) is defined in the canonical space, we need to map query points to the canonical space to perform the occupancy checks. However, the

conventional LBS weights are only defined on the body surface. To bridge this gap, we parameterize inverse LBS functions using neural networks and learn a valid skinning weight for *every* point in space.

Specifically, for a query point $x \in \mathbb{R}^3$, we use a simple MLP to estimate the skinning weight $\hat{w}_x \in \mathbb{R}^K$ to transform the point to the canonical space as in Eq. 10. The input to the MLP consists of the two shape features defined in Sec. 4.1.1 and a pose feature obtained from the input bone transformations $\mathcal{G}$.

Cycle-distance feature. Learning accurate inverse LBS weights is challenging as it is pose-dependent, requiring large amounts of training data. Consequently, the canonicalized occupancy network may produce wrong occupancy values for highly-articulated poses.

To address this, we introduce an auxiliary forward blend skinning network that estimates the skinning weights $\hat{w}_{\hat{\bar{x}}}$, which are used to project a point from the canonical to the posed space (12). The goal of this forward LBS network is to create a *cycle-distance* feature d_x that helps the occupancy network resolve ambiguous scenarios.

For instance, a query point x that is located outside the human geometry in the posed space can be mapped to a point that is located inside the body in the canonical space $\hat{\bar{x}}$. Here, our forward LBS network helps by projecting $\hat{\bar{x}}$ back to the posed space $\hat{x}$ (12) such that these two points define a *cycle distance* that provides information about whether the canonical point is associated with a different body part in the canonical pose and thus should be automatically marked as an outside point. This cycle distance (13) is defined as the l_1 distance between weights predicted by the inverse and the forward LBS networks. Our forward LBS network architecture is similar to the inverse LBS network. It takes the shape features as input, but without the bone transformations since the canonical pose is consistent across all subjects.

4.3. Training

We employ a two-stage training approach. First, both linear blend skinning networks are trained independently. Second, the weights of these two LBS networks are fixed and used as deterministic differentiable functions during the training of the occupancy network.

Learning the occupancy net. The parameters Θ of the learning pipeline $f_\Theta(x|\mathcal{G})$ (except LBS networks) are optimized by minimizing loss function (19):

$$L(\Theta) = \sum_{\mathcal{G} \in \{\mathcal{G}_e\}_{e=1}^E} \sum_{\{(x,o_x)\}_{i=1}^M \sim p(\mathcal{G})} (f_\Theta(x|\mathcal{G}) - o_x)^2, \ (19)$$

where o_x is the ground truth occupancy value for query point x. $\mathcal{G}$ represents a set of input bone transformation matrices and $p(\mathcal{G})$ represents the ground truth body surface. E is the batch size, and M is the number of sampled points per batch.

Learning the LBS nets. Learning the LBS nets is harder than learning the occupancy net in this work because the ground truth skinning weights are only sparsely defined on the mesh vertices. To address this, we create pseudo ground truth skinning weights for every point in the canonical and posed spaces by querying the closest human mesh vertex and using the corresponding SMPL skinning weights as ground truth. Then, both LBS networks are optimized by minimizing the l_1 distance between the predicted and the pseudo ground truth weights.

5. Application: Placing people in scenes

Recent generative approaches [72, 73] first synthesize human bodies in 3D scenes and then employ an optimization procedure to improve the realism of generated humans by avoiding collisions with the scene geometry. However, their human-scene collision loss requires high-quality scene SDFs that can be hard to obtain, and previously generated humans are not considered when generating new bodies, which often results in human-human collisions.

Here, we propose an effective approach to place *multiple persons* in 3D scenes in a physically plausible way. Given a 3D scene (represented by scene mesh or point clouds) and previously generated human bodies, we synthesize another human body using [72]. This new body may interpenetrate existing bodies and this cannot be resolved with the optimization framework proposed in [72] as it requires pre-defined signed distance fields of the 3D scene and existing human bodies. With LEAP, we can straightforwardly solve this problem: we represent the newly generated human body with our neural occupancy representation and resolve the collisions with the 3D scene and other humans by optimizing the input parameters of LEAP with a point-based loss (20). Note that, the parameters of LEAP are fixed during the optimization and we use it as a differentiable module with respect to its input.

Point-based loss. We introduce a *point-based loss* function (20) that can be used to resolve the collisions between the human body represented by LEAP and 3D scenes or other human bodies represented simply by point clouds:

$$l(x) = \begin{cases} 1, & \text{if } f_\Theta(x|\mathcal{G}) - 0.5 > 1 \\ 0, & \text{if } f_\Theta(x|\mathcal{G}) - 0.5 < 0 \quad (20) \\ f_\Theta(x|\mathcal{G}) - 0.5, & \text{otherwise.} \end{cases}$$

We employ an optimization procedure to refine the position of the LEAP body, such that there is no interpenetration with scene and other humans. Given LEAP, the collision detection can be performed without pre-computed scene SDFs. A straightforward way to resolve collisions with scene meshes is to treat mesh vertices as a point cloud and apply the point-based loss (20). A more effective way that we use in this work is to sample additional points along

Encoder type	IOU↑
Pose	91.86%
Shape	96.44%
Structure	97.49%
Shape + Structure	97.96%
Shape + Structure + Pose	**97.99%**

Table 1. **Impact of feature encoders.** Each encoder has a positive contribution to the reconstruction quality, while the best result is achieved when all three encoders are combined.

the opposite direction of the mesh vertex normals and thus impose an effective oriented volumetric error signal to avoid human-scene interpenetrations.

6. Experiments

We ablate the proposed feature encoders (Sec. 6.1), show the ability of LEAP to represent multiple people (Sec. 6.2), demonstrate the generalization capability of LEAP on unseen poses and unseen subjects (Sec. 6.3), and show how our method is used to place people in 3D scenes by using the proposed point-based loss (Sec. 6.4).

Experimental setup. Training data for our method consists of sampled query points x, corresponding occupancy ground truth values o_x and pseudo skinning weights w_x, bone transformations $\mathcal{G}$, and SMPL [40] parameters. We use the DFaust [5] and MoVi [20] datasets, and follow a similar data preparation procedure as [15]. A total of 200k training points are sampled for every pose; one half are sampled uniformly within a scaled bounding box around the human body (10% padding) and the other half are normally distributed around the mesh surface $x \sim \mathcal{N}(m, 0.01)$ (m are randomly selected points on the mesh triangles).

We use the Adam optimizer [35] with a learning rate of 10^{-4} across all experiments and report mean Intersection Over Union (IOU) in percentages and Chamfer distance (Ch.) scaled by the factor of 10^4. Our models presented in this section use a fully articulated body and hand model with $K = 52$ bones (SMPL+H [53] skeleton) and are trained in two stages (Sec. 4.3). The training takes about 200k iterations from the beginning without any pretraining with a batch size of 55. Our baseline, NASA [15], is trained with the parameters specified in their paper, except for the number of bones (increased to 52) and the number of training steps (increased from 200k to 300k).

6.1. The impact of feature encoders

We first quantify the effect of each feature encoder introduced in Sec. 4.1. For this experiment, we use 119 randomly selected training DFaust sequences (≈ 300 frames) of 10 subjects and evaluate results on 1 unseen sequence per subject.

To better understand the efficacy of the encoding schemes, we replace the inverse LBS network with a deterministic searching procedure that creates pseudo ground truth weights w_x at inference time (Sec. 4.3). This procedure, based on the greedy nearest neighbor search, is

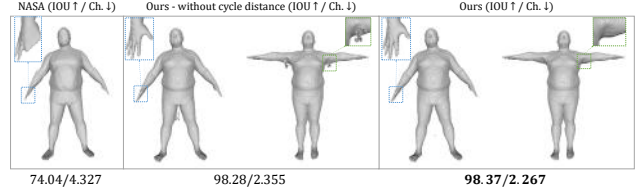| NASA (IOU ↑ / Ch. ↓)	Ours - without cycle distance (IOU ↑ / Ch. ↓)	Ours (IOU ↑ / Ch. ↓)
74.04/4.327	98.28/2.355	**98.37/2.267**

Figure 4. **Multi-person occupancy on DFaust [5].** Results demonstrate that our method can represent small details much better (hand) and the proposed cycle-distance feature further improves the reconstruction quality (armpits). Several high-resolution images of LEAP are given in Figure 1.

Experiment type	NASA[15] (IOU↑/Ch.↓)	Ours (IOU↑/Ch.↓)
Unseen poses	73.69/4.72	**98.39/2.27**
Unseen subjects	78.87/3.67	**92.97/2.80**

Table 2. **Generalization.** Unseen pose and unseen subject experiments (Sec. 6.3) on DFaust [5] and MoVi [20] respectively.

not differentiable w.r.t. the input points, but it provides deterministic LBS weights to ablate encoders. We train the models with 100k iterations and report IOU on unseen sequences in Table 1. We find that the structure encoder has the biggest impact on the model performance and the combination of all three encoders yields the best results.

6.2. Multi-person occupancy

We use the same training/test split as in Sec. 6.1 to evaluate the representation power of our multi-person occupancy. Average qualitative and quantitative results on the test set of our model with and without the cycle distance (13) are displayed in Figure 4, respectively.

Our method has significantly higher representation power than NASA [15]. High-frequency details are better preserved and the connections between adjacent bones are smoother. The cycle-distance feature further improves results, which is highlighted in the illustrated close-ups.

6.3. Generalization

In the previous experiment, we evaluated our model on unseen poses of different humans for actions that were performed by at least one training subject, while here we go one step further and show that **1)** our method generalizes to unseen poses on actions that were not observed during the training and **2)** that our method generalizes even to unseen subjects (Table 2).

For the unseen pose generalization experiment, we use all DFaust [5] subjects and leave out one randomly selected action for evaluation and use the remaining sequences for training. For the unseen subject experiment, we show the ability of our method to represent a much larger set of subjects and to generalize to unseen ones. We use 10 sequences of 86 MoVi [20] subjects and leave out every 10-th subject for evaluation with one randomly selected sequence. Re-

Collision score		PLACE [72]	Ours
human-scene	↓	5.72%	5.72%
scene-human	↓	3.51%	**0.62%**
human-human	↓	5.73%	**1.06%**

Table 3. **Comparison with PLACE [72].** Our optimization method successfully reduces the collisions with 3D scenes and other humans.

PLACE [72]   Our optimization

Figure 5. **Comparison with PLACE [72].** Optimization with the point-based loss successfully resolves interpenetrations with other humans and 3D scenes that are represented with point clouds.

sults show that LEAP largely improves the performance in both settings. Particularly, for the unseen poses, LEAP improves the IOU from 73.69% to 98.39%, clearly demonstrating the benefits of the proposed occupancy representation in terms of fidelity and generality.

6.4. Placing people in 3D scenes

In this section, we demonstrate the application of LEAP to the task of placing people in a 3D scene. We generate 50 people in a Replica [61] room using PLACE [72] and select pairs of humans that collide, resulting in 151 pairs. Then for each person pair, the proposed point-based loss (20) is used in an iterative optimization framework to optimize the global position of one person, similarly to the trajectory optimization proposed in [69]. The person, whose position is being optimized, is represented by LEAP, while other human bodies and the 3D scene are represented by point clouds. We perform a maximum of 1000 optimization steps or stop the convergence when there is no intersection with other human bodies and with the scene.

Note that other pose parameters are fixed and not optimized since PLACE generates semantically meaningful and realistic poses. Our goal is to demonstrate that LEAP can be efficiently and effectively utilized to resolve human-human and human-scene collisions.

Evaluation: We report *human-scene* collision scores defined as the percentage of human mesh vertices that penetrate the scene geometry, *scene-human* scores that represent the normalized number of scene vertices that penetrate the human body, and *human-human* collision scores defined as the percentage of human vertices that penetrate the other human body.

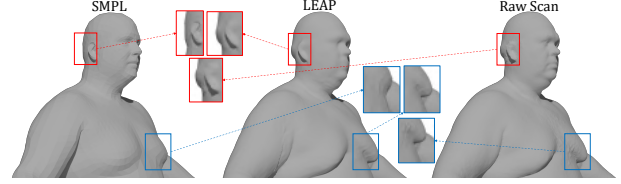Quantitative (Table 3) and qualitative (Figure 5) results

Figure 6. **LEAP learned from raw scans of a single DFaust [5] subject.** LEAP (middle) captures more shape details than SMPL with 10 shape components (left).

demonstrate that our method successfully optimizes the location of human body by avoiding collisions with the scene mesh and other humans represented by point clouds. Note that the human-scene score has remained almost unchanged because of the noisy scene SDF that is used to compute this metric. However, we keep it for a fair comparison with the baseline [72] that uses it for evaluation. Furthermore, the inference time of LEAP for the occupancy checks with respect to the other human body (6080 points) is 0.14s. This is significantly faster than [29] (25.77s), which implements differentiable interpenetration checks using a 256^3-resolution volumetric grid for fine approximation.

7. Conclusion

We introduced LEAP, a novel articulated occupancy representation that generalizes well across a variety of human shapes and poses. Given a set of bone transformations and 3D query points, LEAP performs efficient occupancy checks on the points, resulting in a fully-differentiable volumetric representation of the posed human body. Like SMPL, LEAP represents both identity- and pose-dependent shape variations. Results show that LEAP outperforms NASA in terms of generalization ability and fidelity in all settings. Furthermore, we introduced an effective point-based loss that can be used to efficiently resolve the collisions between human and objects that are represented by point clouds.

Future work. We plan to learn LEAP from imagery data and extend it to articulated clothed bodies. Preliminary results (Figure 6) of learning LEAP from raw scans show that LEAP can represent realistic surface details, motivating the future extension of LEAP to model clothed humans.

LEAP: Learning Articulated Occupancy of People
– Supplementary Material –

A. Overview

In this supplementary document, we first provide details about the proposed neural network modules (Sec. B) and their training procedure (Sec. C). Then, we present more qualitative and quantitative results that were not included in the paper due to the page limit (Sec. D). We also discuss the limitations of the proposed method (Sec. E). Lastly, we summarize the notations used in the paper to improve paper readability (Sec. F).

B. Network architecture

We first present details about the proposed neural encoders that are used to create the global feature vector $z \in \mathbb{R}^{596}$ (128, 312, and 156 dimensions for structure, shape, and pose features respectively) and then detail the proposed neural network architectures.

B.1. Structure encoder

The structure encoder consists of 52 small multi-layer perceptrons that are organized in a tree structure, where each node of the tree corresponds to one joint in the human skeleton and outputs a compact feature vector $b_k \in \mathbb{R}^6$.

Each MLP node (Table B.1) takes as input a 19-dimensional feature vector – 6 dimensions for the parent feature, 9 for the rotation matrix, 1 for the bone length, and 3 for the joint location – and outputs a small bone code. These bone codes are concatenated to form one structure feature vector as explained in the main paper.

Since the root node does not have a parent node to be conditioned on its feature vector $b_0 \in \mathbb{R}^6$, we create b_0 with a single linear layer that takes as input vectorized θ pose parameters and joint locations $\mathbf{J}$.

B.2. PointNet encoder

We implement a PointNet encoder (Figure B.1) to encode a point cloud into a fix-size feature vector. This network is used in Sec. 4.1.1 to create a 128-dimensional shape

feature vector ($P = 128$ in Figure B.1) and two 100-dimensional feature vectors for the inverse and the forward LBS networks ($P = 100$).

B.3. Bone projection layers

The bone projection layers $\Pi_{\omega_k} : \mathbb{R}^{596} \mapsto \mathbb{R}^{12}$ create small per-bone features z_k and are implemented as efficient grouped 1D convolutions [26].

B.4. ONet

The architecture of the occupancy network is similar to the one proposed in [42] and is illustrated in Figure B.2.

B.5. Linear blend skinning networks

The inverse and the forward LBS networks are similar and implemented as MLPs conditioned on a latent feature vector (Figure B.3).

Forward LBS network. The latent feature vector for the forward LBS network $c_{\text{fwd}} \in \mathbb{R}^{200}$ is created as a concate-
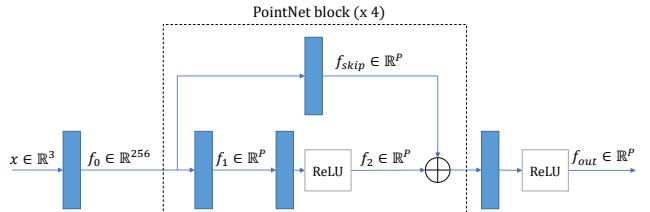

Figure B.1. **PointNet architecture.** The PointNet encoder encodes an arbitrary set of input points $\{x \in \mathbb{R}^3\}$ into a feature vector of length P. All layers except for the skip connections use bias. Blue rectangular blocks are trainable linear layers. $\oplus$ operator is the element-wise sum.
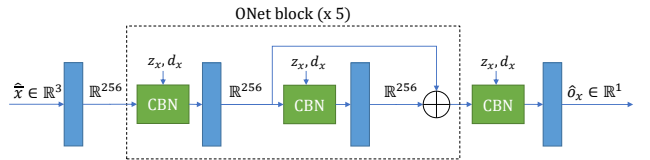

Figure B.2. **ONet architecture.** The occupancy neural network is implemented as a multi-layer perceptron that consists of five ONet blocks. Blue rectangular blocks are trainable linear layers. $\oplus$ operator is the element-wise summation. CBN blocks are Conditional Batch-Normalization blocks [14, 16, 42] followed by ReLU activation function that are conditioned on the local point feature $z_x \in \mathbb{R}^{12}$ and the cycle-distance feature d_x.

Linear$(19, 19)$ + bias
ReLU
Linear$(19, 6)$ + bias
ReLU

Table B.1. **The MLPs of the structure encoder**. This compact neural network architecture consists of 500 trainable parameters.
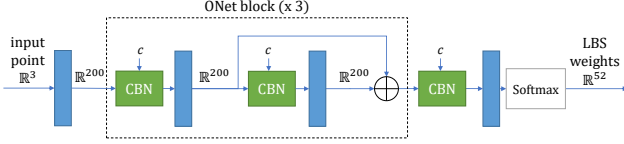
Figure B.3. **The architecture of the inverse and the forward LBS networks.** Both networks use this architecture to regress skinning weights. Blue rectangular blocks are trainable linear layers. $\oplus$ operator is the element-wise sum. ONet and CBN blocks are introduced in Figure B.2. The feature vector c is $c_{\mathrm{inv}} \in \mathbb{R}^{280}$ for the inverse LBS network and $c_{\mathrm{fwd}} \in \mathbb{R}^{200}$ for the forward LBS network.

nation of two 100-dimensional feature vectors created by the PointNet encoder. One is created by encoding the estimated canonical vertices $\hat{V}$, and the other one is created by encoding the estimated posed vertices $\hat{V}$.

Inverse LBS network. The latent feature vector for the inverse LBS network $c_{\mathrm{inv}} \in \mathbb{R}^{280}$ is created as a concatenation of the conditional features produced for the forward LBS network c_{fwd} and an additional 80-dimensional feature vector created by a single linear layer that takes as input concatenated vectorized pose parameters and joint locations.

C. Training

We provide additional details for three independent training procedures. First, we train the inverse and the forward LBS networks and then use these two modules as deterministic differentiable functions for the occupancy training.

Occupancy training. All modules except the LBS networks are trained together and have a total of about 1.5M trainable parameters. For each batch, 1536 points are sampled uniformly and 1536 near the surface (1024 in the posed and 512 in the canonical space). Points that are sampled directly in the canonical space are not propagated through the forward LBS network and are associate with pseudo ground truth skinning weights $w_{\bar{x}}$ to calculate local codes z_x.

The inverse LBS network has about 1.5M parameters. Each training batch consists of 1024 uniformly sampled points and 1024 points sampled around the mesh surface.

The forward LBS network has about 1.2M parameters. Each training batch consists of 1024 points sampled in the canonical space (512 uniformly sampled and 512 sampled near the surface) and points that are sampled for the training of the inverse LBS network. The latter set of points is mapped to the canonical space via the proposed pseudo ground truth weights.

D. Additional experiments and results

We supplement experiments for generalization (Figure D.4), for learning LBS (Sec. D.1), and placing people

in scenes (Sec. D.2).

D.1. Evaluation of linear blend skinning networks

Our forward LBS network operates in the canonical space and does not need to deal with challenging human poses as the inverse LBS network. Here, we quantify the performance gap between these two networks on the unseen portion of query points for three experimental setups presented in the main paper.

As an evaluation metric, we report the l_1 distance between pseudo ground truth weights and predicted weights by the inverse l_1^{inv} and the forward l_1^{fwd} LBS networks.

Quantitative results (Table D.2) show that the forward LBS network consistently outperforms the inverse LBS network across all settings. Furthermore, the inverse LBS network performs worse when subjects are not seen during the training.

D.2. Generating people in scenes

We provide additional qualitative results (Figure D.5) for the experiment presented in Figure 5 and visualize SDF (Figure D.7) that is used to compute the *human-scene* collision score. Although the SDF is very noisy, we still use it to compute this score for a fair comparison with PLACE [72].

We further provide an experiment on a larger Replica room [61]. Similar to Sec. 6.4, we sample 50 people from PLACE [72] and select 60 human body pairs that interpenetrate. These pairs are then optimized with our method by minimizing the proposed point-based loss (20).

Quantitative results (Table D.3) demonstrate that our approach improved collision scores over the baseline [72], except for the human-scene score which is unreliable due to the aforementioned noisy SDF (Figure D.7). The qualitative results displayed in Figure D.6 show that our method successfully resolves deep interpenetrations with scene geometry which could not be straightforwardly achieved with differentiable mesh-based collision methods – for example, a modified version of the approach presented in [62]. This indicates that our volumetric error signal is more effective than the surface error signal imposed by mesh-based methods.

E. Limitations

We observed some challenging scenarios in which our learned inverse linear blend skinning network may fail to correctly map a query point to the canonical space and consequently distort occupancy in the posed space. This problem occurs when the network is not well trained and two body parts are close to each other or even self-intersect. An example of a failure case of an unseen subject is displayed in Figure E.8. Therefore, a promising future direction is to explicitly model self-contact for learning the occupancy representation.
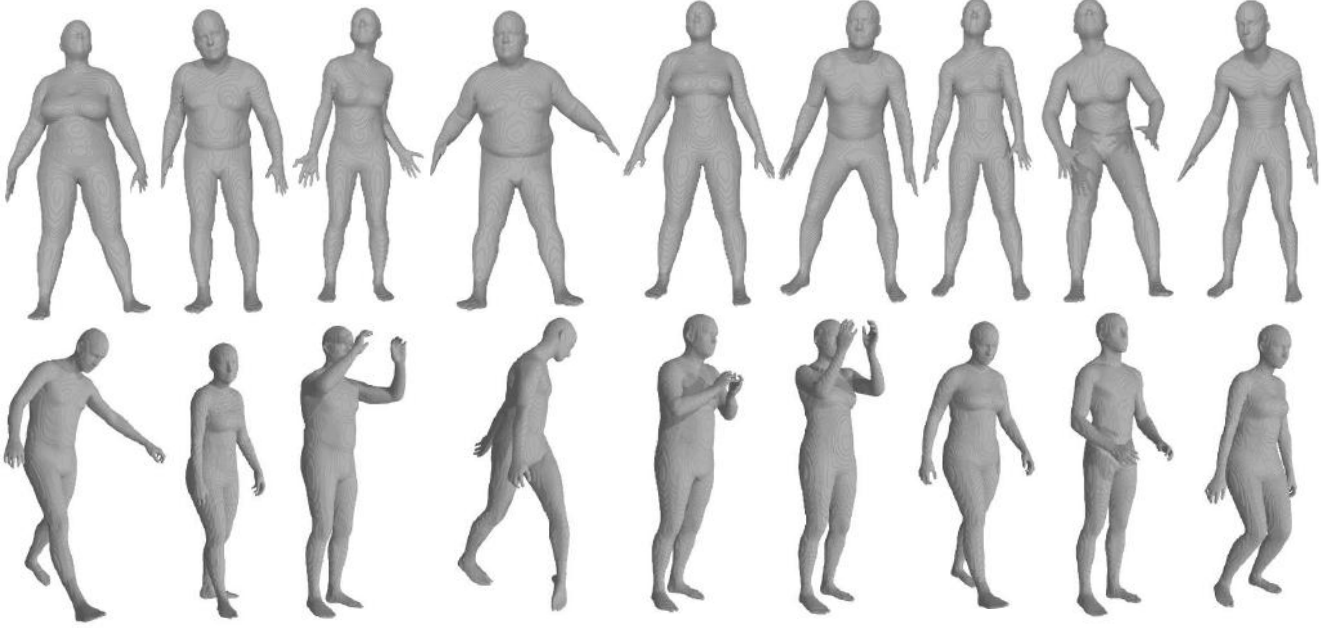
Figure D.4. **Generalization experiment.** Qualitative results for the generalization experiment (Sec. 6.3, Table 2) for DFaust [5] unseen poses (top row) and MoVi [20] unseen subjects (bottom row).

Experiment type	$l_1^{inv} \downarrow$	$l_1^{fwd} \downarrow$
Multi-person occupancy (Sec. 6.2)	0.1894	0.1252
Generalization: unseen poses (Sec. 6.3)	0.1997	0.0818
Generalization: unseen subjects (Sec. 6.3)	0.2138	0.1098

Table D.2. **Evaluation of the inverse and the forward linear blend skinning networks.** Reported l_1 distance shows that the forward LBS network consistently outperforms the inverse LBS network across all experiment settings. This is expected because the inverse LBS network reasons about different body shapes and poses, while the fwd-LBS network reasons only about body shapes since the pose in the canonical space is constant. "Multi-person" and "unseen poses" experiments are performed on the DFaust [5] dataset, while the "unseen subjects" experiment is performed on the MoVi [20] dataset. More details on the experimental setups are available in the paper (Sec. 6.2, Sec. 6.3).

F. Notation

Lastly, we summarize the key notation terms in Table F.4 for improved readability.

Collision score		Room 1 (Figure D.5)		Room 2 (Figure D.6)	
		PLACE [72]	Ours	PLACE [72]	Ours
human-scene	↓	5.72%	5.72%	**0.34**%	1.20%
scene-human	↓	3.51%	**0.62%**	0.98%	**0.77%**
human-human	↓	5.73%	**1.06%**	7.64%	**1.09%**

Table D.3. **Improved PLACE [72].** Results on two Replica [61] rooms. Our proposed optimization method successfully mitigates interpenetrations between scene geometry and other humans. Note that the human-scene score is unreliable metric due to noisy scene SDF.
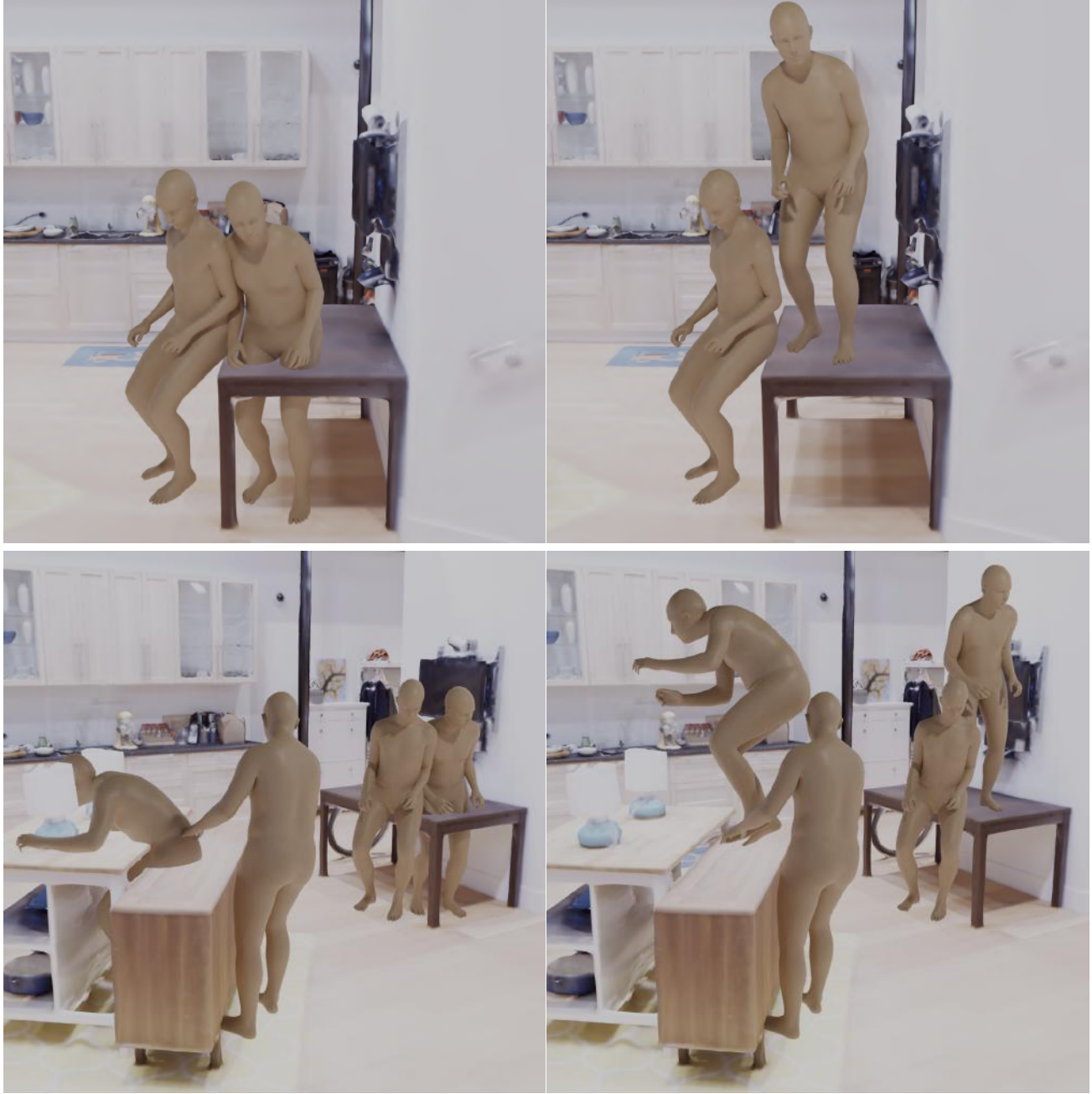
PLACE [72]   Our optimization

Figure D.5. **Improved PLACE [72].** Additional viewpoints of a Replica room [61] for results presented in the paper (Figure 5). Our point-based loss effectively resolves collisions of the human pairs. Quantitative results are reported in Table D.3.

PLACE [72]   Our optimization

Figure D.6. **Improved PLACE [72].** Results demonstrate that our method can resolve challenging interpenetrations with scene geometry. Note that this complex penetrations with the thin mesh geometry cannot be straightforwardly fixed with mesh-based intersection methods [62] that impose a surface-based error signal. This demonstrates that our flexible volumetric point-based loss is more efficient, which is quantified by the improved collisions scores displayed in Table D.3.

Figure D.7. **Noisy SDF** that is used to compute the human-scene score for results presented in the paper (Sec. 6.4, Table 3, Figure 5).
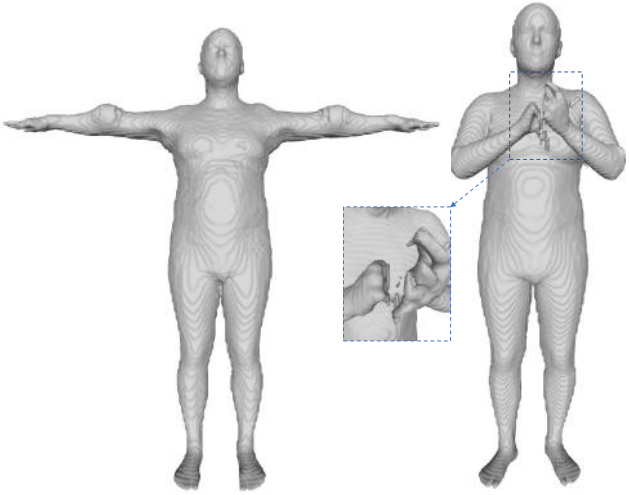

Figure E.8. **Failure case.** An example of an unseen MoVi [20] subject when two hands self-intersect. The inverse LBS network may incorrectly map a given query point to the canonical space for self-intersected regions which consequently distorts occupancy representation in the posed space.

<div align="center">Input parameters</div>

$K \in \mathbb{R}$	:	the number of input bones (52)
$x \in \mathbb{R}^3$	:	query point
$G_k \in \mathbb{R}^{4,4}$	:	bone transformation matrix of part k

<div align="center">SMPL parameters</div>

N	:	the number of vertices (6890)
$\mathcal{S}$	:	shape blendshape parameters
$\mathcal{P}$	:	pose blendshape parameters
$\mathcal{W}$	:	blend weights
$\mathcal{J}$	:	joint regressor matrix
$\bar{\mathbf{T}} \in \mathbb{R}^{N,3}$	:	template mesh
$\mathbf{V} \in \mathbb{R}^{N,3}$	:	mesh vertices
$\bar{\mathbf{V}} \in \mathbb{R}^{N,3}$	:	canonical mesh vertices

<div align="center">SMPL functions</div>

B_P	:	pose blendshape function
B_S	:	shape blendshape function

<div align="center">Estimated parameters</div>

$\hat{\bar{x}} \in \mathbb{R}^3$	:	estimated canonical point
$\hat{\mathbf{V}} \in \mathbb{R}^{N,3}$	:	estimated mesh vertices
$\hat{\bar{\mathbf{V}}} \in \mathbb{R}^{N,3}$	:	estimated canonical mesh vertices
$w_{\hat{x}} \in \mathbb{R}^K$	:	weights predicted by the inverse LBS network
$w_{\hat{\bar{x}}} \in \mathbb{R}^K$	:	weights predicted by the forward LBS network

Table F.4. **Notation summary.**

References

[1] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. In *Proc. International Conference on Computer Vision (ICCV)*, 2019. 3, 5

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

[3] Matan Atzmon and Yaron Lipman. SALD: Sign agnostic learning with derivatives. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. 3

[4] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. LoopReg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. In *Proc. Neural Information Processing Systems (NeurIPS)*, December 2020. 3

[5] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7, 8, 3

[6] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proc. International Conference on Computer Vision (ICCV)*, 2019. 3

[7] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 3

[8] Will Chang and Matthias Zwicker. Range scan registration using reduced deformable models. In *Computer Graphics Forum*, 2009. 2

[9] Yinpeng Chen, Zicheng Liu, and Zhengyou Zhang. Tensor-based human body modeling. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

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

[12] Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020. 3

[13] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Proc. International Conference on Computer Vision (ICCV)*, 2019. 3

[14] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2017. 1

[15] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 1, 3, 5, 7

[16] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *Proc. International Conference on Learning Representations (ICLR)*, 2017. 1

[17] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 2005. 3

[18] Oren Freifeld and Michael J Black. Lie bodies: A manifold representation of 3d human shape. In *Proc. European Conference on Computer Vision (ECCV)*, 2012. 2

[19] Georgios Georgakis, Ren Li, Srikrishna Karanam, Terrence Chen, Jana Kosecka, and Ziyan Wu. Hierarchical kinematic human mesh recovery. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 1, 3, 5

[20] Saeed Ghorbani, Kimia Mahdaviani, Anne Thaler, Konrad Kording, Douglas James Cook, Gunnar Blohm, and Nikolaus F Troje. Movi: A large multipurpose motion and video dataset. *arXiv preprint arXiv:2003.01888*, 2020. 7, 3, 6

[21] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proc. International Conference on Machine Learning (ICML)*, 2020. 3

[22] Nils Hasler, Thorsten Thormählen, Bodo Rosenhahn, and Hans-Peter Seidel. Learning skeletons for shape and pose. In *Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2010. 2

[23] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *Proc. International Conference on Computer Vision (ICCV)*, 2019. 2

[24] David T. Hoffmann, Dimitrios Tzionas, Michael J. Black, and Siyu Tang. Learning to train with synthetic humans. In *Proc. German Conference on Pattern Recognition (GCPR)*, 2019. 1

[25] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[26] Yani Ioannou, Duncan Robertson, Roberto Cipolla, and Antonio Criminisi. Deep roots: Improving cnn efficiency with hierarchical filter groups. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[27] Alec Jacobson, Zhigang Deng, Ladislav Kavan, and JP Lewis. Skinning: Real-time shape deformation. In *ACM SIGGRAPH 2014 Courses*, 2014. 2

[28] Timothy Jeruzalski, David IW Levin, Alec Jacobson, Paul Lalonde, Mohammad Norouzi, and Andrea Tagliasacchi. NiLBS: Neural inverse linear blend skinning. *arXiv preprint arXiv:2004.05980*, 2020. 3

[29] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *Proc. Interna-*

tional Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 2, 8

[30] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE Transactions on Visualization and Computer Graphics*, 2015. 3

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

[32] Tero Karras. Maximizing parallelism in the construction of bvhs, octrees, and k-d trees. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2012. 2

[33] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael Black, Krikamol Muandet, and Siyu Tang. Grasping Field: Learning implicit representations for human grasps. In *Proc. International Conference on 3D Vision (3DV)*. IEEE, 2020. 3

[34] Ladislav Kavan and Jiří Žára. Spherical blend skinning: a real-time deformation of articulated models. In *Proc. Symposium on Interactive 3D graphics and games*, 2005. 2

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015. 7

[36] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proc. International Conference on Computer Vision (ICCV)*, 2019. 1

[37] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[38] John P Lewis, Matt Cordner, and Nickson Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 2000. 2

[39] Ming C Lin, Dinesh Manocha, Jon Cohen, and Stefan Gottschalk. Collision detection: Algorithms and applications. *Algorithms for robotic motion and manipulation*, 1997. 2

[40] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 2015. 1, 2, 3, 5, 7

[41] Bruce Merry, Patrick Marais, and James Gain. Animation space: A truly linear framework for character animation. *ACM Transactions on Graphics*, 2006. 2

[42] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3d reconstruction in function space. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[43] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proc. International Conference on Computer Vision (ICCV)*, 2019. 3

[44] Marko Mihajlovic, Silvan Weder, Marc Pollefeys, and Mar-

tin R. Oswald. DeepSurfels: Learning online appearance fusion. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

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

[46] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics*, 2013. 3

[47] Ahmed A. A. Osman, Timo Bolkart, and Michael J. Black. STAR: Sparse trained articulated human body regressor. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[48] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3

[49] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 1, 3

[50] Ralf Plankers and Pascal Fua. Articulated soft objects for multiview shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003. 2

[51] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[52] Anurag Ranjan, David T Hoffmann, Dimitrios Tzionas, Siyu Tang, Javier Romero, and Michael J Black. Learning multi-human optical flow. *International Journal of Computer Vision*, 2020. 1

[53] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics*, 2017. 2, 7

[54] 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. International Conference on Computer Vision (ICCV)*, 2019. 2, 3

[55] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. PIFuHD: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3

[56] Shunsuke Saito, Jinlong Yang, Qianli Ma1, and Michael J. Black. SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[57] Hanan Samet. *The design and analysis of spatial data structures*. Addison-Wesley Reading, MA, 1990. 2

[58] Christian Sigg. *Representation and rendering of implicit surfaces*. PhD thesis, ETH Zurich, 2006. 3

[59] Vincent Sitzmann, Julien Martel, Alexander Bergman, David

Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020. 3

[60] Frank Steinbrucker, Christian Kerl, and Daniel Cremers. Large-scale multi-resolution surface reconstruction from rgb-d sequences. In *Proc. International Conference on Computer Vision (ICCV)*, 2013. 3

[61] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 8, 2, 3, 4

[62] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 2016. 2, 5

[63] Shaofei Wang, Andreas Geiger, and Siyu Tang. Locally aware piecewise transformation fields for 3d human mesh registration. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[64] Xiaohuan Corina Wang and Cary Phillips. Multi-weight enveloping: least-squares approximation techniques for skin animation. In *Annual Conference of the European Association for Computer Graphics (Eurographics)*, 2002. 2

[65] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[66] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. DISN: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2019. 3

[67] Yi Yang and Deva Ramanan. Articulated human detection with flexible mixtures of parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012. 3

[68] 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 *Proc. Neural Information Processing Systems (NeurIPS)*, 2020. 3

[69] Andrei Zanfir, Elisabeta Marinoiu, and Cristian Sminchisescu. Monocular 3d pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 8

[70] Ming Zeng, Fukai Zhao, Jiaxiang Zheng, and Xinguo Liu. A memory-efficient kinectfusion using octree. In *International Conference on Computational Visual Media*, 2012. 3

[71] Ming Zeng, Fukai Zhao, Jiaxiang Zheng, and Xinguo Liu. Octree-based fusion for realtime 3d reconstruction. *Graphical Models*, 2013. 3

[72] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J Black, and Siyu Tang. PLACE: Proximity learning of articulation and contact in 3d environments. In *Proc. International Conference on 3D Vision (3DV)*, 2020. 1, 2, 3, 6, 8, 4, 5

[73] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J Black, and Siyu Tang. Generating 3d people in scenes without people. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 6

[74] Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. Deep kinematic pose regression. In *Proc. European Conference on Computer Vision (ECCV)*, 2016. 3