RECONSTRUCTING 3D HUMANS FROM IMAGES

Nikolaos Kolotouros

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2022

Supervisor of Dissertation

Kostas Daniilidis, Professor, Computer and Information Science

Graduate Group Chairperson

Mayur Naik, Professor, Computer and Information Science

Dissertation Committee

Dinesh Jayaraman, Assistant Professor, Computer and Information Science

Jianbo Shi, Professor, Computer and Information Science

Pratik Chaudhari, Assistant Professor, Electrical and Systems Engineering

Michael J. Black, Director, Max Planck Institute for Intelligent Systems

RECONSTRUCTING 3D HUMANS FROM IMAGES

ACKNOWLEDGEMENT

First of all I would like to thank my advisor Kostas Daniilidis. He was the one that believed in me in the first place and gave me the chance to pursue my PhD at Penn. I want to thank him for our productive discussions throughout the years and for being a great mentor. His experience was really valuable in helping me navigate the world of research. For a young PhD student it's often hard to get started in research, but Kostas inspired me to work on 3D Computer Vision and gave me the freedom to chose a research topic that excited me. Kostas also made sure we always had all the resources needed so that we can focus on our research. He always had trust in me and even allowed me to work remotely from Greece during the pandemic. I am really grateful for all of this.

I would also like to thank Michael Black for giving me the opportunity to spend 6 amazing months as a visiting student at the Max Planck Institute in Tübingen. Unfortunately my visit there coincided with the outbreak of the Covid-19 pandemic so I only got the chance to spend 2 months physically in Germany. Nevertheless, I really learned a lot from my personal interactions with him. Michael is a great mentor, with fresh ideas and a clear vision for the future of the field. I want to also thank Michael for being very helpful with last minute requests such as writing recommendation letters or accepting to serve on my doctoral committee.

I am especially grateful for the other members of my thesis committee; Dinesh, Jianbo and Pratik. They were really responsive and made it possible to finish my WPE2, thesis proposal and defense in less than a month. I did not have a lot of time to interact with Dinesh and Pratik them in person due to the pandemic and me being away on internships but I enjoyed our discussions, their great insight and the hard questions they asked me during my presentations. For Jianbo specifically, I will remember our random encounters in Levine and fun research discussions in his office. I also had a great time working as a teaching assistant in his Deep Learning course twice.

Next, I want to dedicate a full paragraph to my amazing collaborator and mentor George Pavlakos. Through George I learned how to do approach a research problem, how to pose the right questions, how to write a paper, how to prepare for the rebuttals. It's extremely hard to find another person that you can collaborate so well, blending work and fun so seamlessly. With George we share the same work ethic and taste for 90's Greek tv shows. We also accomplished an amazing feat: we published 5 papers together, and never had a paper rejected. Looking forward to future collaborations with him!

I want to also thank my labmates and collaborators. Ken, for building the entire GRASP compute infrastructure; Oleh for the all the nice and full-of-mutual-sarcasm discussions we had throughout the years; Spiros for his valuable advice as a senior member of the lab; Berndt for teaching me about the US from a European's perspective; A shout out to some of the MPI people too: Vassilis, Nikos, Ahmed, Soubhik. They made my short stay there enjoyable and I hope I'll see you again now that I'm coming back to Europe.

A big thanks to the rest of my friends here in Philadelphia. First and foremost to my roommate Tasos; we spent countless hours playing music and arguing about politics. See you in Zurich Taso! Vassilis for our never-ending discussions about pretty much everything. The rest of Greek gang: George Kissas, Eliza, Vaso, Ioanna, Vagelis, Mariliza. Special thanks to George for letting me stay in his room thee the last months before my graduation.

Last but not least, I want thank my friends and family from Greece. My parents Thanasis and Mimika and my brother Giannis for always being there for me. Mary for her constant support in the first years of my studies. My highschool gang from Amaliada (in alphabetical order): Nikos, Stefanos, Thanasis, Thodoris. Stelios for unknowingly being the cover model for my job talk. I really hope that I'm not leaving out anyone.

ABSTRACT

RECONSTRUCTING 3D HUMANS FROM IMAGES

Nikolaos Kolotouros

Kostas Daniilidis

The past decade we have seen remarkable progress in Computer Vision, fueled by the recent advances in Deep Learning. Unsurprisingly, human perception has been the center of attention. We now have access to systems that can work remarkably well for traditional 2D tasks like segmentation or pose estimation. However, scaling this to 3D remains particularly challenging because of the inherent ambiguities and the scarcity of annotations.

The goal of this dissertation is to describe our contributions towards automating 3D human reconstruction from images. First, we will explore the use of different representations for human mesh recovery, discuss their advantages and show how they can be useful for learning deformations beyond standard parametric body models. Next, motivated by the limited availability of annotated data, we will present a method that leverages a collaboration between regression and optimization methods to successfully address this. Subsequently, we will describe our work on modeling the ambiguities in 3D human reconstruction and demonstrate its usefulness for solving a variety of downstream tasks, such as human body model fitting. Last, we will move beyond single-person 3D pose estimation and show how we can scale our methods to work on challenging real-world scenes with multiple humans.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

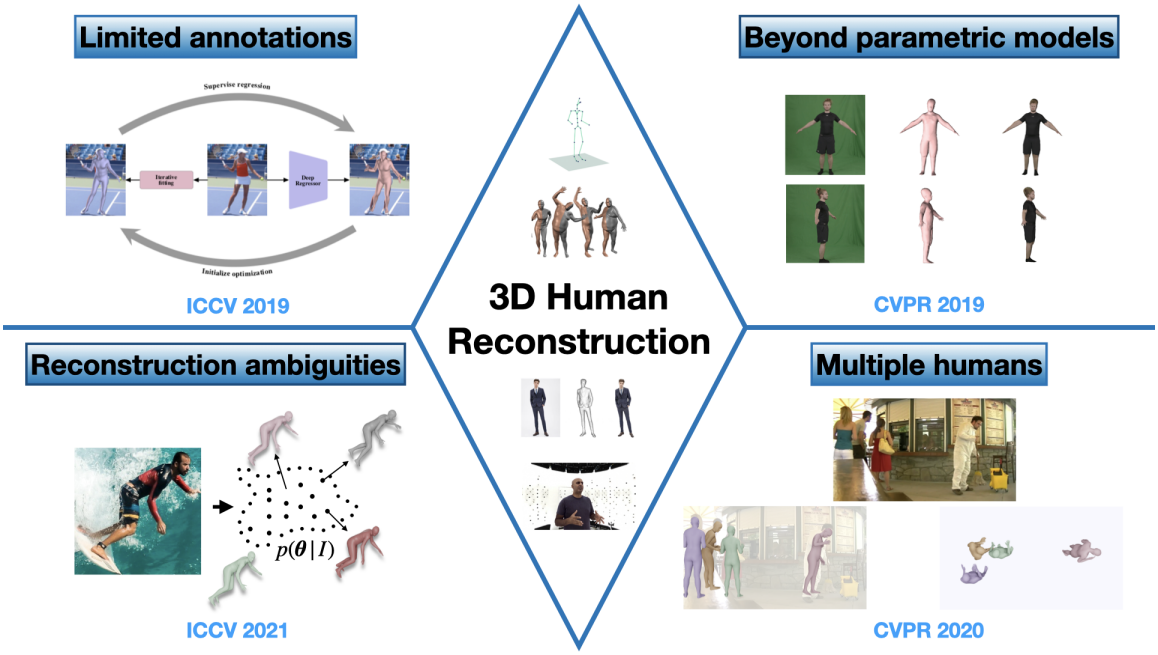

Figure 1: Outline of the thesis. This dissertation explores 4 important problems in 3D human reconstruction: The limited availability of annotated data, learning deformations beyond standard parametric models, modeling the reconstruction ambiguities, and scaling methods to work for scenes with more than one person.

Understanding humans has been one of the most important and well-studied topics in Computer Vision. Researchers have devised methods for detecting humans in images [30, 29, 106], segmenting them from the background [36, 13] estimating their pose [136, 91], and modeling their interactions with the environment. Human understanding also extends beyond just static images to tasks such as motion prediction [8, 82], tracking [145, 23], re-identification [24, 156] and action recognition [141, 44].

This dissertation focuses on estimating the pose and shape of humans in 3D. Traditionally, most methods attempt to reason about humans in 2D, by predicting their silhouettes [13] or 2D pose in the form of a skeleton [9]. This is indeed easier, and convolutional neural networks have been proven really successful for this task. However, constraining our reasoning only in 2D is sub-optimal since humans are 3-dimensional entities.

Estimating 3D human pose from images has been a long-standing problem. Early approaches such as the seminal works of Lee and Chen [68] and Taylor [130] assume a kinematic chain structure of the human body and then attempt to predict the 3D pose from 2D landmarks using a combination of geometric constraints and hand-crafted priors. More recent works have also made significant advances in the frontier of skeleton-based 3D human pose estimation from single images, with many approaches achieving impressive results [83, 87, 111, 126, 133, 160]. Although these skeleton-based methods have boosted the interest for 3D human pose estimation, in this thesis we will focus specifically on model-based pose estimation. Approaches in this category consider a parametric model of the human body, like SMPL [78] and the goal is to estimate the full body 3D pose and shape of a person given an input signal, *e.g.* an image containing a person.

A standard learning-based pipeline for reconstructing 3D humans would be to collect a dataset with model parameter annotations and use this to supervise the neural network output directly. However such data is generally not available. Most datasets with 3D annotations are typically captured in studio environments and use special equipment or multiple cameras [118, 40, 84], while other datasets captured in outdoor settings [140] do not have enough pose and subject diversity. Consequently, models trained only on these datasets fail to generalize in challenging new scenes. There exist however large-scale datasets like COCO [73] and MPII [3] with large scene variation, but unfortunately they come with only 2D annotations. Thus it is crucial to find efficient ways of exploiting all the available data to improve the performance of our trained models.

At the same time, estimating 3D human pose from 2D evidence is an inherently ambiguous problem. There are three main sources of ambiguities. First, we often have truncations that happen when part of the body is not observed at all in the image. Next, there are occlusions. These occlusions can be from other people, objects, the environment, or even self-occlusions. However, the most common ambiguities that are almost always present in single-image reconstruction are depth ambiguities. Since we only observe the 2D projection

of the different body parts, it is very hard to recover the relative depths of the different body parts. In fact, Lee and Chen [68] showed that without pose priors, the problem of lifting 2D landmarks to 3D has an exponential number of plausible solutions in the worst case.

Outline and contributions. This dissertation will present our contributions towards automating 3D human reconstruction from images. These are outlined in Figure 1. Chapter 2 explores the use of different output representations for human mesh recovery and discusses their advantages, as shown in [64]. We also discuss how our model can be used for learning deformations beyond standard parametric models and capture details such as hair or clothing. Chapter 3 focuses on the limited availability of annotated 3D data. More specifically, in this chapter we describe SPIN [63] that proposes to leverage a close collaboration between regression and optimization methods to train a network for 3D human pose and shape estimation from images. Chapter 3 presents ProHMR [65], our work on modeling the ambiguities in 3D human reconstruction . In ProHMR we propose to learn a mapping from an input image to a distribution of plausible poses, and we demonstrate the usefulness of our probabilistic framework for solving a variety of downstream tasks such as body model fitting. Last, Chapter 5 moves beyond single-person 3D pose estimation and shows how we can scale our methods to work on scenes with multiple humans. We present our work on reconstructing scenes with multiple people [45] that achieves this by using 2 novel geometric losses that encourage the coherency of the holistic result.

Future directions. This thesis focused on reconstructing one or more humans in from image data. However, our work [64, 63, 65, 45] considered humans in isolation, independently from their surroundings. It is important to note that our environment defines a set of affordances that dictate how we live and move inside it. Conversely, humans also constrain their environment. Typically, reconstructing general scenes is harder than reconstructing humans, and an accurate human reconstruction should give us a strong signal about their surroundings. There is already work on multi-person modeling or human-scene interactions

[154, 153, 45, 88, 127, 34, 155], but previous work did not go beyond imposing geometric constraints, such as collision avoidance. Human interactions are much more complex than modeling simple spatial proximities. Often times there is additional semantic context involved, and to take reconstruction and scene understanding to the next level we need to take it into account. Chairs are an example of an object that we interact with in a particular way. People normally sit rather than stand on them. Similarly when people grab objects such as forks, knives, hammers, scissors Thus the next important step is to tackle the problem of modeling humans jointly with the environment, both at the level of reconstructing people in scenes and at a finer level where we model hand-object interactions. These 2 problems have a lot of applications in augmented and virtual reality. The key insight here is that we should reason about humans and their environment jointly. We need to incorporate semantics and move beyond purely geometric methods because scene understanding and context are more important.

In parallel, another observation is that in this thesis we only considered static images as input and did not explore the use of time as an additional source of information. Indeed, there are works that make use of time for human pose estimation or forecasting [54, 62, 82, 102], but they operate on short-time intervals. Short-term motion prediction is all about modeling the biomechanics of human bodies. Predicting human motion long term in the future, more than a couple of seconds, requires reasoning about intent and the context we are currently in. In fact this is still an open problem, and has a lot of applications in autonomous driving and robots that actively interact with people.

CHAPTER 2 : Convolutional Mesh Regression for Single-Image Human Shape Reconstruction

2.1 Introduction

Figure 2: Summary of our approach. Given an input image we directly regress a 3D shape with graph convolutions. Optionally, from the 3D shape output we can regress the parametric representation of a body model.

Analyzing humans from images goes beyond estimating the 2D pose for one person [91, 143] or multiple people [11, 103], or even estimating a simplistic 3D skeleton [83, 87]. Our understanding relies heavily on being able to properly reconstruct the complete 3D pose and shape of people from monocular images. And while this problem is well addressed in settings with multiple cameras [38, 51], the excessive ambiguity, the limited training data, and the wide range of imaging conditions make this task particularly challenging in the monocular case.

Traditionally, optimization-based approaches [7, 67, 153] have offered the most reliable solution for monocular pose and shape recovery. However, the slow running time, the

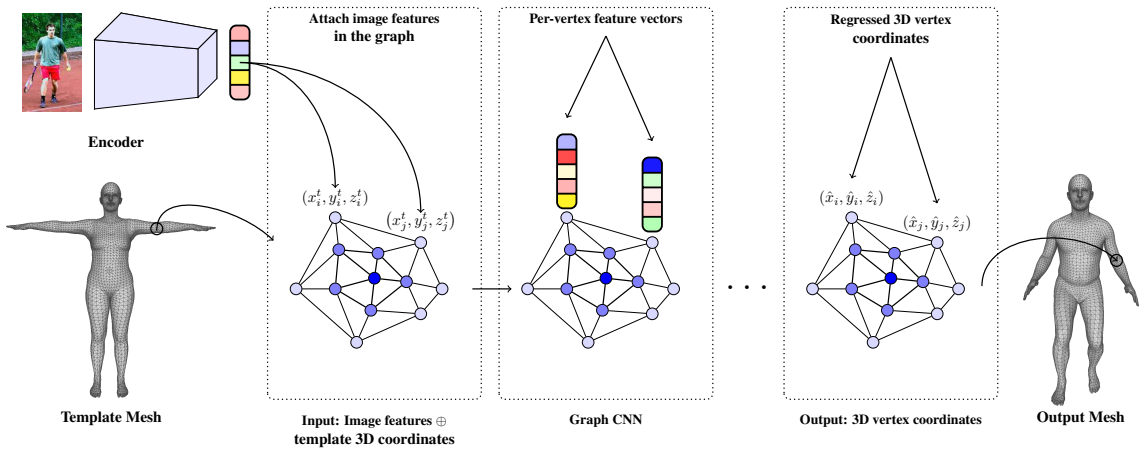

Figure 3: Overview of proposed framework. Given an input image, an image-based CNN encodes it in a low dimensional feature vector. This feature vector is embedded in the graph defined by the template human mesh by attaching it to the 3D coordinates (x_i^t, y_i^t, z_i^t) of every vertex i. We then process it through a series of Graph Convolutional layers and regress the 3D vertex coordinates $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ of the deformed mesh.

reliance on a good initialization and the typical failures due to bad local minima have recently shifted the focus to learning-based approaches [52, 67, 92, 101, 129, 137], that regress pose and shape directly from images. The majority of these works investigate what is the most reliable modality to regress pose and shape from. Surface landmarks [67], pose keypoints and silhouettes [101], semantic part segmentation [92], or raw pixels [52] have all been considered as the network input. And while the input representation topic has received much debate, all the above approaches nicely conform to the SMPL model [78] and use its parametric representation as the regression target of choice. However, taking the decision to commit to a particular parametric space can be quite constraining itself. For example, SMPL is not modeling hand pose or facial expressions [51, 112]. What is even more alarming is that the model parameter space might not be appropriate as a regression target. In the case of SMPL, the pose space is expressed in the form of 3D rotations, a pretty challenging prediction target [80, 89]. Depending on the selected 3D rotation representation (e.g., axis angle, rotation matrices, quaternions), we might face problems of periodicity, non-minimal representation, or discontinuities, which complicate the prediction task. And in fact, all the above model-based approaches underperfom in pose estimation metrics compared to

approaches regressing a less informative, yet more accurate, 3D skeleton through 3D joint regression [18, 83, 99, 125].

In this work, we propose to take a more hybrid route towards pose and shape regression. Even though we preserve the template mesh introduced by SMPL, we do not directly regress the SMPL model parameters. Instead, our regression target is the 3D mesh vertices. Considering the excessive number of vertices of the mesh, if addressed naively, this would be a particular heavy burden for the network. Our key insight though, is that this task can be effectively and efficiently addressed by the introduction of a Graph-CNN. This architecture enables the explicit encoding of the mesh structure in the network, and leverages the spatial locality of the graph. Given a single image (Figure 3), any typical CNN can be used for feature extraction. The extracted features are attached on the vertex coordinates of the template mesh, and the processing continues on the graph structure defined for the Graph-CNN. In the end, each vertex has as target its 3D location in the deformed mesh. This allows us to recover the complete 3D geometry of the human body without explicitly committing to a pre-specified parametric space, leaving the mesh topology as the only hand-designed choice. Conveniently, after estimating the 3D position for each vertex, if we need our prediction to conform to a specific model, we can regress its parameters quite reliably from the mesh geometry (Figure 2). This enables a more hybrid usage for our approach, making it directly comparable to model-based approaches. Furthermore, our graph-based processing is largely agnostic to the input type, allowing us to attach features extracted from RGB pixels [52], semantic part segmentation [92], or even from dense correspondences [33]. In all these cases we demonstrate that our approach outperforms the baselines that regress model parameters directly from the same type of features, while overall we achieve state-of-the-art pose estimation results among model-based baselines.

Our contributions can be summarized as follows:

- We reformulate the problem of human pose and shape estimation in the form of regressing the 3D locations of the mesh vertices, to avoid the difficulties of direct

model parameter regression.

- We propose a Graph CNN for this task which encodes the mesh structure and enables the convolutional mesh regression of the 3D vertex locations.

- We demonstrate the flexibility of our framework by considering different input representations, always outperforming the baselines regressing the model parameters directly.

- We achieve state-of-the-art results among model-based pose estimation approaches.

2.2 Related work

There is rich recent literature on 3D pose estimation in the form of a simplistic body skeleton, e.g., [18, 71, 79, 83, 87, 99, 100, 109, 110, 125, 132, 133, 134, 159, 161]. However, in this Section, we focus on the more relevant works recovering the full shape and pose of the human body.

Optimization-based shape recovery: Going beyond a simplistic skeleton, and recovering the full pose and shape, initially, the most successful approaches followed optimization-based solutions. The work of Guan *et al.* [31] relied on annotated 2D landmarks and optimized for the parameters of the SCAPE parametric model that generated a mesh optimally matching this evidence. This procedure was made automatic with the SMPLify approach of Bogo *et al.* [7], where the 2D keypoints where localized through the help of a CNN [103]. Lassner *et al.* [67] included auxiliary landmarks on the surface of the human body, and additionally considered the estimated silhouette during the fitting process. Zanfir *et al.* [153] similarly optimized for consistency of the reprojected mesh with semantic parts of the human body, while extending the approach to work for multiple people as well. Despite the reliable results obtained, the main concern for approaches of this type is that they pose a complicated non-convex optimization problem. This means that the final solution is very sensitive to the initialization, the optimization can get stuck in local minima, and simultaneously the whole procedure can take several minutes to complete. These draw-

backs have motivated the increased interest in learning-based approaches, like ours, where the pose and shape are regressed directly from images.

Direct parametric regression: When it comes to pose and shape regression, the vast majority of works adopt the SMPL parametric model and consider regression of pose and shape parameters. Lassner *et al*. [67] detect 91 landmarks on the body surface and use a random forest to regress the SMPL model parameters for pose and shape. Pavlakos *et al*. [101] rely on a smaller number of keypoints and body silhouettes to regress the SMPL parameters. Omran *et al*. [92] follow a similar strategy but use a part segmentation map as the intermediate representation. On the other hand, Kanazawa *et al*. [52] attempt to regress the SMPL parameters directly from images, using a weakly supervised approach relying on 2D keypoint reprojection and a pose prior learnt in an adversarial manner. Tung *et al*. [137] present a self-supervised approach for the same problem, while Tan *et al*. [129] rely on weaker supervision in the form of body silhouettes. The common theme of all these works is that they have focused on using the SMPL parameter space as a regression target. However, the 3D rotations involved as the pose parameters have created issues in the regression (e.g., discontinuities or periodicity) and typically underperform in terms of pose estimation compared to skeleton-only baselines. In this work, we propose to take an orthogonal approach to them, by regressing the 3D location of the mesh vertices by means of a Graph-CNN. Our approach is transparent to the type of the input representation we use, since the flexibility of the Graph network allows us to consider different types of input representations employed in prior work, like semantic part-based features [92], features extracted directly from raw pixels [52], or even dense correspondences [33].

Nonparametric shape estimation: Recently, nonparametric approaches have also been proposed for pose and shape estimation. Varol *et al*. [138] use a volumetric reconstruction approach with a voxel output. Different tasks are simultaneously considered for intermediate supervision. Jackson *et al*. [42] also propose a form of volumetric reconstruction by extending their recent face reconstruction network [41] to work for full body images. The

main drawback of these approaches adopting a completely nonparametric route, is that even if they recover an accurate voxelized sculpture of the human body, there is none or very little semantic information captured. In fact, to recover the body pose, we need to explicitly perform an expensive body model fitting step using the recovered voxel map, as done in [138]. In contrast to them, we retain the SMPL mesh topology, which allows us to get dense semantic correspondences of our 3D prediction with the image, and in the end we can also easily regress the model's parameters given the vertices 3D location.

Graph CNNs: Wang *et al.* [142] use a Graph CNN to reconstruct meshes of objects from images by deforming an initial ellipsoid. However, mesh reconstruction of arbitrary objects is still an open problem, because shapes of objects even in the same class, e.g., chairs, do not have the same genus. Contrary to generic objects, arbitrary human shapes can be reconstructed as continuous deformations of a template model. In fact, recently there has been a lot of research in applying Graph Convolutions for human shape applications. Verma *et al.* [139] propose a new data-driven Graph Convolution operator with applications on shape analysis. Litany *et al.* [74] use a Graph VAE to learn a latent space of human shapes, that is useful for shape completion. Ranjan *et al.* [105] use a mesh autoencoder network to recover a latent representation of 3D human faces from a series of meshes. The main difference of our approach is that we do not aim to learn a generative shape model from 3D shapes, but instead perform single-image shape reconstruction; the input to our network is an image, not a 3D shape. The use of a Graph CNN alone is not new, but we consider as a contribution the insight that Graph CNNs provide a very natural structure to enable our hybrid approach. They assist us in avoiding the SMPL parameter space, which has been reported to have issues with regression [83, 101], while simultaneously allowing the explicit encoding of the graph structure in the network, so that we can leverage spatial locality and preserve the semantic correspondences.

2.3 Technical approach

In this Section we present our proposed approach for predicting 3D human shape from a single image. First, in Subsection 2.3.1 we briefly describe the image-based architecture that we use as a generic feature extractor. In Subsection 2.3.2 we focus on the core of our approach, the Graph CNN architecture that is responsible to regress the 3D vertex coordinates of the mesh that deforms to reconstruct the human body. Then, Subsection 2.3.3 describes a way to combine our non-parametric regression with the prediction of SMPL model parameters. Finally, Subsection 5.3.5 focuses on important implementation details.

2.3.1 Image-based CNN

The first part of our pipeline consists of a typical image-based CNN following the ResNet-50 architecture [37]. From the original design we ignore the final fully connected layer, keeping only the 2048-D feature vector after the average pooling layer. This CNN is used as a generic feature extractor from the input representation. To demonstrate the flexibility of our approach, we experiment with a variety of inputs, i.e., RGB images, part segmentation and DensePose input [33]. For RGB images we simply use raw pixels as input, while for the other representations, we assume that another network [33], provides us with the predicted part segmentation or DensePose. Although we present experiments with a variety of inputs, our goal is not to investigate the effect of the input representation, but rather we focus our attention on the graph-based processing that follows.

2.3.2 Graph CNN

At the heart of our approach, we propose to employ a Graph CNN to regress the 3D coordinates of the mesh vertices. For our network architecture we draw inspiration from the work of Litany *et al*. [74]. We start from a template human mesh with N vertices as depicted in Figure 3. Given the 2048-D feature vector extracted by the generic image-based network, we attach these features to the 3D coordinates of each vertex in the template mesh. From a high-level perspective, the Graph CNN uses as input the 3D coordinates of each vertex along with the input features and has the goal of estimating the 3D coordinates

for each vertex in the output, deformed mesh. This processing is performed by a series of Graph Convolution layers.

For the graph convolutions we use the formulation from Kipf *et al*. [61] which is defined as:

$$Y = \tilde{A}XW \tag{2.1}$$

where $X \in \mathbb{R}^{N \times k}$ is the input feature vector, $W \in \mathbb{R}^{k \times \ell}$ the weight matrix and and $\tilde{A} \in \mathbb{R}^{N \times N}$ is the row-normalized adjacency matrix of the graph. Essentially, this is equivalent to performing per-vertex fully connected operations followed by a neighborhood averaging operation. The neighborhood averaging is essential for producing a high quality shape because it enforces neighboring vertices to have similar features, and thus the output shape is smooth. With this design choice we observed that there is no need of a smoothness loss on the shape, as for example in [55]. We also experimented with the more powerful graph convolutions proposed in [139] but we did not observe quantitative improvement in the results, so we decided to keep our original and simpler design choice.

For the graph convolution layers, we make use of residual connections as they help in speeding up significantly the training and also lead in higher quality output shapes. Our basic building block is similar to the Bottleneck residual block [37] where 1×1 convolutions are replaced by per-vertex fully connected layers and Batch Normalization [39] is replaced by Group Normalization [146]. We noticed that Batch Normalization leads to unstable training and poor test performance, whereas with no normalization the training is very slow and the network can get stuck at local minima and collapse early during training.

Besides the 3D coordinates for each vertex, our Graph CNN also regresses the camera parameters for a weak-perspective camera model. Following Kanazawa *et al*. [52], we predict a scaling factor s and a 2D translation vector $\mathbf{t}$. Since the prediction of the network is already on the camera frame, we do not need to regress an additional global camera rotation. The camera parameters are regressed from the graph embedding and not from the image features

directly. This way we get a much more reliable estimate that is consistent with the output shape.

Regarding training, let $\hat{Y} \in \mathbb{R}^{N\times 3}$ be the predicted 3D shape, Y the ground truth shape and X the ground truth 2D keypoint locations of the joints. From our 3D shape we can also regress the location for the predicted 3D joints $\hat{J}_{3D}$ employing the same regressor that the SMPL model is using to recover joints from vertices. Given these 3D joints, we can simply project them on the image plane, $\hat{X} = s\mathbf{\Pi}(\hat{J}_{3D}) + \mathbf{t}$. Now, we train the network using two forms of supervision. First, we apply a per-vertex L_1 loss between the predicted and ground truth shape, i.e.,

$$\mathcal{L}_{shape} = \sum_{i=1}^{N} ||\hat{Y}_i - Y_i||_1. \tag{2.2}$$

Empirically we found that using L_1 loss leads to more stable training and better performance than L_2 loss. Additionally, to enforce image-model alignment, we also apply an L_1 loss between the projected joint locations and the ground truth keypoints, i.e.,

$$\mathcal{L}_J = \sum_{i=1}^{M} ||\hat{X}_i - X_i||_1. \tag{2.3}$$

Finally, our complete training objective is:

$$\mathcal{L} = \mathcal{L}_{shape} + \mathcal{L}_J. \tag{2.4}$$

This form of supervised training requires us to have access to images with full 3D ground truth shape. However, based on our empirical observation, it is not necessary for all the training examples to come with ground truth shape. In fact, following the observation of Omran *et al.* [92], we can leverage additional images that provide only 2D keypoint ground truth. In these cases, we simply ignore the first term of the previous equation and train only with the keypoint loss. We have included evaluation under this setting of weaker supervision

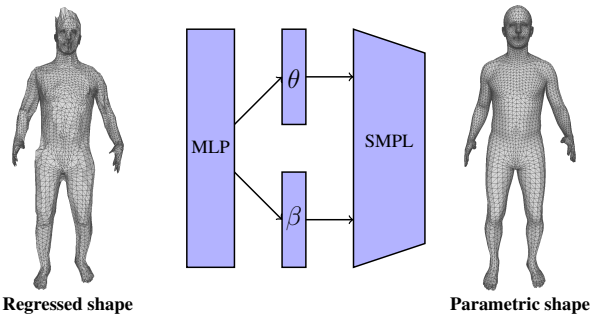

Figure 4: Predicting SMPL parameters from regressed shape. Given a regressed 3D shape from the network of Figure 3, we can use a Multi-Layer Perceptron (MLP) to regress the SMPL parameters and produce a shape that is consistent with the original non-parametric shape

in the Sup. Mat.

2.3.3 SMPL from regressed shape

Although we demonstrate that non-parametric regression is an easier task for the network, there are still many applications where a parametric representation of the human body can be very useful (e.g., motion prediction). In this Subsection, we present a straightforward way to combine our non-parametric prediction with a particular parametric model, i.e., SMPL. To achieve this goal, we train another network that regresses pose (θ) and shape (β) parameters of the SMPL parametric model given the regressed 3D shape as input. The architecture of this network can be very simple, i.e., a Multi-Layer Perceptron (MLP) [113] for our implementation. This network is presented in Figure 4 and the loss function for training is:

$$\mathcal{L} = \mathcal{L}_{shape} + \mathcal{L}_J + \mathcal{L}_\theta + \lambda\mathcal{L}_\beta. \tag{2.5}$$

Here, $\mathcal{L}_{shape}$ and $\mathcal{L}_J$ are the losses on the 3D shape and 2D joint reprojection as before, while $\mathcal{L}_\theta$ and $\mathcal{L}_\beta$ are L_2 losses on the SMPL pose and shape parameters respectively.

As observed by previous works, e.g., [101, 83], it is challenging to regress the pose parameters θ, which represent 3D rotations in the axis-angle representation. To avoid this, we followed the strategy employed by Omran *et al.* [92]. More specifically, we convert the parameters from axis-angle representation to a rotation matrix representation using the Rodrigues

formula, and we set the output of our network to regress the elements of the rotation matrices. To ensure that the output is a valid rotation matrix we project it to the manifold of rotation matrices using the differentiable SVD operation. Although this representation does not explicitly improve our quantitative results, we observed faster convergence during training, so we selected it as a more practical option.

2.3.4 Implementation details

An important detail regarding our Graph CNN is that we do not operate directly on the original SMPL mesh, but we first subsample it by a factor of 4 and then upsample it again to the original scale using the technique described in [105]. This is essentially performed by precomputing downsampling and upsampling matrices D and U and left-multiply them with the graph every time we need to do resampling. This downsampling step helps to avoid the high redundancy in the original mesh due to the spatial locality of the vertices, and decrease memory requirements during training.

Regarding the training of the MLP, we employ a 2-step training procedure. First we train the network that regresses the non-parametric shape and then with this network fixed we train the MLP that predicts the SMPL parameters. We also experimented with training them end-to-end but we observed a decrease in the performance of the network for both the parametric and non-parametric shape.

2.4 Empirical evaluation

In this Section, we present the empirical evaluation of our approach. First, we discuss the datasets we use in our evaluation (Subsection 5.4.1), then we provide training details for our pipeline (Subsection 2.4.2), and finally, the quantitative and qualitative evaluation (Subsection 2.4.3) follows.

2.4.1 Datasets

We employ two datasets that provide 3D ground truth for training, Human3.6M [40] and UP-3D [67], while we evaluate our approach on Human3.6M and the LSP dataset [46].

Human3.6M: It is an indoor 3D pose dataset including subjects performing activities like Walking, Eating and Smoking. We use the subjects S1, S5, S6, S7 and S8 for training, and keep the subjects S9 and S11 for testing. We present results for two popular protocols (P1 and P2, as defined in [52]) and two error metrics (MPJPE and Reconstruction error, as defined in [161]).

UP-3D: It is a dataset created by applying SMPLify [7] on natural images of humans and selecting the successful fits. We use the training set of this dataset for training.

LSP: It is a 2D pose dataset, including also segmentation annotations provided by Lassner *et al*. [67]. We use the test set of this dataset for evaluation.

2.4.2 Training details

For the image-based encoder, we use a ResNet50 model [37] pretrained on ImageNet [19]. All other network components (Graph CNN and MLP for SMPL parameters) are trained from scratch. For our training, we use the Adam optimizer, and a batch size of 16, with the learning rate set to $3e-4$. We did not use learning rate decay. Training with data only from Human3.6M lasts for 10 epochs, while mixed training with data from Human3.6M and UP-3D requires training for 25 epochs, because of the greater image diversity. To train the MLP that regresses SMPL parameters from our predicted shape, we use 3D shapes from Human3.6M and UP-3D. Finally, for the models using Part Segmentation or DensePose [33] predictions as input, we use the pretrained network of [33] to provide the corresponding predictions.

2.4.3 Experimental analysis

Regression target: For the initial ablative study, we aim to investigate the importance of our mesh regression for 3D human shape estimation. To this end, we focus on the Human3.6M dataset and we evaluate the regressed shape through 3D pose accuracy. First, we evaluate the direct regression of the 3D vertex coordinates, in comparison to generating the 3D shape implicitly through regression of the SMPL model parameters directly from

Method	MPJPE	Reconst. Error
SMPL Parameter Regression [52]	-	77.6
Mesh Regression (FC)	200.8	105.8
Mesh Regression (Graph)	**102.1**	69.0
Mesh Regression (Graph + SMPL)	113.2	**61.3**

Table 1: Evaluation of 3D pose estimation in Human3.6M (Protocol 2). The numbers are MPJPE and Reconstruction errors in mm. Our graph-based mesh regression (with or without SMPL parameter regression) is compared with a method that regresses SMPL parameters directly, as well as with a naive mesh regression using fully connected (FC) layers instead of a Graph-CNN.

Input	Regression Type	MPJPE		Reconst. Error	
		P1	P2	P1	P2
RGB	Parameter [52]	88.0	-	58.1	56.8
	Mesh (Graph + SMPL)	**74.7**	71.9	**51.9**	**50.1**
Parts	Parameter [92]	-	-	-	59.9
	Mesh (Graph + SMPL)	80.4	77.4	56.1	**53.3**
DP[33]	Parameter [52]	82.7	79.5	57.8	54.9
	Mesh (Graph + SMPL)	**78.9**	**74.2**	**55.3**	**51.0**

Table 2: Comparison of direct SMPL parameter regression versus our proposed mesh regression on Human3.6M (Protocol 1 and 2) for different input representations. The numbers are mean 3D joint errors in mm, with and without Procrustes alignment (Rec. Error and MPJPE respectively). Our results are computed after regressing SMPL parameters from our non-parametric shape. Number are taken from the respective works, except for the baseline of [52] on DensePose images, which is evaluated by us.

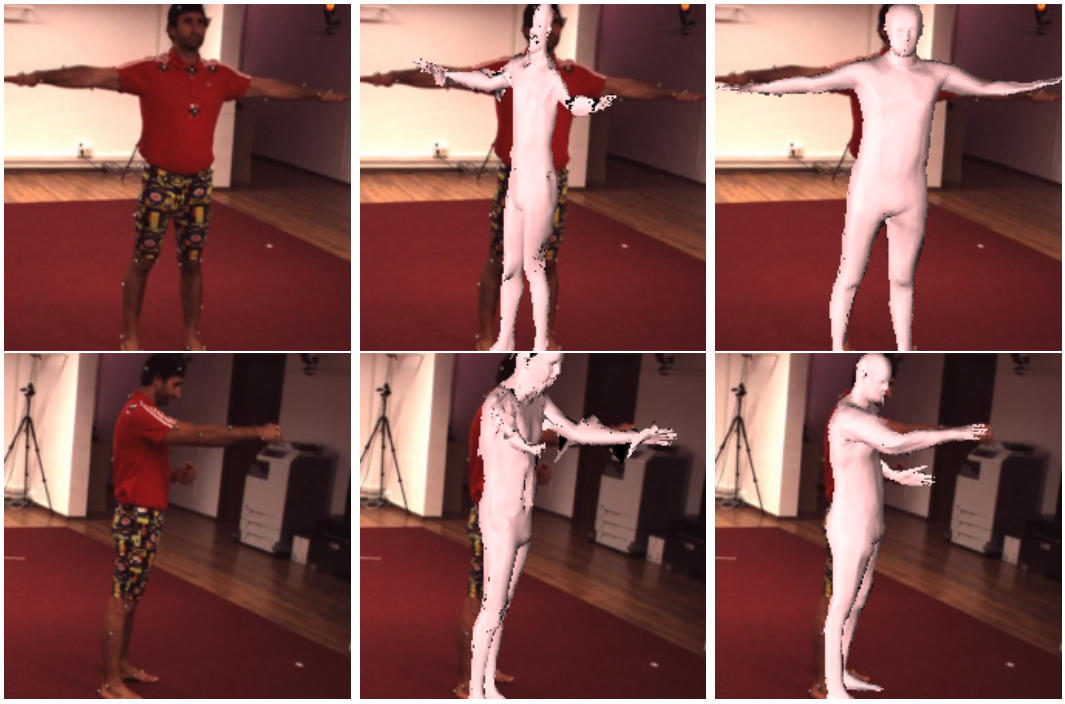

Image FC Graph CNN

Figure 5: Using a series of fully connected (FC) layers to regress the vertex 3D coordinates severely complicates the regression task and gives non-smooth meshes, since the network cannot leverage directly the topology of the graph.

images. The most relevant baseline in this category is the HMR method of [52]. In Table 1, we present the comparison of this approach (*SMPL parameter regression*) with our non-parametric shape regression (*Mesh Regression - (Graph)*). For a more fair comparison, we also include our results for the MLP that regresses SMPL parameters using our non-parametric mesh as input (*Mesh Regression - (Graph + SMPL)*). In both cases, we outperform the strong baseline of [52], which demonstrates the benefit of estimating a more flexible non-parametric regression target, instead of regressing the model parameters in one shot.

Beyond the regression target, one of our contributions is also the insight that the task of regressing 3D vertex coordinates can be greatly simplified when a Graph CNN is used for the prediction. To investigate this design choice, we compare it with a naive alternative that regresses vertex coordinates with a series of fully connected layers on top of our image-

Input	Output shape	MPJPE P1	MPJPE P2	Reconst. Error P1	Reconst. Error P2
RGB	Non parametric	75.0	72.7	51.2	49.3
	Parametric	74.7	71.9	51.9	50.1
Parts	Non parametric	78.0	73.4	54.6	50.6
	Parametric	80.4	77.4	56.1	53.3
DP[33]	Non parametric	78.0	72.3	55.3	50.3
	Parametric	78.9	74.2	55.3	51.0

Table 3: Comparison on Human3.6M (Protocol 1 and 2) of our non-parametric mesh with the SMPL parametric mesh regressed from our shape. Numbers are 3D joint errors in mm. The performance of the two baselines is similar.

Method	Reconst. Error
Lassner *et al.* [67]	93.9
SMPLify [7]	82.3
Pavlakos *et al.* [101]	75.9
NBF [92]	59.9
HMR [52]	56.8
Ours	**50.1**

Table 4: Comparison with the state-of-the-art on Human3.6M (Protocol 2). Numbers are Reconstruction errors in mm. Our approach outperforms the previous baselines.

based encoder (*Mesh Regression - (FC)*). This design clearly underperforms compared to our Graph-based architecture, demonstrating the importance of leveraging the mesh structure through the Graph CNN during the regression. The benefit of graph-based processing is demonstrated also qualitatively in Figure 5.

Input representation: For the next ablative, we demonstrate the effectiveness of our mesh regression for different types of input representations, i.e., RGB images, Part Segmentation as well as DensePose images [33]. The complete results are presented in Table 2. The RGB model is trained on Human3.6M + UP-3D whereas the two other models only on Human3.6M. For every input type, we compare with state-of-the-art methods [52, 92] and show that our method outperforms them in all setting and metrics. Interestingly, when training only with Human3.6M data, RGB input performs worse than the other representations (Table 1), because of over-fitting. However, we observed that RGB features capture richer information for in-the-wild images, thus we select it for the majority of our experiments.

	FB Seg.		Part Seg.	
	acc.	f1	acc.	f1
SMPLify *oracle* [7]	92.17	0.88	88.82	0.67
SMPLify [7]	91.89	0.88	87.71	0.64
SMPLify on [101]	92.17	0.88	88.24	0.64
Bodynet [138]	92.75	0.84	-	-
HMR [52]	91.67	0.87	87.12	0.60
Ours	91.46	0.87	88.69	0.66

Table 5: Segmentation evaluation on the LSP test set. The numbers are accuracies and f1 scores. We include approaches that are purely regression-based (bottom) and approaches that perform some optimization (post)-processing (top). Our approach is competitive with the state-of-the-art.

SMPL from regressed shape: Additionally we examine the effect of estimating the SMPL model parameters from our predicted 3D shape. As it can be seen in Table 3, adding the SMPL prediction, using a simple MLP on top of our non-parametric shape estimate, only has a small effect in the performance (positive in some cases, negative in others). This means that our regressed 3D shape encapsulates all the important information needed for the model reconstruction, making it very simple to recover a parametric representation (if needed), from our non-parametric shape prediction.

Comparison with the state-of-the-art: Next, we present comparison of our approach with other state-of-the-art methods for 3D human pose and shape estimation. For Human3.6M, detailed results are presented in Table 4, where we outperform the other baselines. We clarify here that different methods use different training data (e.g., Pavlakos *et al*. [101] do not use any Human3.6M data for training, NBF *et al*. [92] uses only data from Human3.6M, while HMR [52] makes use of additional images with 2D ground truth only). However, here we collected the best results reported by each approach on this dataset.

Besides 3D pose, we also evaluate 3D shape through silhouette reprojection on the LSP test set. Our approach outperforms the regression-based approach of Kanazawa *et al*. [52], and is competitive to optimization-based baselines, e.g., [7], which tend to perform better than regression approaches (like ours) in this task, because they explicitly optimize for the

Image Non-parametric Parametric

Figure 6: Examples of erroneous reconstructions. Typical failures can be attributed to challenging poses, severe self-occlusions, or interactions among multiple people.

image-model alignment.

Qualitative evaluation: Figures 6 and 7 present qualitative examples of our approach, including both the non-parametric mesh and the corresponding SMPL mesh regressed using our shape as input. Typical failures can be attributed to challenging poses, severe self-occlusions, as well as interactions among multiple people.

Runtime: On a 2080 Ti GPU, network inference for a single image lasts 33ms, which is effectively real-time.

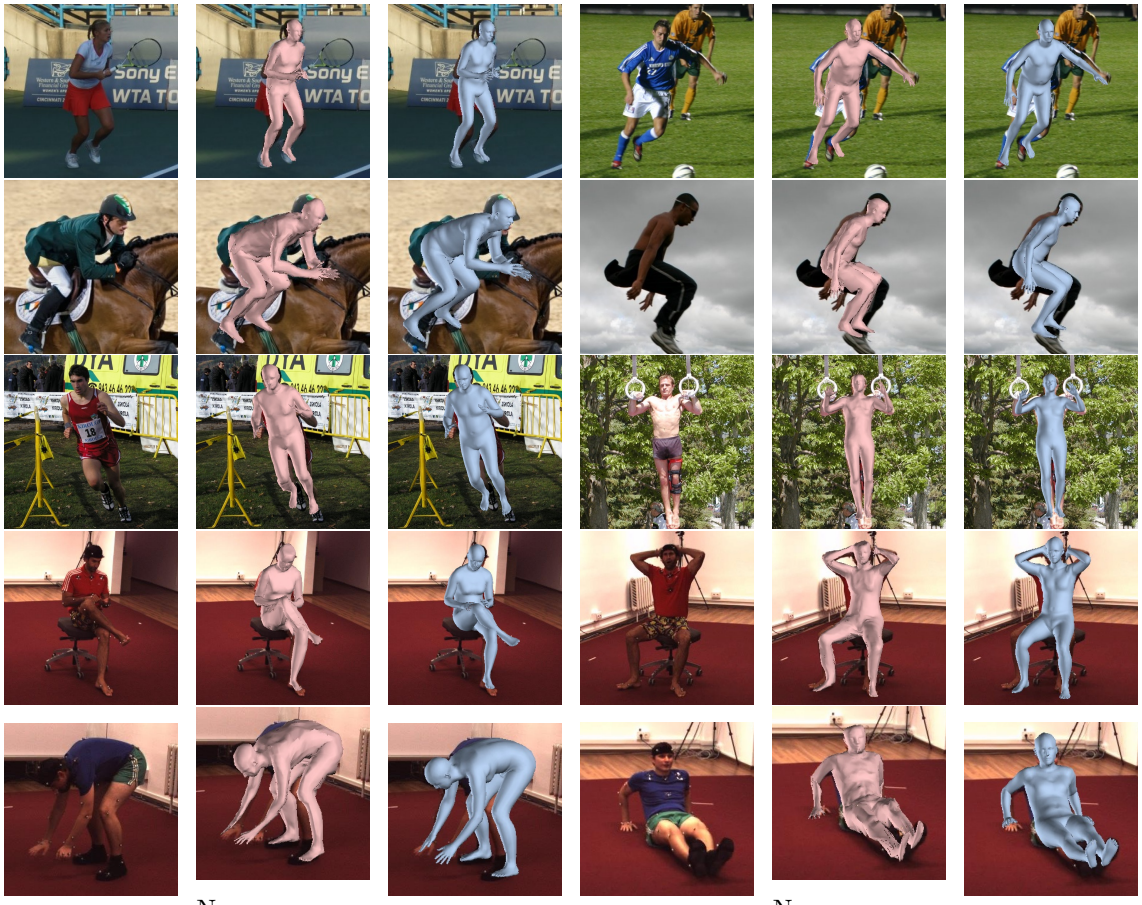

Figure 7: Successful reconstructions of our approach. Rows 1-3: LSP [46]. Rows 4-5: Human3.6M [40]. With light pink color we indicate the regressed non parametric shape and with light blue the SMPL model regressed from the previous shape.

2.5 Summary

The goal of this paper was to address the problem of pose and shape estimation by attempting to relax the heavy reliance of previous works on a parametric model, typically SMPL [78]. While we retain the SMPL mesh topology, instead of directly predicting the model parameters for a given image, our target is to first estimate the locations of the 3D mesh vertices. For this to be achieved effectively, we propose a Graph-CNN architecture, which explicitly encodes the mesh structure and processes image features attached to its vertices. Our convolutional mesh regression outperforms the relevant baselines that regress model parameters directly for a variety of input representations, while ultimately,

it achieves state-of-the-art results among model-based pose estimation approaches. Future work can focus on current limitations (e.g., low resolution of output mesh, missing details in the recovered shape), as well as opportunities that this non-parametric representation provides (e.g., capture aspects missing in many human body models, like hand articulation, facial expressions, clothing and hair).

2.6 Supplementary Material

This supplementary material provides additional details about the training and model architecture as well as some extra evaluations. First, in Section 2.6.1 we provide additional informations about the training process. Then, we investigate the potential of regressing details like hair and clothing with our approach (Section 2.6.2) and extend our empirical evaluation (Section 2.6.3). Finally, in Section 2.6.4 we describe in detail the architecture of the models used in our experiments.

2.6.1 Training Details

During training we randomly rotate and flip the input images, rescale the bounding boxes and also introduce color jittering in the RGB input case. All input images are rescaled to 224×224 before feeding them in the encoder. For mixed training with Human3.6M [40] and UP-3D [67], since UP-3D is significantly smaller than Human3.6M we do not uniformly sample from all images. Instead, first we randomly pick one of the two datasets with probability 0.5, and then we select an image from this dataset uniformly at random. This ensures that an equal number of in-the-wild and indoor examples are included in our batch.

2.6.2 Clothing and hair

As suggested in the main manuscript, our approach should be able to capture details like hair and clothing which are not modeled by typical human body models. To demonstrate this potential, we use the data of Alldieck *et al.* [1] for training and apply our model on hold-out sequences. Interestingly, our regressed mesh (Fig. 8) indeed captures some rough details (e.g., hair bun and shorts). We clarify that these results are purely to demonstrate feasibility and the data is limited for a proper evaluation, but we believe this is a promising

direction for future work.

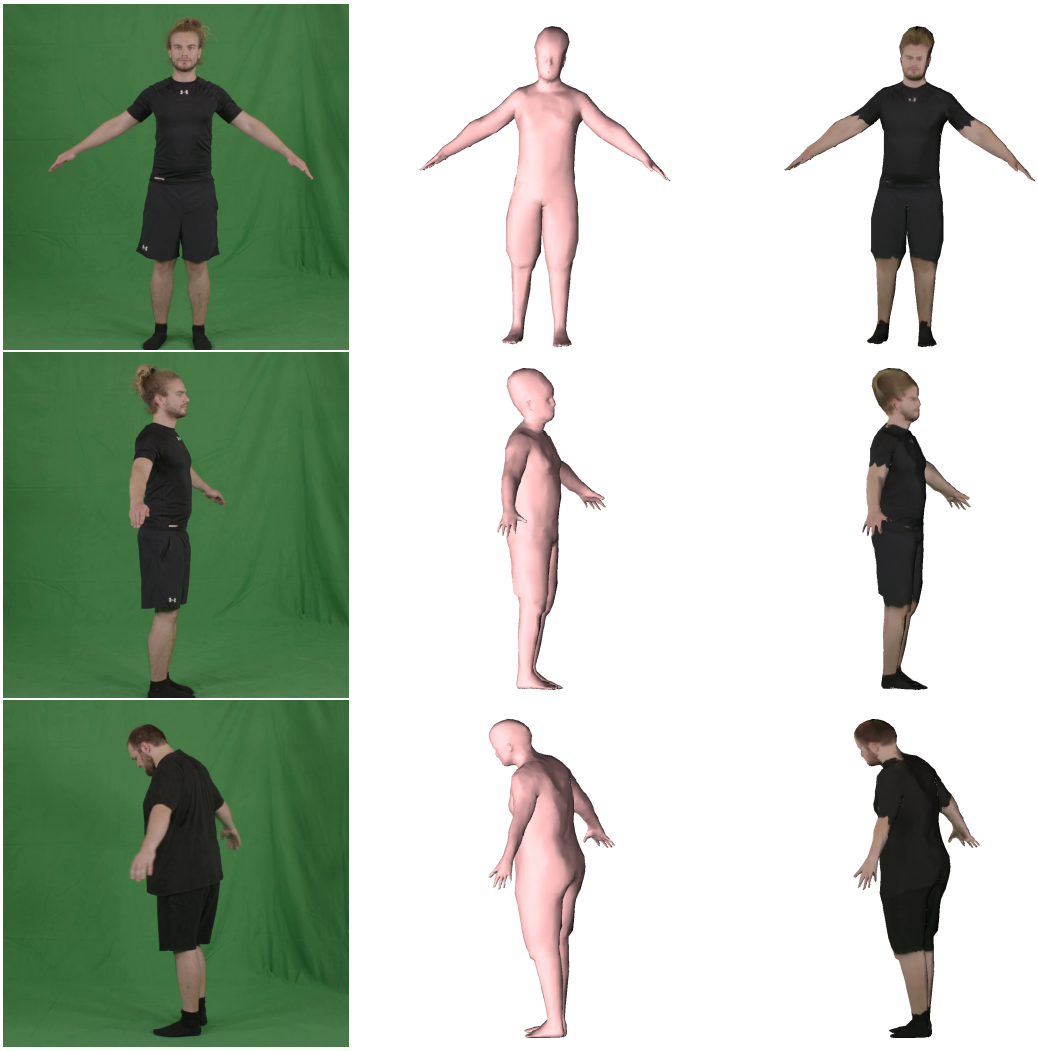

Image Regressed mesh Shape + Texture

Figure 8: Qualitative results on the data of Alldieck *et al.* [1]. From left to right: input image, regressed mesh, ground truth texture applied on the regressed mesh.

2.6.3 Further experimental exploration

In the main manuscript we report results using images from Human3.6M and UP-3D, that provide 3D shape ground truth for training. Although we can expect to have access to such ground truth for indoor datasets (i.e., Human3.6M), in-the-wild examples do not typically come with 3D annotations. Here we demonstrate that our approach is applicable even when these annotations are not available. To this end, we ignore the UP-3D data, and instead

Method	MPJPE	Rec. Error
HMR [52]	88.0	58.1
Ours (H3.6M + LSP,MPII)	78.6	56.6
Ours (H3.6M + UP-3D)	74.7	51.9
Ours (H3.6M + UP-3D + LSP,MPII,COCO)	73.3	51.3

Table 6: Evaluation of our approach on Human3.6M (Protocol 1) for weaker 2D annotations. The numbers are mean joint errors in mm. Training with 2D ground truth only for in-the-wild examples leads to less accurate results compared to our model trained on UP-3D data. However, we are still able to outperform [52] which is trained on significantly more data than our approach. Unsurprisingly, combining our best model with more data that include 2D annotations can further improve our accuracy.

train with images from MPII [3] and LSP [46] that provide only 2D keypoint annotations. Effectively, for these examples, we use only the 2D reprojection loss and not the 3D shape loss. The results for this setting are reported in Table 6. We have also included the results of [52] that trains in a similar setting, i.e., using only 2D annotations for in-the-wild examples. Although the results of this training setting are worse compared to our best model trained on Human3.6M and UP-3D, we are still able to outperform [52] although they use significantly more data (i.e., COCO [73] and MPI-INF-3DHP [87]).

Besides being an alternative to UP-3D data, in-the-wild images with 2D annotations can also be combined with data that have 3D ground truth shape. In this case, we follow a mixed training strategy, using data from Human3.6M, UP-3D (3D ground truth), as well as LSP [46], MPII [3] and COCO [73] (2D ground truth). Unsurprisingly, this combination can further improve performance (Table 6). This last model also achieves the best results on the UP-3D dataset, where we evaluate surface reconstruction accuracy and report the vertex-to-vertex error. Our model achieves 99.8mm error, compared with 102.5mm error of BodyNet [138] and 127.8mm error of Pavlakos *et al*. [101].

Moreover, in Figure 9 and Figure 10 we include qualitative results when using part segmentations and DensePose [33] images respectively as the input representation. We can see that even with non-perfect detections, i.e., parts of the body missing in some difficult poses, the network is still able to correctly regress the 3D shape.

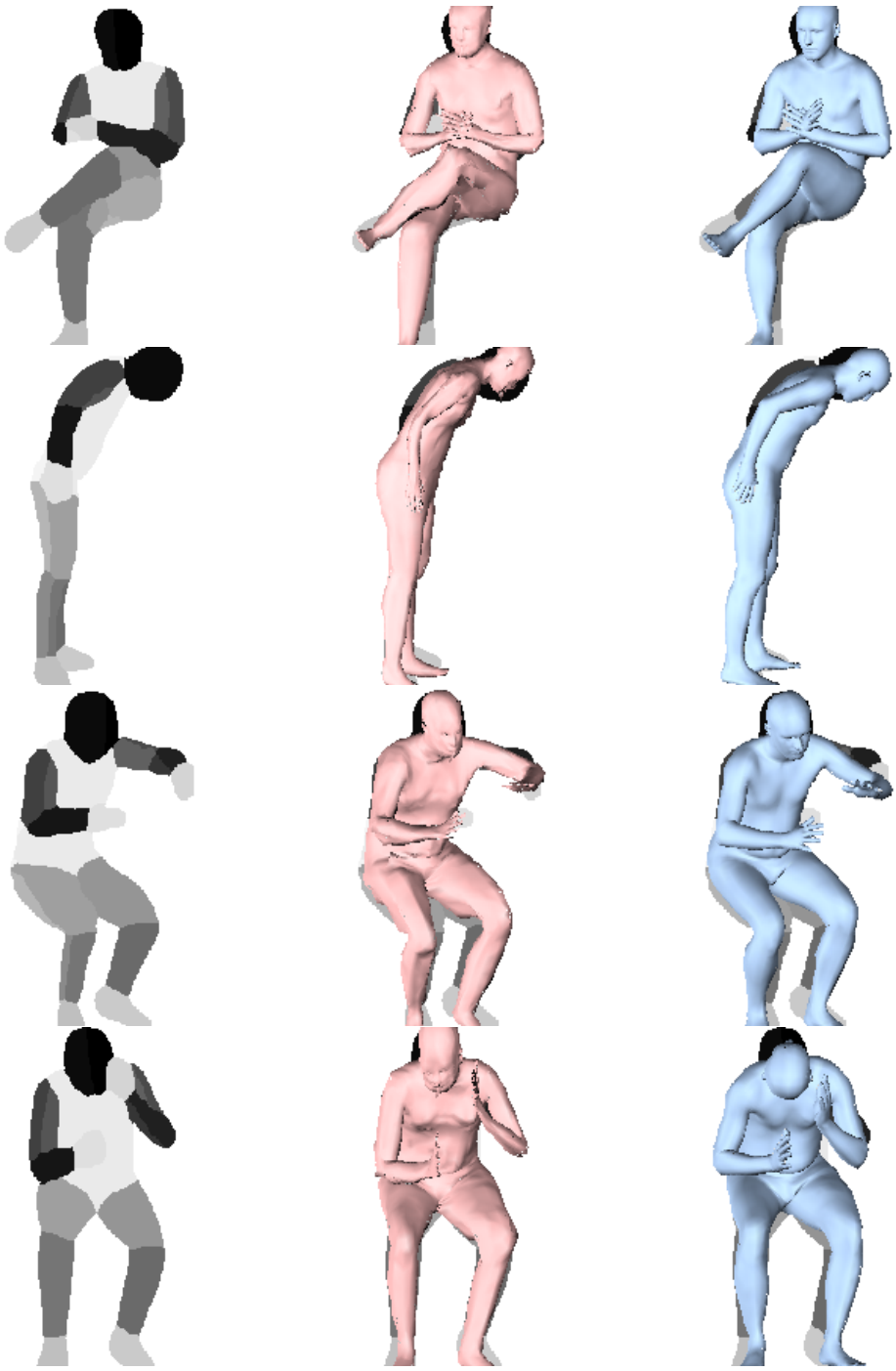

Figure 9: Qualitative results on Human 3.6M with parts segmentation input. With light pink color we indicate the regressed non parametric shape and with light blue the SMPL model regressed from the previous shape.

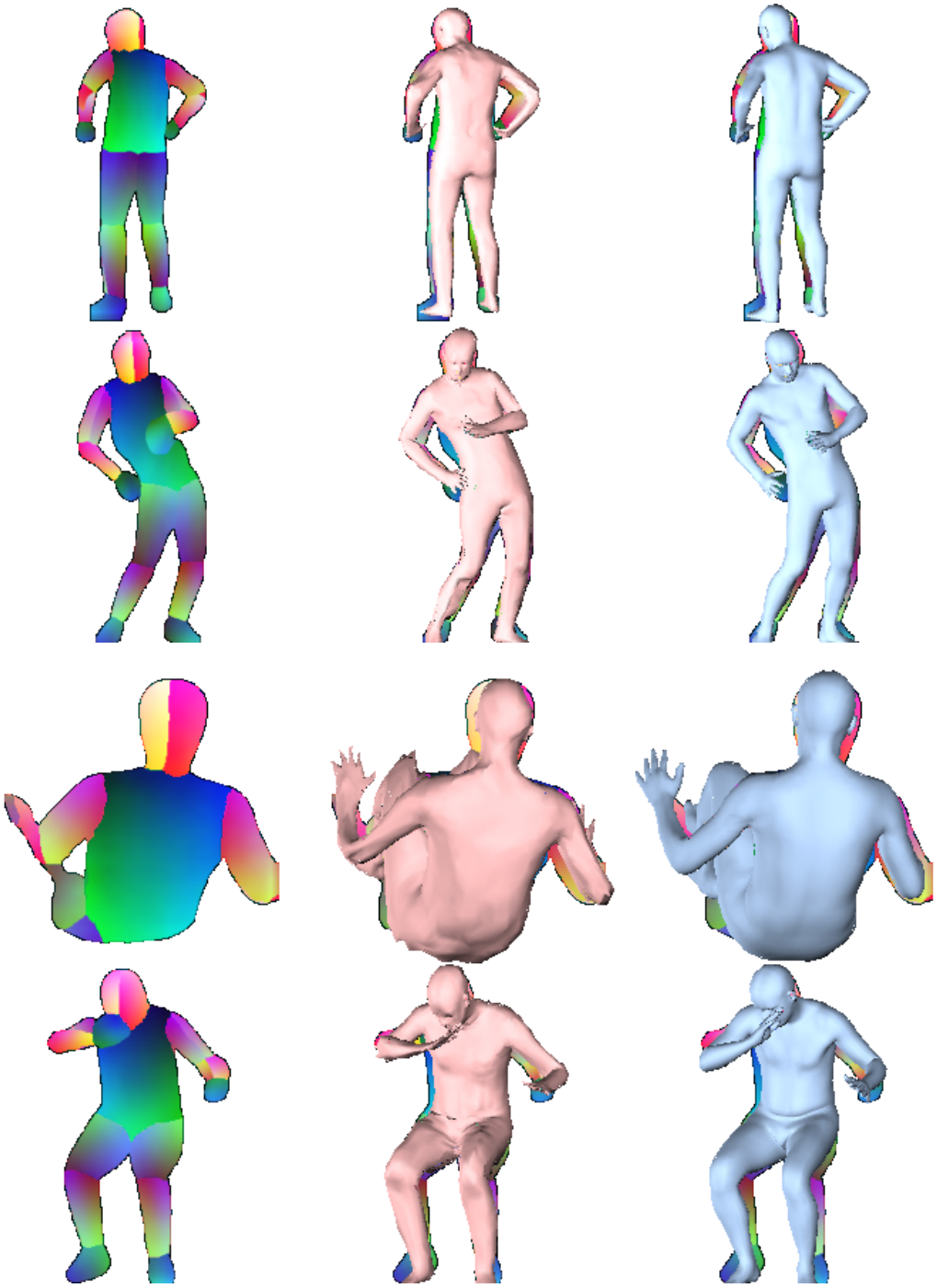

Figure 10: Qualitative results on Human3.6M with DensePose input. With light pink color we indicate the regressed non parametric shape and with light blue the SMPL model regressed from the previous shape.

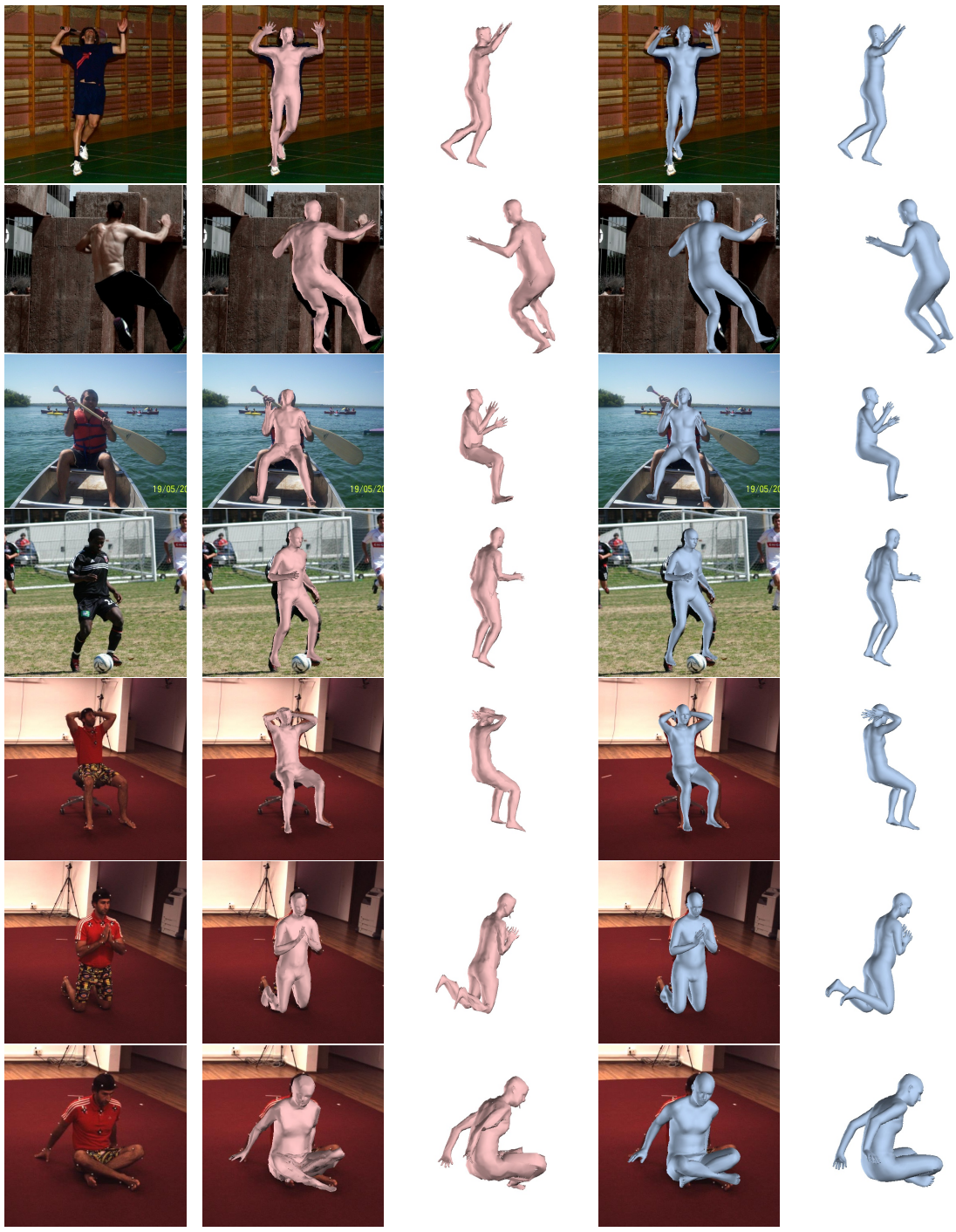

Figure 11: Visualization of our results from novel viewpoints. Rows 1-5: LSP [46]. Rows 6-7: Human3.6M [40]. From left to right: Input image, Non-parametric shape, Non-parametric shape (side view), Parametric shape, Parametric shape (side view).

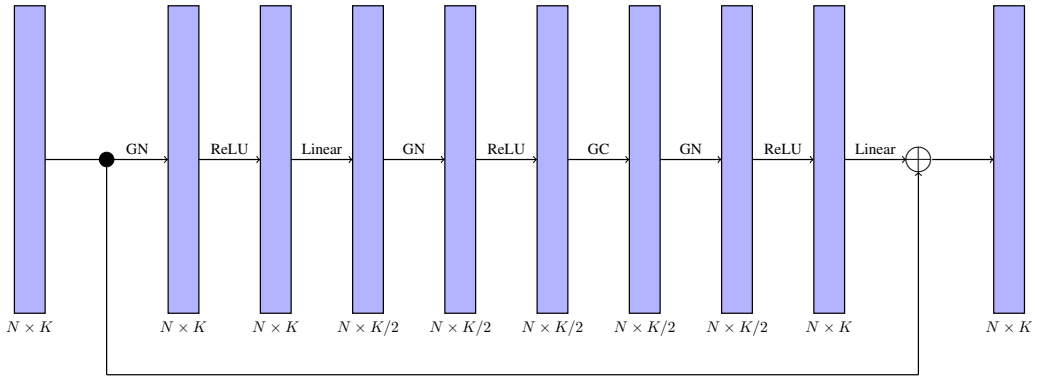

Figure 12: Graph Residual Block. Our basic building block for the Graph CNN is a redesign of the Bottleneck Residual Block [37]. GN stands for Group Normalization [146], and the Linear layers are simply per-vertex fully connected layers.

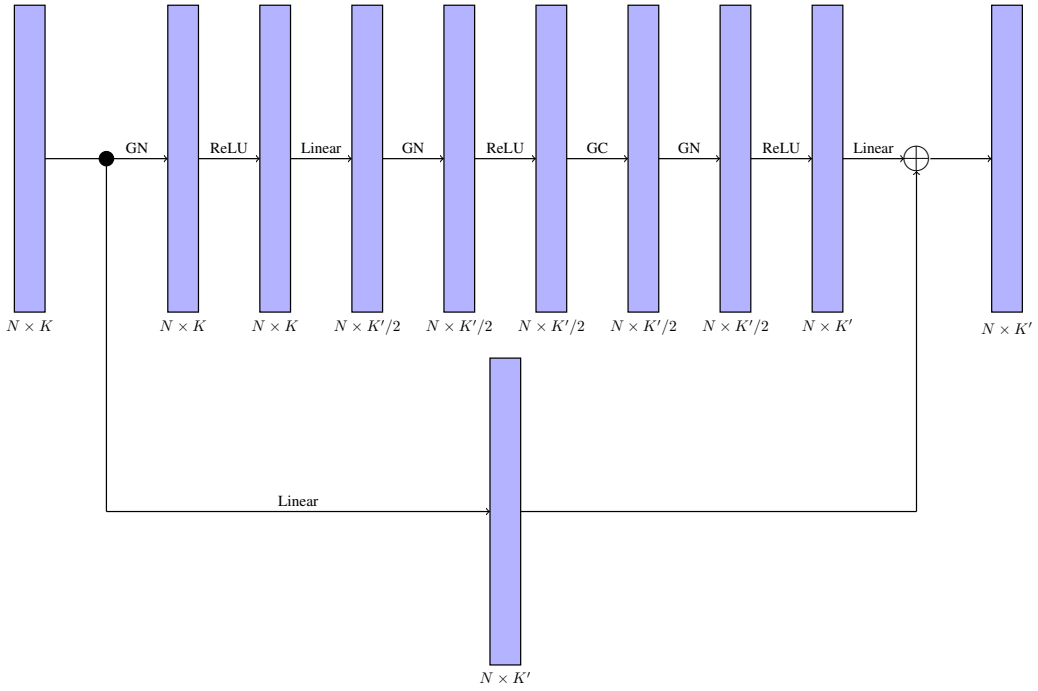

Figure 13: Graph Residual Block v2. This is the modified version of the Graph Residual Block when the number of input features is different from the number of output features.

Finally, in Figure 11 we present results of our approach visualized also from a novel viewpoint. This type of visualization allows us to inspect the accuracy of our results beyond the visible side which, and focus on the non-visible parts which is where we typically observe most errors.

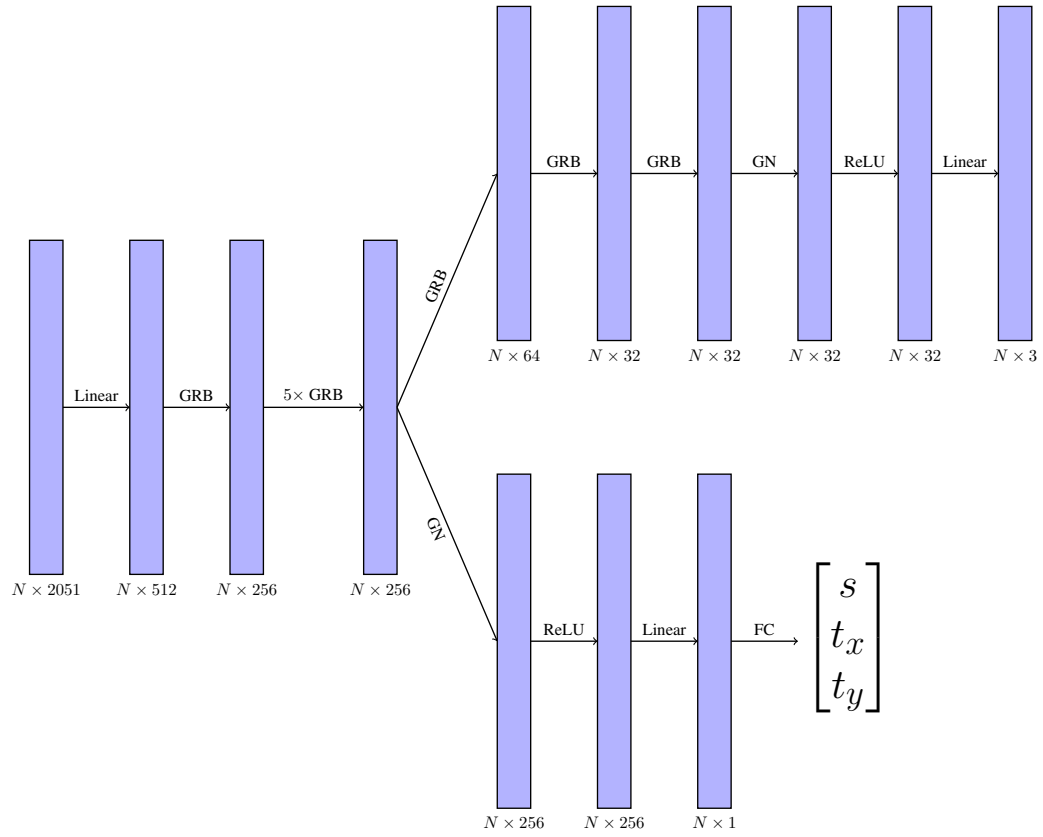

Figure 14: Graph CNN. Our Graph CNN makes use of the Graph Residual Block (GRB) of Figure 12 and Figure 13. The network eventually splits in 2 branches that predict the 3D shape and camera parameters s, t_x, t_y respectively.

2.6.4 Model Architecture

Graph CNN

As discussed in the main manuscript, the basic building block that we use in the Graph CNN is the Graph Residual Block depicted in Figure 12. It resembles the Bottleneck Residual Block [37], but we replace 1×1 convolutions with per-vertex Linear (fully connected) layers, 3×3 convolutions with the Graph convolutions proposed in [61] and Batch Normalization [39] with Group Normalization [146]. Whenever the number of input channels is different from the number of output channels, we feed the input to an additional Linear layer that maps it to the correct feature map size before adding it to the output, as seen in Figure 13. Using this Graph Residual Block, the full network architecture used in all our experiments

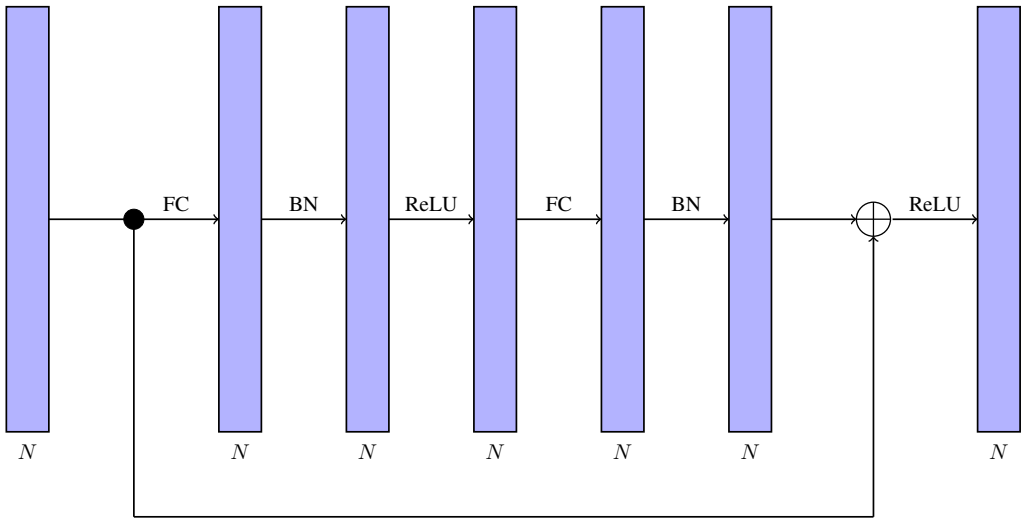

Figure 15: Fully Connected Residual Block. This figure depicts the residual block that is used in the MLP that regresses the SMPL parameters from the 3D shape. BN stands for Batch Normalization [39].

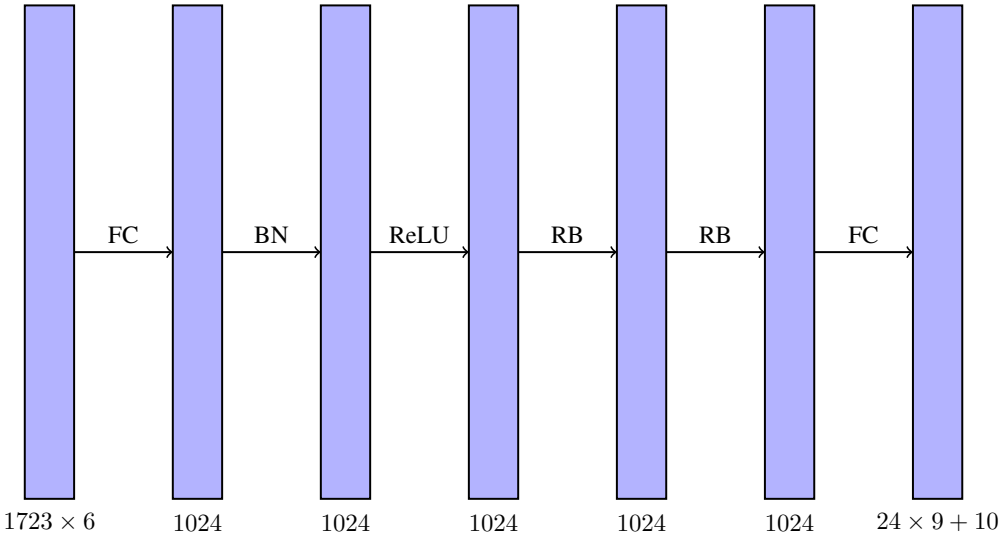

Figure 16: SMPL Regressor. This figure presents the architecture of the MLP that regresses the SMPL parameters from the 3D shape. RB stands for the Fully Connected Residual Block of Figure 15.

is depicted in Figure 14.

SMPL regressor

To estimate the SMPL parameters from the regressed shape we use a simple MLP with skip

connections. The input to the MLP is the regressed 3D shape together with the template SMPL shape, both subsampled by a factor of 4. Subsampling here is essential to avoid the explosion in the number of parameters for the fully connected layers. Also, we found that including the template 3D shape in the input speeds up the learning significantly. The network architecture is shown in Figure 16. The input size is $1723 \times 3 \times 2$ (3D vertex coordinates for both the output and the template mesh), whereas the output size is $24 \times 3 \times 3 + 10$ (rotation matrices for each of the 24 joints and 10-dimensional SMPL shape parameters).

CHAPTER 3 : Learning to Reconstruct 3D Human Pose and Shape via Model-fitting in the Loop

3.1 Introduction

Input image | Optimization result | Regression result

Figure 17: Both optimization and regression approaches have successes and failures, so this motivates our approach to build a tight collaboration between the two.

With the emergence of deep learning architectures, the dilemma between regression-based and optimization-based approaches for many computer vision problems has been more relevant than ever. Should we regress the relative camera pose, or use bundle adjustment? Is it more appropriate to regress the parameters of a face model, or fit the model to facial landmarks? These types of questions are ubiquitous within our community. Among others, 3D model-based human pose estimation has initiated similar discussions, since both optimization-based [7, 67] and regression-based approaches [52, 92, 101] have had signifi-

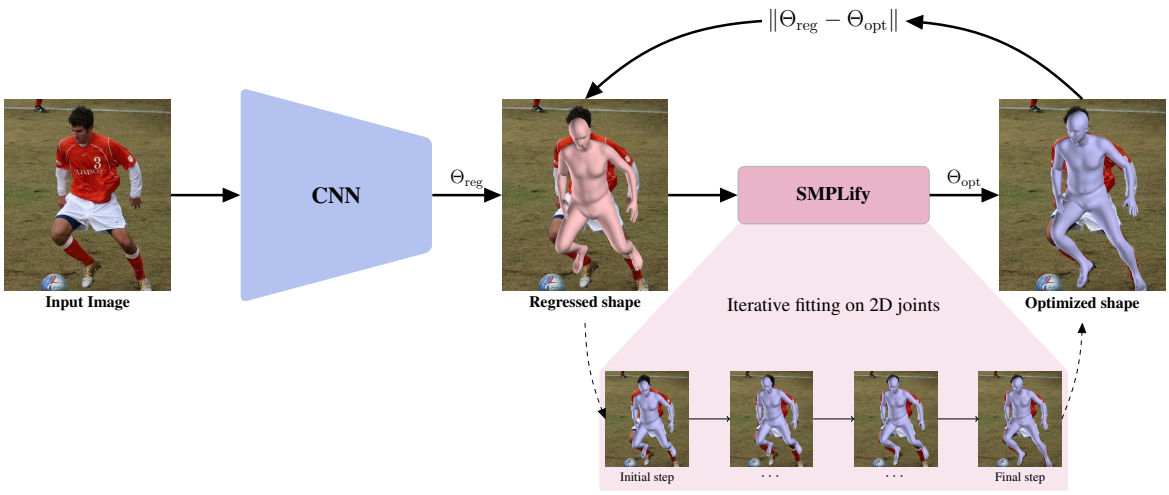

Figure 18: Overview of the proposed approach. SPIN trains a deep network for 3D human pose and shape estimation through a tight collaboration between a regression-based and an iterative optimization-based approach. During training, the network predicts the parameters Θ_{reg} of the SMPL parametric model [78]. Instead of using the ground truth 2D keypoints to apply a weak reprojection loss, we instead propose to use our regressed estimate to initialize an iterative optimization routine that fits the model to 2D keypoints (SMPLify). This procedure is done *within the training loop*. The optimized model parameters Θ_{opt} are used to explicitly supervise the output of the network and supply it with privileged model-based supervision, that is beneficial compared to the weaker and typically ambiguous 2D reprojection losses. This collaboration leads to a self-improving loop, since better fits help the network train better, while better initial estimates from the network help the optimization routine converge to better fits.

cant success recently. However, one can argue that both paradigms have weak and strong points (Figure 17). Based on this, in this work we advocate that instead of focusing on which paradigm is better, if we aim to push the field forward, we need to consider ways for collaboration between the two.

Although 3D model-based human pose is a very challenging and highly ambiguous problem, there have been fundamental works that attempt to address it. Optimization-based methods [7, 31, 67], are pretty well explored and understood. Given a parametric model of the human body, e.g., SMPL [78], an iterative fitting approach attempts to estimate the body pose and shape that best explains 2D observations, most typically 2D joint locations. Since we explicitly optimize for the agreement of the model with image features, we typically get a

good fit, but the optimization tends to be very slow and is quite sensitive to the choice of the initialization. On the other hand, recent deep learning advances have shifted the spotlight towards purely regression-based methods, using deep networks to regress the parameters of the model directly from images [52, 92, 101]. In theory, this is a very promising direction, since the deep regressor can take all pixel values into consideration, instead of relying only on a sparse set of 2D locations. Unfortunately, this type of one-shot prediction might lead to mediocre image-model alignment, while at the same time a large amount of data is necessary to properly train the network. So naturally, there is a large list of arguments in favor and against each method.

In this work, we advocate that instead of arguing over one paradigm or the other we should embrace the strengths and the weaknesses of each method and use them in a tight collaboration during training. In our approach, a deep network is used to regress the parameters of the SMPL parametric model [78]. These regressed values initialize the iterative fitting routine that aligns the model to the image given the 2D keypoints. Subsequently, the parameters of the fitted model are used as supervision for the network, closing the loop between the regression and the optimization method. This is the core of our approach, within the training loop, and uses it as a privileged form of supervision for the neural network (Figure 18). A critical characteristic of our proposed approach is that it is self-improving by nature. In the early training stages, the network will produce results close to the mean pose meaning that the iterative fitting will be prone to make errors. As more examples are provided to the network as supervision by the iterative fitting module, it will learn to produce more meaningful shapes that will also lead the optimization to more accurate model fits. Moreover, since the iterative fitting requires only 2D keypoints to fit the model, our network can be trained even when no image with corresponding 3D ground truth is available, since the 3D supervision will be provided by the optimization module. Finally, and most crucially in terms of performance, our network is trained with explicit 3D supervision, in the form of model parameters and full shape instead of weaker 2D reprojection errors as in previous works [52, 101]. This privileged form of supervision turns out to be

very important to improve the regression performance. Our approach is benchmarked in different settings and in a variety of indoor and in-the-wild datasets and it outperforms state-of-the-art model-based approaches by a significant margin.

We summarize the contributions of our approach below:

- We present SPIN , a self-improving approach for training a neural network for 3D human pose and shape estimation, through the tight collaboration of a regression- and an optimization-based method.
- Since the supervision is supplied by the iterative fitting module, training is feasible even when no image with 3D ground truth is available for training.
- The fitted model supplies our network with explicit model-based supervision which is crucial to improve performance compared to weaker 2D supervision (e.g., reprojection losses).
- We achieve state-of-the-art results in model-based 3D pose and shape estimation across many benchmarks.

3.2 Related work

Recent works have made significant advances in the frontier of skeleton-based 3D human pose estimation from single images, with many approaches achieving impressive results [83, 87, 111, 126, 133, 160]. Although this line of work has boosted the interest for 3D human pose estimation, here we will focus our review on model-based pose estimation. Approaches in this category consider a parametric model of the human body, like SMPL [78] or SCAPE [4], and the goal is to estimate the full body 3D pose and shape.

Optimization-based methods: Optimization-based approaches used to be the leading paradigm for model-based human pose estimation. Early work in the area [31, 117] attempted to estimate the parameters of the SCAPE model using silhouettes or keypoints and often there was some manual user intervention needed. Recently, the first fully auto-

matic approach, SMPLify, was introduced by Bogo *et al*. [7]. Using an off-the-shelf keypoint detector [103], SMPLify fits SMPL to 2D keypoint detections, using strong priors to guide the optimization. Beyond SMPLify, different updates to the standard pipeline have investigated incorporating in the fitting procedure, silhouette cues [67], multiple views [38], or even handle multiple people [153]. More recently, works have demonstrated fits for more expressive models in the multi-view [51], as well as the single-view setting [101, 147]. In this work, we exploit the particular effectiveness of optimization-based approaches to produce pixel-accurate fittings, but instead of using them to produce good predictions at test time, our goal is to leverage them to supply direct supervision for a neural network.

Regression-based methods: On the other end of the spectrum, recent works rely exclusively on regression to address the problem of 3D human pose and shape estimation. In most cases, given a single RGB image, a deep network is used to regress the model parameters. Considering the lack of images with full 3D shape ground truth, the majority of these works have focused on alternative supervision signals to train the deep networks. Most of them rely heavily on 2D annotations including 2D keypoints, silhouettes, or parts segmentation. This information can be used as input [137], intermediate representation [92, 101], or as supervision, by enforcing different reprojection losses [52, 92, 101, 129, 137]. Although these constraints are very useful, they are providing weak supervision for the network. Instead, we argue that strong model-based supervision, i.e., direct supervision on the model parameters and/or output mesh is crucial to improve performance. Although this type of ground truth is rarely available, we use a fitting routine in the training loop to provide the strong supervision signal to train the network.

Iterative fitting meets direct regression: Ideas of using regression approaches to improve fitting and vice versa have also been considered before in the literature. Early optimization methods required a good initial estimate which could be obtained by a discriminative approach [117]. Lassner *et al*. [67] used SMPLify to get good model fits, which could be later used for regression tasks (e.g., part segmentation or landmark detection). Rogez *et*

al. [111] also employed 3D pose pseudo annotations for training. Pavlakos *et al*. [101] used an initial prediction from their network to initialize and anchor the SMPLify optimization routine. Varol *et al*. [138] proposed an extension of SMPLify to fit SMPL on the regressed volumetric representation of their network. Although previous works have also considered the benefits of these two approaches, in our work we propose a much tighter collaboration by incorporating the fitting method within the training loop, in a self-improving manner, to harness better supervision for the network.

To put our approach in a larger context, the idea of combining direct regression networks with different optimization routines has also emerged in different settings. Training a network jointly with a graphical model has been proposed by Tompson *et al*. [135] in the context of 2D human pose estimation. Similarly, for segmentation, it is popular to use a CRF on top of the segmentation network [13], while, unrolling the CRF optimization to train the network jointly with the optimization has also been investigated [114, 158]. These ideas have also translated to 3D, where Paschalidou *et al*. [95] unrolls the MRF optimization to train it jointly with a network for depth regression. Although we draw inspiration from these works, our motivation is different since instead of unrolling the optimization, or doing a simple post-processing, we leverage the iterative fitting to provide strong supervision to the network.

3.3 Technical approach

In the following, we describe the parametric human body model, SMPL [78], and we define the basic notation. Then we provide more details about the regression network and the iterative optimization routine, based on SMPLify [7]. Finally, we describe our approach, SPIN , and give the necessary implementation details.

3.3.1 SMPL model

The SMPL body model [78], provides a function $\mathcal{M}(\theta, \beta)$ that takes as input the pose parameters θ and the shape parameters β, and returns the body mesh $M \in \mathbb{R}^{N\times 3}$, with $N = 6890$ vertices. Conveniently, the body joints X of the model can be defined as a linear

combination of the mesh vertices. A linear regressor W can be pre-trained for this task, so for k joints of interest, we define the major body joints $X \in \mathbb{R}^{k\times 3} = WM$.

3.3.2 Regression network

For the regression model, we use a deep neural network. Our architecture has the same design with Kanazawa *et al.* [52] with the only difference that we use the representation proposed by Zhou *et al.* [162] for the 3D rotations, since we empirically observed faster convergence during training. Let us now denote with f the function approximated by the neural network. A forward pass of a new image provides the regressed prediction for the model parameters $\Theta_{reg} = \{\theta_{reg}, \beta_{reg}\}$ and the camera parameters Π_{reg}. These parameters allow us to estimate the 2D projection of the joints $J_{reg} = \Pi_{reg}(X_{reg})$. Our prediction allows us to generate the mesh corresponding to the regressed parameters, $M_{reg} = \mathcal{M}(\theta_{reg}, \beta_{reg})$, as well as the joints and their reprojection J_{reg}. In this setting, a common supervision is provided using a reprojection loss on the joints:

$$L_{\text{2D}} = ||J_{reg} - J_{gt}||, \tag{3.1}$$

where J_{gt} are the ground truth 2D joints. However, in this work, we argue that this supervisory signal is very weak and puts an extra burden on the network, forcing it to search in the parameter space for a valid pose that agrees with the ground truth 2D locations.

3.3.3 Optimization routine

The iterative fitting routine follows the SMPLify work by Bogo *et al.* [7]. We give a short introduction here, but we also refer the reader to [7] for more details. SMPLify tries to fit the SMPL model to a set of 2D keypoints using an optimization-based approach. The objective function it minimizes consists of a reprojection loss term and a number of pose and shape priors. More specifically, the total objective is:

$$E_J(\beta, \theta; K, J_{est}) + \lambda_\theta E_\theta(\theta) + \lambda_a E_a(\theta) + \lambda_\beta E_\beta(\beta) \tag{3.2}$$

where β and θ are the parameters of the SMPL model, J_{est} the detected 2D joints and K the camera parameters. The first term $E_J(\beta, \theta; K, J_{est})$ is a penalty on the weighted 2D distance between J_{est} and the projected SMPL joints. $E_\theta(\theta)$ is a mixture of Gaussians pose prior trained with shapes fitted on marker data, $E_a(\theta)$ is a pose prior penalizing unnatural rotations of elbows and knees, while $E_\beta(\beta)$ is a quadratic penalty on the shape coefficients. We did not include the interpenetration error term of [7], since it makes fitting slower, while having little performance benefit.

The first step of SMPLify involves an optimization over the camera translation and body orientation, while keeping the model pose and shape fixed. After estimating the camera translation, SMPLify attempts to minimize (3.2), using a 4-stage fitting procedure. The 4-stage optimization is crucial to avoid getting trapped in local minima because the optimization is initialized from the mean pose. In contrast, since our approach uses the network prediction to initialize the optimization, we observed that a single optimization stage, with a small number of iterations, is typically enough to converge to a good fit. Also instead of estimating the initial translation using triangle similarity as in [7] we can also use the predicted camera translation from the network. This can be helpful in cases where the assumptions made in [7] (e.g., person is always standing) are not valid.

Another modification aiming at faster runtime is that we run SMPLify in batch mode. Instead of optimizing for each image sequentially, the optimization runs in parallel. Although SMPLify can have high latency that makes it unsuitable for single-image inference, we can achieve high throughput on a modern GPU by optimizing for several examples concurrently. Moreover, while SMPLify uses joints J_{est} along with their detection confidences provided by DeepCut [103], for our ground truth, we can only assume that all joints have the same confidence. This can affect negatively the fitting procedure, since typically there are small annotation mistakes, e.g., annotating joints under occlusion, or generally geometrically inconsistent annotations. To alleviate this problem, we combine the provided ground truth 2D joints for each person with the corresponding OpenPose detections [9, 11, 119, 143].

This enables us to leverage the confidence in each detection and avoid mistakes because of high-confidence erroneous annotations.

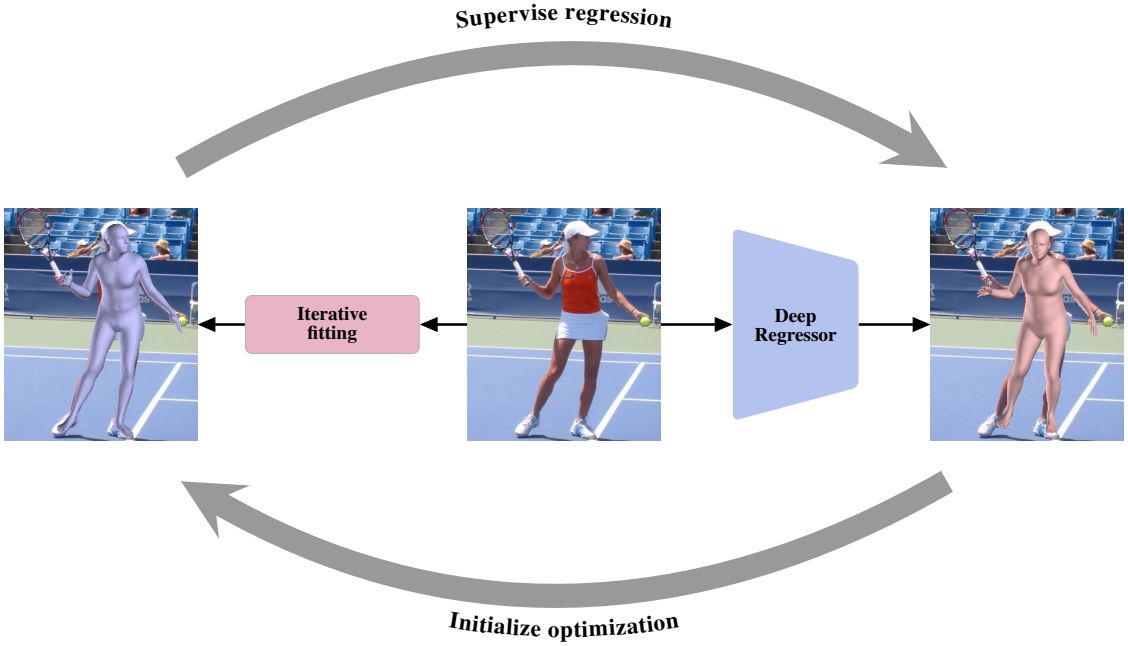

Figure 19: SPIN builds a tight collaboration between an optimization-based and a regression-based approach. A reasonable regressed estimate from the network initializes properly the optimization, thus leading to a better optimum. Similarly, a value optimized by iterative fitting can act as supervision to better train the network. The two procedures continue this collaboration forming a self-improving loop.

3.3.4 SPIN

Our approach, SPIN , builds on the insight that the previous two paradigms can form a tight collaboration to train a deep regressor for human pose and shape estimation (Figure 19). During a typical training loop, an image is forwarded through the network providing the regressed parameters Θ_{reg}. Instead of applying the typical 2D reprojection losses right away, the regressed parameters are instead used to initialize the optimization routine. This optimization is usually very slow if we start from the mean pose as an initial value. However, given a reasonable initial estimate, it can be greatly accelerated. This enables us to employ the fitting routine within the training loop. Let us now denote with $\Theta_{opt} = \{\theta_{opt}, \beta_{opt}\}$ the set of model parameters produced by the iterative fitting. These values are explicitly

optimized such that the produced shape $M_{opt} = \mathcal{M}(\theta_{opt}, \beta_{opt})$ and reprojected joints J_{opt}, align with the 2D keypoints. Given these optimized values, we can directly supervise the network function f on the parameter level:

$$L_{\text{3D}} = ||\Theta_{reg} - \Theta_{opt}||, \tag{3.3}$$

and/or the mesh level:

$$L_{\text{M}} = ||M_{reg} - M_{opt}||. \tag{3.4}$$

In practice, this has a very different effect than applying a reprojection loss for the 2D joints. Instead of forcing the network to identify a set of parameters that satisfy the joints reprojection, we supply it directly with a parametric solution that corresponds to a feasible 3D shape. Intuitively, we bypass the search of the network on the parameter space, and we directly provide a privileged set of parameters Θ_{opt} which tend to be very close to the actual optimal solution.

Another crucial characteristic of SPIN is that it is self-improving by nature. A good initial network estimate Θ_{reg} will lead the optimization to a better fit Θ_{opt}, while a good fit from the iterative routine will provide even better supervision to the network. This makes running the routine in the loop particularly important, since it enables the close collaboration between the two components.

Moreover, since the optimization routine uses only 2D joints for the fitting, and the network relies primarily on this routine for the necessary model-based supervision, our approach is applicable even in cases where no image with corresponding 3D ground truth is available for training. This resembles the unpaired setting of [52], where only 2D keypoint annotations are available, and an adversarial prior is trained to penalize invalid poses/shapes. The benefit of our approach in this setting is that instead of providing a yes/no answer to the network as the discriminator does, we explicitly supervise it with a valid pose, which leads to better performance empirically, as we demonstrate in our evaluation.

3.3.5 Implementation details

Here we discuss in more detail some further implementation details that were important for the training procedure. Although SMPLify is quite accurate, for some cases we can still get bad failures. These bad fits can make training unstable and potentially decrease performance. This motivated us to use a criterion to reject supervision from these shapes. Empirically, a simple thresholding based on the joint reprojection error worked very well in our case. For the images with rejected fits, we only supervise the regression network with a reprojection loss on the joints. Additionally, to avoid training with improbable values for the shape parameters (i.e., beyond $\pm 3\sigma$), when SMPLify returns shape values outside this range, we only supervise the β parameters with a simple L_2 loss, i.e., pushing it close to the mean shape.

To improve and accelerate training, we also incorporated a dictionary, such that for each image in our training set we can keep track of the best fit we have seen for it over all epochs. In practice, every time we compute a new optimized shape in the loop, we compare with the best fit we have seen until that point in time and if the new fit is better, we update the dictionary accordingly. To compare the quality of the fits, we again use the reprojection error on the joints. Our dictionary is initially populated with SMPLify fits, a process done offline before the training starts. To initialize SMPLify for this process, we can start from the mean pose, or use a more accurate pose, regressed from the 2D keypoints (e.g., using a network similar to Martinez *et al.* [83]). For our empirical evaluation we focus on the second strategy, but we also present similar results with the first approach in the Sup.Mat. We run the SMPLify optimization for a total of 50 iterations for each batch.

3.4 Empirical evaluation

3.4.1 Datasets

Here we give a quick description of the datasets we use for training and evaluation. We report results on Human3.6M [40], MPI-INF-3DHP [84], LSP [46], and 3DPW [140]. We train using the first three datasets (no training data from 3DPW), while similarly to [52],

we also incorporate training data with 2D annotations from other datasets, i.e., LSP-Extended [47], MPII [3], and COCO [73]. For the different settings we investigate, e.g., training with/without in the loop update, or training with/without 3D ground truth), we train a single model per setting and we use it to report results on all datasets, without fine-tuning on each particular dataset. Moreover, we clarify, that we always evaluate the network's output. No additional fitting-based post-processing is applied, as is done for example in [32]. Also, since different datasets often use different error metrics to report results, we use the metrics that are more often met in the literature for each dataset. We give a detailed definition of the various metrics in Sup.Mat.

Human3.6M: It is an indoor benchmark for 3D human pose estimation. It includes multiple subjects performing actions like Eating, Sitting and Walking. Following typical protocols, e.g., [52], we use subjects S1, S5, S6, S7, S8 for training and we evaluate on subjects S9 and S11.

MPI-INF-3DHP: It is a dataset captured with a multi-view setup mostly in indoor environments. No markers are used for the capture, so 3D pose data tend to be less accurate compared to other datasets. We use the provided training set (subjects S1 to S8) for training and we report results on the test set of the dataset.

LSP: It is a standard dataset for 2D human pose estimation. Here we employ the test set for evaluation, using the silhouette/parts annotations from Lassner *et al.* [67].

3DPW: It is a very recent dataset, captured mostly in outdoor conditions, using IMU sensors to compute pose and shape ground truth. We use this dataset only for evaluation on its defined test set.

3.4.2 Quantitative evaluation

Ablative studies: First we evaluate the components of our approach. We use in-the-wild datasets for this, since they are much more challenging, compared to the indoor benchmarks, where the models tend to overfit [40, 84].

	Rec. Error
HMR [52]	81.3
Kanazawa *et al.* [54]	72.6
Arnab *et al.* [5]	72.2
Kolotouros *et al.* [64]	70.2
Ours - static fits	66.3
Ours - in the loop	**59.2**

Table 7: Evaluation on the 3DPW dataset. The numbers are mean reconstruction errors in mm. The model-based supervision alone (Ours - static fits) outperforms similar architectures trained on the same ([52, 64]) or more data ([5, 54]). Incorporating the fitting in the loop (Ours - in the loop) further improves performance.

	FB Seg.		Part Seg.	
	acc.	f1	acc.	f1
SMPLify *oracle*	**92.17**	**0.88**	88.82	0.67
SMPLify	91.89	**0.88**	87.71	0.64
SMPLify on [101]	92.17	**0.88**	88.24	0.64
HMR [52]	91.67	0.87	87.12	0.60
Ours - static fits	91.07	0.86	88.48	0.65
Ours - in the loop	91.83	0.87	**89.41**	**0.68**

Table 8: Evaluation on foreground-background and six-part segmentation on the LSP test set. The numbers are accuracies and f1 scores. Using the model-based supervision without updating the fits achieves very competitive results, while the incorporation of the fitting in the loop propels our approach beyond the state-of-the-art. The numbers for the first two rows are taken from [67].

On the new 3DPW dataset, we evaluate pose estimation. In Table 7, we provide the results for two versions of our approach, one where the network is supervised only with the initial dictionary fits, without running the optimization in the loop (Ours - static fits), and a second where we run the optimization in the loop, and the network can benefit from the improved fits that the iterative fitting tends to produce (Ours - in the loop). To put our results into perspective, we also compare with four recent baselines ([5, 52, 54, 64]). As we can see, the use of model supervision is enough to improve performance over the other baselines. Unsurprisingly, running the iterative fitting in the loop, we can further improve the performance of the network, since it gradually gets access to better and better fits.

The same comparison is performed for the LSP dataset. In this case, we evaluate 3D shape implicitly through mesh reprojection and evaluation of silhouette and part segmentation

	Rec. Error
Lassner *et al.* [67]	93.9
SMPLify [7]	82.3
Pavlakos *et al.* [101]	75.9
HMR (unpaired) [52]	66.5
Ours (unpaired)	**62.0**
NBF [92]	59.9
HMR [52]	56.8
Ours	**41.1**

Table 9: Evaluation on the Human3.6M dataset. The numbers are mean reconstruction errors in mm. We compare with approaches that output a mesh of the human body. Approaches on the top part require no image with 3D ground truth, while approaches on the bottom part make use of 3D ground truth too. In both settings, our approach outperforms the state-of-the-art by significant margins.

accuracy. The full results for this setting are presented in Table 8. The trend here is similar to the 3DPW results. Using a static set of fits and providing model-based supervision achieves very compelling results. However, it is the incorporation of the optimization in the loop that propels our approach beyond the state-of-the-art.

To better illustrate the degree of improvement for fits in our dictionary, we provide some typical examples in Figure 20. As the training progresses, the fits improve significantly, giving to the network access to better supervision.

Comparison with the state-of-the-art: For further comparison with the state-of-the-art, we report results in additional datasets for 3D human pose estimation. Based on the different settings, proposed in the literature, we report results both when we use 3D ground truth whenever it is available (e.g., Human3.6M), and also when no image with 3D ground truth is available for training. Similarly to [52], we call this setting "unpaired", since images and 3D ground truth do not come in pairs for training.

In Table 9, we present the results of our approach on Human3.6M against other approaches that also output a full mesh of the human body (SMPL, in particular). Our approach outperforms the previous baselines when 3D ground truth is not available for training (top of the table) and when it is (bottom). We highlight that for the case that no 3D ground

	Absolute			Rigid Alignment		
	PCK	AUC	MPJPE	PCK	AUC	MPJPE
HMR (unpaired) [52]	59.6	27.9	169.5	77.1	40.7	113.2
Ours (unpaired)	**66.8**	**30.2**	**124.8**	**87.0**	**48.5**	**80.4**
Mehta *et al.* [84]	75.7	39.3	117.6	-	-	-
VNect [87]	**76.6**	**40.4**	124.7	83.9	47.3	98.0
HMR [52]	72.9	36.5	124.2	86.3	47.8	89.8
Ours	76.4	37.1	**105.2**	**92.5**	**55.6**	**67.5**

Table 10: Evaluation on the MPI-INF-3DHP dataset. The comparison is under different metrics before (left) and after (right) rigid alignment. Our approach outperforms the previous baselines. (For PCK and AUC, higher is better, while for MPJPE, lower is better).

truth is available (e.g., unpaired setting), our network does not have access to poses from Human3.6M as Kanazawa *et al.* [52], since our pose prior is trained only on CMU data. Despite that, we still outperform [52].

Similarly, we also report results on the MPI-INF-3DHP dataset, for the two settings (paired/unpaired supervision). Again, we outperform [52], while being very competitive against two approaches that do not use a parametric model of the human body [84, 87].

Finally, Figure 21 includes qualitative results of our approach from the different datasets involved in our evaluation, while Figure 24 includes some failure cases. A larger variety of results can also be found in the Sup.Mat.

3.5 Summary

This work describes SPIN , an approach that proposes a close collaboration between a regression method and an optimization-based method to train a deep network for 3D human pose and shape estimation. Our approach uses the network to provide an initial estimate to the optimization routine, which then fits the model in the loop and provides model-based supervision for the training of the network. Thus, the optimization-module and regression-module form a self-improving cycle since they can both benefit through their tight collaboration. Moreover, the privileged model-based supervision is valuable to improve the training of our network, which is also demonstrated by the empirical results, where our approach outperforms previous approaches by large margins. Simultaneously, since the

fitting routine requires only 2D keypoints to fit the model, we can train our deep network even in the absence of 3D annotations. Future work could consider extending this approach to capture multiple people [153, 154], or incorporate more expressive models of the human body [51, 97].

3.6 Supplementary Material

The goal of this Supplementary Material is to provide additional details and evaluations. Section 3.6.1, aims to provide additional qualitative results for a wide range of settings, including: qualitative results from novel viewpoints, typical failure cases, comparison with the approach of Kanazawa *et al*. [52], comparison of the "unpaired" version and the version that has access to 3D ground truth, etc. Then, in Section 3.6.2, we provide more details about the training procedure. Finally, in Section 3.6.3, we discuss the evaluation metrics used to report results in the main manuscript.

3.6.1 Further qualitative evaluation

In this Section we provide more qualitative results of our approach, that were not included in the main manuscript due to space constraints.

Side views: Figure 5 of the main manuscript provides a variety of qualitative results of our approach for all the datasets involved in our quantitative evaluation. Here we provide even more examples, with the addition of visualizations from novel viewpoints, which is a typical way to evaluate qualitatively 3D human pose estimation methods. These additional visualizations including novel viewpoints have been collected in Figure 23.

Failure cases: Despite achieving state-of-the-art results, there are still challenging cases where SPIN fails to recover a successful reconstruction of the human body. We have collected some of these failures in Figure 24. to provide further intuition regarding the potential improvements we can achieve. From inspection, the failures can be attributed to very challenging poses, viewpoints that are not common in the training set, as well as ordinal depth ambiguities.

SMPLify failures: In the main manuscript (Subsection 3.5), we discuss the typical failure modes of SMPLify. Here we provide more visual results of these failures in Figure 25. These errors motivate our decision to avoid training with some very bad fits that SMPLify can provide to our network. From inspection, these failures include wrong orientation of the body and/or extreme shape parameters. In the second case of extreme shape parameters, we observed that the camera translation is typically off (estimated to be too close or too far), because the assumptions of [7] are violated (i.e., the person is not standing parallel to the image plane). It is important to clarify though, that these failures happen when the optimization starts from the mean pose, and results are typically improved over the course of training, when the SMPLify routine is initialized with a reasonable pose estimate from the network.

Comparison with HMR [52]: Based on the results of the main manuscript, our closest competitor is the HMR approach of Kanazawa *et al*. [52]. To provide additional intuition over the benefits of our approach with respect to [52], (beyond the quantitative results), here we include further qualitative comparison with our approach, by applying HMR and our network on the same images. Example reconstruction from both approaches are presented in Figure 26. Based on this comparison, we identify that although HMR is quite robust, it has more issues with estimating the global orientation correctly, while it is less accurate for the body extremities. In contrast, our network is trained with successful SMPLify fits, which tend to get these cases correctly, so we observe more successful reconstructions also from a qualitative point of view.

"Paired" vs "Unpaired" supervision: Although for our best models we do use examples where 3D ground truth is available for training (e.g., Human3.6M and MPI-INF-3DHP), our approach is applicable even when we have access to no image with corresponding 3D ground truth. Here, we provide a qualitative comparison between these two training settings. The corresponding results are presented in Figure 27. Interestingly, our "unpaired" network produces very similar results to the network that has been trained with limited access to

3D ground truth. Significant differences can be observed only in cases with very challenging poses, or in cases with ordinal depth ambiguities, where SMPLify itself is also prone to failure.

3.6.2 Training details

Our model follows the architecture of Kanazawa *et al*. [52]. The only difference is that instead of using an axis-angle representation for the 3D rotations (as done by [52]), we instead change the output to regress the representation of Zhou *et al*. [162]. Our models were trained using the Adam optimizer with a batch size of 64, and the learning rate set to $3e-5$. We did not use learning rate decay. Training with SPIN lasts for $350k$ iterations. The model without access to any form of 3D ground truth ("unpaired") was initialized from a model pretrained on ImageNet. The model with limited access to 3D ground truth ("paired") was initialized with a model pretrained on Human3.6M [40] using full 3D pose and shape ground truth. Pretraining in this case was useful, such that the model provides better initial 3D shape estimates for the iterative fitting. All the training losses were equally weighted using a multiplier equal to 60. The only exception is the weight on the shape parameters where the multiplier is equal to 6.

The SMPLify optimization in the loop is done using the Adam optimizer in batch mode. The learning rate is equal to $1e-2$ and and the maximum number of iterations is set to 100. In these conditions, for a batch size of 64 images, the optimization takes about 3 seconds on a GeForce 1080Ti GPU, allowing us to include it within the training loop.

3.6.3 Evaluation metrics

In the main manuscript, we report results using a variety of different metrics, always following the literature and computing the metrics the same way that competing approaches do. In this Section, we provide more details about the relevant metrics and give pointers to previous works that use or define them.

Rec. Error: In Tables 1 and 3 of the main manuscript we report results on 3DPW and

Human3.6M respectively using the Reconstruction error. This error computes the mean Euclidean error over all the joints after aligning the prediction with the ground truth 3D pose through Procrustes alignment. A definition of this error is given formally in [161]

Segmentation: In Table 2 of the main manuscript we evaluate 3D shape implicitly through mesh reprojection using segmentation accuracy metrics. We report accuracy scores and f1 scores when considering only the silhouette (FB - Foreground/Background case), and also considering Part segmentation. The evaluation on LSP using these segmentation metrics is originally done by Lassner *et al*. [67].

MPI-INF-3DHP evaluation: The evaluation on MPI-INF-3DHP [84] in Table 4 of the main manuscript includes a variety of metrics reported with or without rigid alignment, i.e., with or without aligning our prediction with the ground truth using Procrustes alignment. In this case, MPJPE stands for the mean Euclidean error over all the joints. PCK is the percentage of correctly localized keypoints, where a keypoint is considered to be correctly localized if its Euclidean error is below a specific threshold (here $150mm$). Finally AUC stands for Area Under the Curve and is computed as in [84], by estimating the PCK for a variety of thresholds, from 0 to 150, with a step equal to 5.

Input image | Initial fit | Final fit

Figure 20: Examples of SMPLify fits in our dictionary at the beginning of training and at the end of training. Although SMPLify can fail when starting from an inaccurate pose (second column), given a good prediction from our network as initialization, the optimization can converge to an accurate solution (third column).

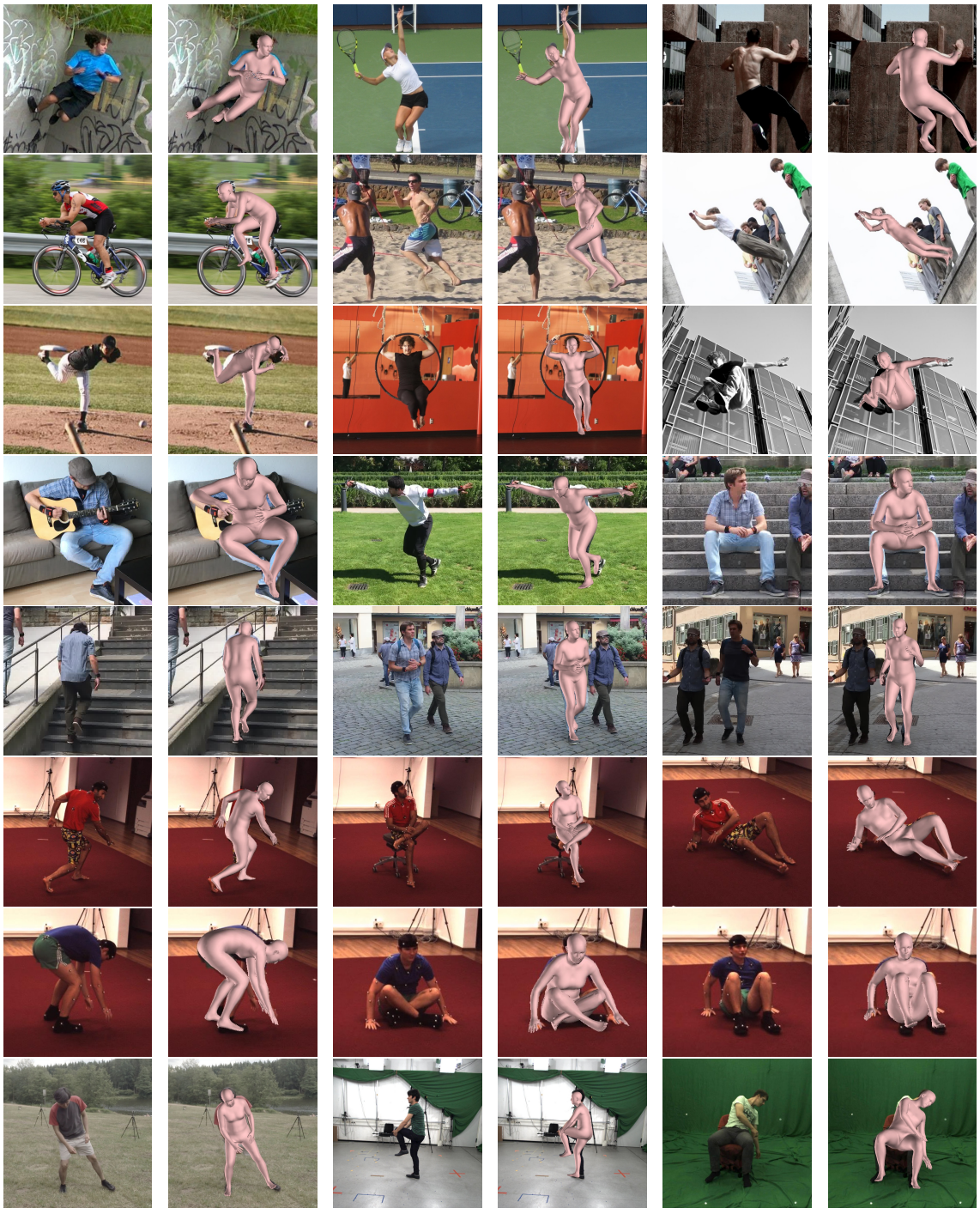

Figure 21: Qualitative results from various datasets, LSP (rows 1-3), 3DPW (rows 4-5), H36M (rows 6-7) and MPI-INF-3DHP (row 8).

Image Result Image Result

Figure 22: Erroneous reconstructions of our network. Typical failure cases can be attributed to challenging poses, ordinal depth ambiguities, viewpoints which are rare in the training set, as well as confusion due to the existence of multiple people in the scene.

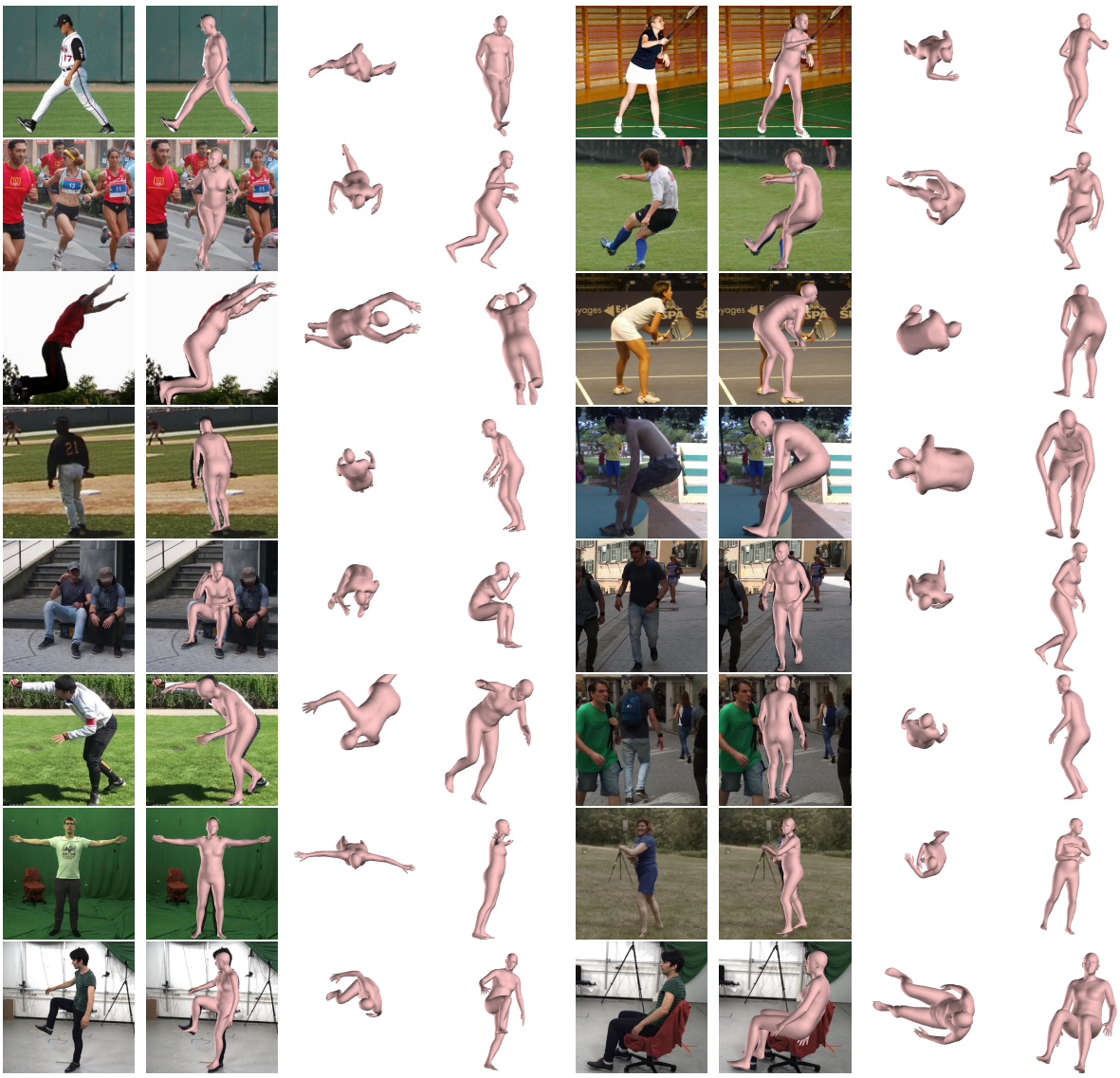

Figure 23: Successful results of SPIN. For each example from left to right: Image, Our reconstruction result in the camera frame, Our reconstruction result from a novel view (top view), Our reconstruction result from a novel view (side view).

Image Result Image Result

Figure 24: Erroneous reconstructions of our network. Typical failure cases can be attributed to challenging poses, viewpoints which are rare in the training set, as well as confusion due to the existence of multiple people in the scene.

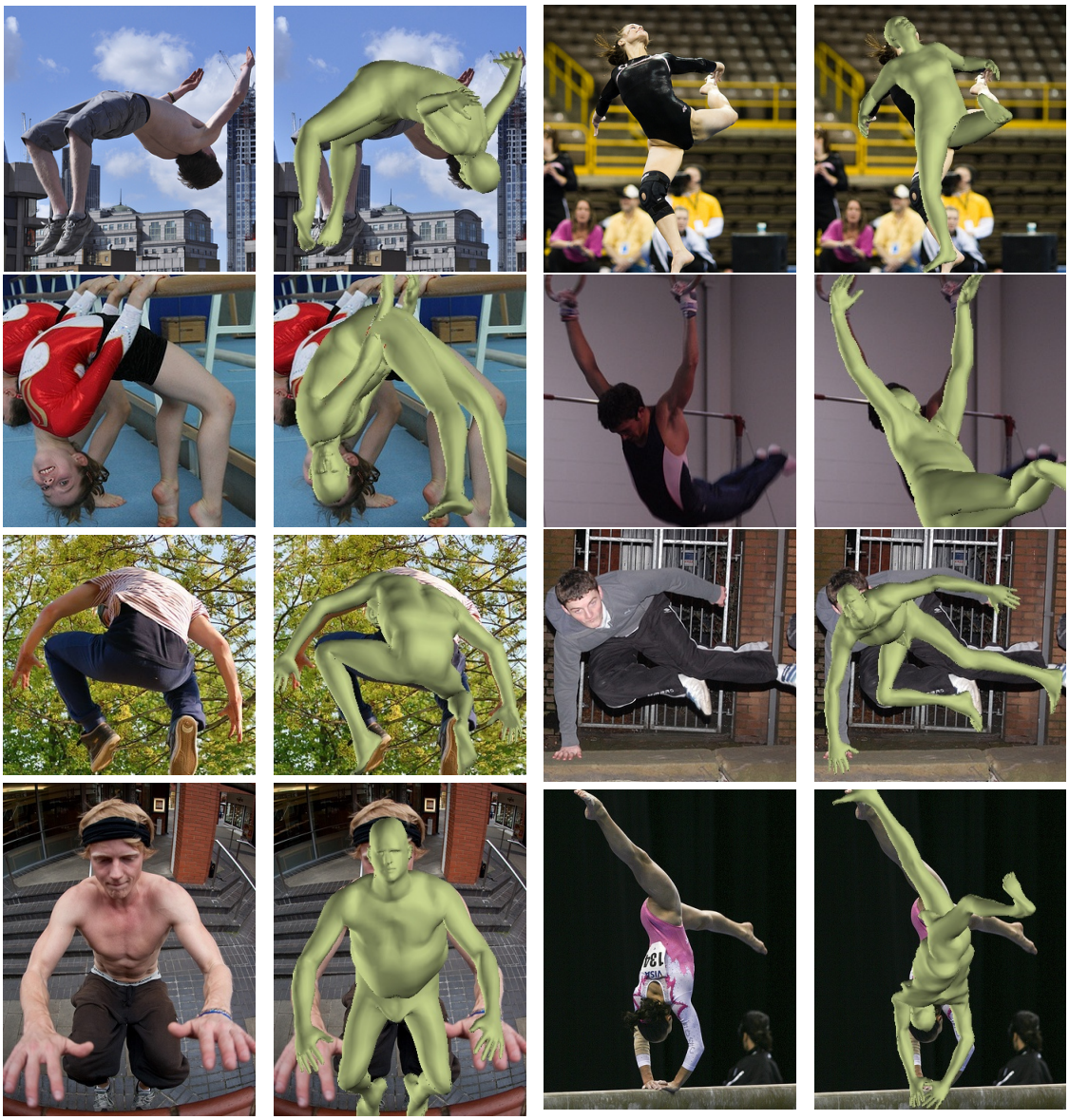

Figure 25: Typical erroneous reconstructions of SMPLify. The majority of failures occur because of errors in the orientation of the body or specific parts (first and second row), or in the estimated shape parameters (third and fourth row). In the second case, the distance from the camera has been heavily over- or under-estimated, which can produce extreme values for the shape parameters.

Image HMR Ours

Figure 26: Comparison of SPIN with HMR [52] on the LSP dataset [46]. From left to right: Input image, HMR result, Our result. HMR failures include errors in the estimation of the global orientation and the pose of the extremities (arms and legs). In contrast, SPIN is more robust in these cases, because adding the optimization in the training loop, provides more accurate supervision to the network.

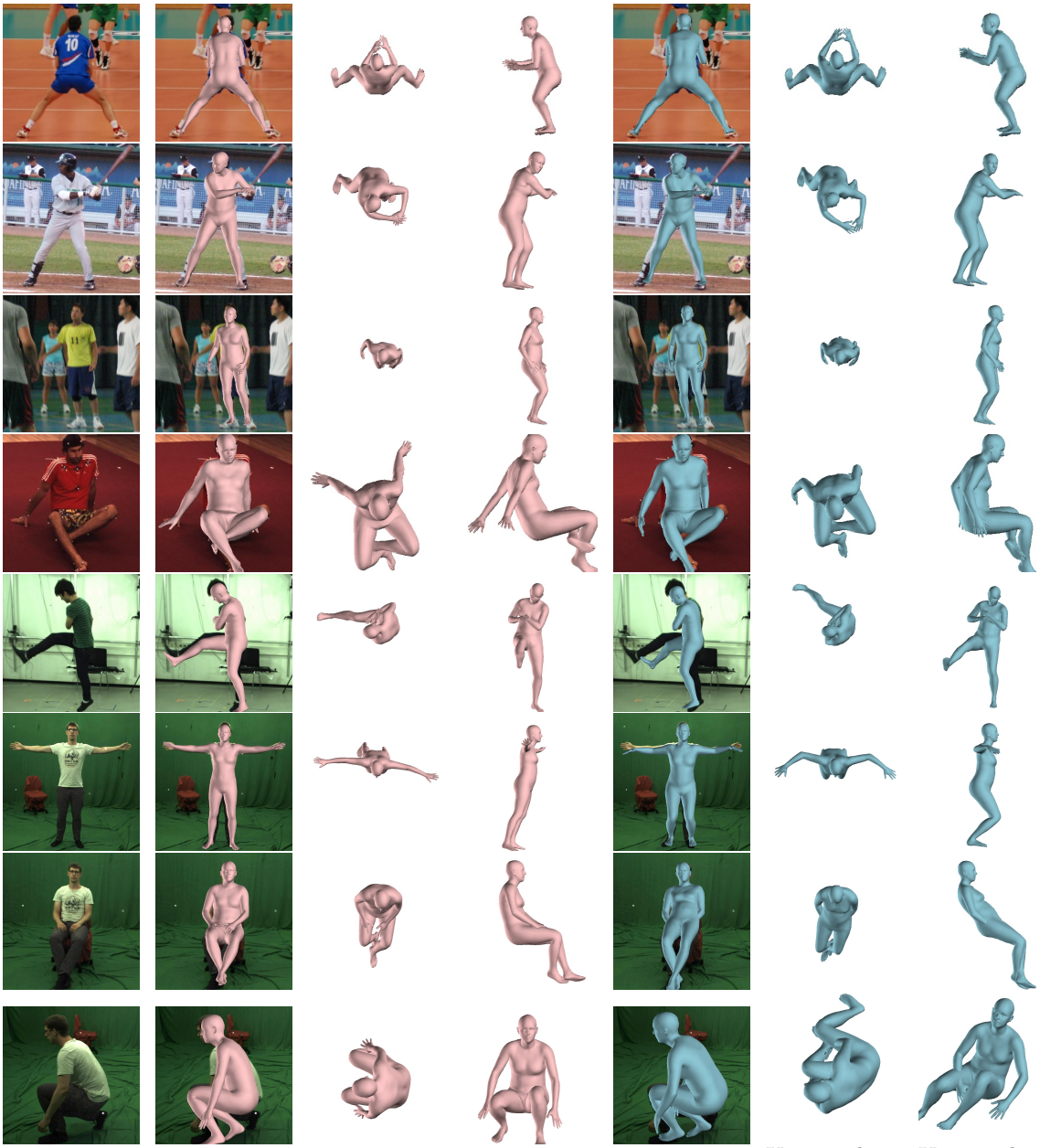

Figure 27: Comparison of "unpaired" model (no access to images with 3D ground truth) with the "paired" version (limited access to images with 3D ground truth). For each row from left to right: Image, Reconstruction result of "paired" model in the camera frame, Reconstruction result of "paired" model from top view, Reconstruction result of "paired" model from side view, Reconstruction result of "unpaired" model in the camera frame, Reconstruction result of "unpaired" model from top view, Reconstruction result of "unpaired" model from side view. Interestingly, in most cases the two versions recover similar human shapes. Important differences can only be observed in the presence of very challenging poses.

4.1 Introduction

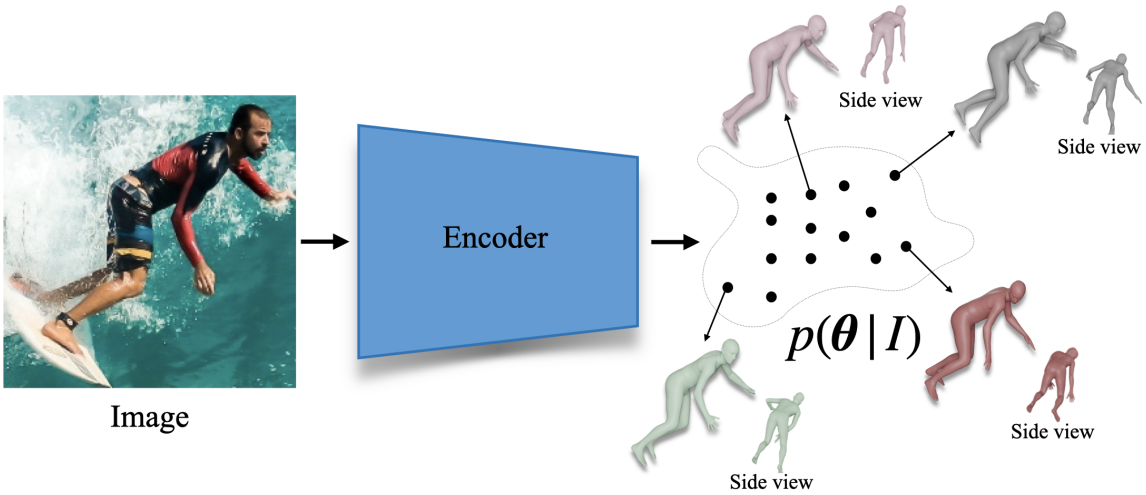

Figure 28: Probabilistic modeling for 3D human mesh recovery. We propose to recast the problem of 3D human reconstruction as learning a mapping from the input to a distribution of 3D poses. The output distribution has high probability mass on a diverse set of poses that are consistent with the 2D evidence.

Reconstructing 3D human pose from any form of 2D observations (image, 2D keypoints, silhouettes) is a fundamentally ambiguous problem. Of course, this is a very old insight, identified even from the very first approaches [68] dealing with the problem of single-view human pose reconstruction. However, the current norm for the state-of-the-art approaches is to return a single 3D estimate which is typically computed in a deterministic manner. In this work, we argue that there is great value at capturing a distribution of 3D poses conditioned on the preferred input.

Our reliance on systems that return a single deterministic 3D pose output often happens out of convenience; it makes comparison on conventional benchmarks straightforward and fair, while a single output is enough for many downstream applications. Recent literature for 3D human pose reconstruction is currently dominated by such approaches and they are very popular for image [63] or keypoint [121] input, for skeleton-based [83] or mesh-based [64] reconstruction, as well as regression [52] or optimization-based [7] approaches.

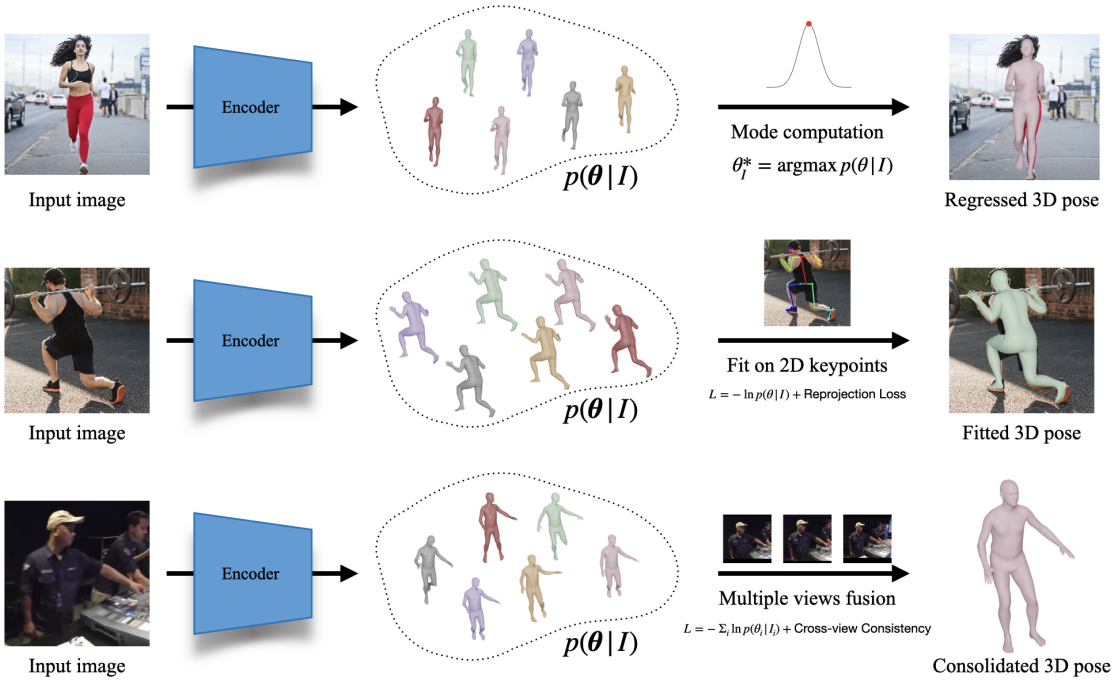

Figure 29: The value of probabilistic modeling for 3D human mesh estimation. We demonstrate that probabilistic modeling in the case of 3D human mesh estimation can be particularly useful because of its elegant and flexible form, which enables a series of downstream applications. First row: In the typical case of 3D mesh regression, we can naturally use the mode of the distribution and perform on par with approaches regressing a single 3D mesh. Second row: When keypoints (or other types of 2D evidence) are available we can treat our model as an image-based prior and fit a human body model to the keypoints by combining it with a 2D reprojection term. Third row: When multiple views are available, we can naturally consolidate all single-frame predictions by adding a cross-view consistency term. We underline that all these applications refer to test-time behavior and they use the same trained probabilistic model (no per-task training required).

On the other end of the spectrum, there have always been approaches that advocate in favor of generating multiple predictions per input. Recent efforts have demonstrated interesting potential [6, 70], but often rely on ensemble-type prediction, modifying current systems into combining N output heads instead of one. This can lead to cumbersome architectural choices, inability to scale and/or limited expressivity for the output distribution.

Our approach aims to bridge this gap and demonstrate the value of predicting a distribution of 3D poses conditioned on the provided 2D input. To achieve this, we propose an elegant and efficient approach with many desirable properties missing from recent work, and we

demonstrate its effectiveness. Instead of regressing a single estimate for the provided input, we use Normalizing Flows to regress a distribution of plausible poses. This allows us to train a network which returns a conditional distribution of 3D poses as a function of the input (*e.g.*, image or 2D keypoints), as depicted in Figure 28. Our probabilistic model allows for fast sampling of diverse outputs, we can efficiently compute the likelihood of each sample, and there is a fast and closed form solution to compute the mode of the distribution. The importance of the above is manifested in a variety of ways, which are summarized in Figure 29. First, we can easily compute the mode of the distribution, which returns the most likely 3D pose for the particular input. This is convenient, when a single estimate is required for some applications. Interestingly, this regressed value is on par with the state-of-the-art deterministic methods, so our model can be valuable even in the more conventional settings. More importantly though, by treating our trained probabilistic model as a conditional distribution, we can use it in many downstream applications to combine information from different sources. For example, when 2D keypoints are available, optimization approaches [7, 97], are used to fit parametric human body models to these 2D locations. In this case, our model can act as a powerful image-based prior that can guide the optimization towards accurate solutions that satisfy both 2D keypoint reprojection and image evidence. Similarly, when multiple views are available, we can consolidate information from all conditional distributions, by optimizing for cross-view consistency and recover a 3D result that is consistent with the available observations. Last but not least, we highlight that all these applications are available at test-time with the same trained probabilistic model, without any need for task-specific retraining.

We conduct extensive experiments to demonstrate the importance of our learned probabilistic model. We focus primarily on image-based mesh recovery [52], proposing the **ProHMR** model, but we also investigate 2D keypoint input [83]. We achieve particularly strong performance across different tasks and evaluation settings. Our contributions can be summarized as follows:

- We propose a probabilistic model for human mesh recovery and demonstrate its value in various tasks.
- In the conventional evaluations with single estimate methods, our model is on par with the state of the art.
- We demonstrate that in the presence of additional information sources, *e.g.*, multiple views or 2D keypoints, our model offers an elegant and effective way to consolidate said sources.
- In the setting of human body model fitting, our model acts as a powerful image-based prior, achieving significant boost over previous baselines.

4.2 Related work

Although our formulation is quite general and can handle different inputs/outputs, here we focus mainly on human mesh recovery from a single image [52], while we briefly touch upon other settings, specifically 3D pose estimation from 2D keypoints [83]. Since the related work is vast, here we discuss the more relevant approaches. We direct the interested reader to a recent and extensive survey [157].

4.2.1 Human mesh recovery from a single image

Regression: Recent approaches for mesh recovery are following the regression paradigm, where the parameters of a parametric model [78, 97, 149, 93] are regressed from a deep network, given a single image as input. The canonical example here is HMR [52], with many of the design decisions being adopted also by follow-up works, *e.g.*, [5, 32, 64, 98, 16, 27, 45]. Here, our regression network also follows the principles of HMR, however, instead of regressing a single 3D pose estimate, it regresses a whole distribution of plausible 3D poses given the input image.

Optimization: These methods estimate iteratively the parameters of the body model, such that it is consistent with a set of 2D cues. The canonical example of SMPLify [7] optimizes SMPL parameters given 2D keypoints. Follow-up works investigate other inputs, *e.g.*,

silhouettes [67], POFs [147], dense correspondences [32] or contact [90, 128]. However, most recent approaches [5, 63, 97] rely almost exclusively on 2D keypoints; losing the majority of pictorial cues, but gaining robustness. In this work, we demonstrate how our probabilistic model can leverage image-based information to guide the keypoint-based optimization.

Optimization-Regression hybrids: The idea of building a hybrid between the two paradigms has been explored extensively in recent work. HMR [52] and HUND [152] use a network to mimic the optimization steps and regress the updates to the model parameters. Song *et al*. [121] use the reprojection error of the model joints to guide their learning-based gradient descent approach. SPIN [63] initializes the optimization with a regression network and supervises the network with the output of the optimization. EFT [49] builds on that by updating the network weights during the fitting procedure. Our probabilistic model also investigates this type of collaboration by regressing a distribution of poses which can then be used as a prior term for the fitting.

4.2.2 Multiple hypotheses for 3D human pose

Multiple hypotheses methods have been used in the context of 3D human pose estimation to deal with the inherent ambiguities of the reconstruction such as occlusions, truncations or depth ambiguities. Jahangiri and Yuille [43] use a compositional model and anatomical constraints to generate multiple hypotheses consistent with 2D keypoint evidence. Li and Lee [70] use a Mixture Density Network instead and generate a fixed number of proposals based on the centroids of the Gaussian kernels, while Sharma *et al*. [116] tackle the same problem using a Conditional VAE. Recently, Biggs *et al*. [6] extend HMR [53] with N prediction heads. This leads to a discrete set of hypotheses, instead of a full probability of poses as we do. In a concurrent work, Sengupta *et al*. [115] use a Gaussian posterior to model the uncertainty in the parameter prediction. Differently from these methods, our approach is not limited to learning a generative model of plausible 3D poses, but rather shows how one can use such a model for useful downstream applications.

4.2.3 Normalizing Flows

Normalizing Flows are used to represent complex distributions as a series of invertible transformations of a simple base distribution. They were originally developed for modeling posterior distributions for variational inference [107, 59]. Popular examples include MADE [28], NICE [20], MAF [94], RealNVP [21] and Glow [58].

Normalizing Flows have been used in the context of 3D human pose estimation to learn a prior on the distribution of plausible poses [6, 149, 151]. These priors are usually trained using unpaired MoCap data [81]. Our work is fundamentally different from these methods in the sense that we are interested in learning a pose prior *conditioned* on 2D image evidence rather than a generic prior on the 3D pose space.

4.3 Method

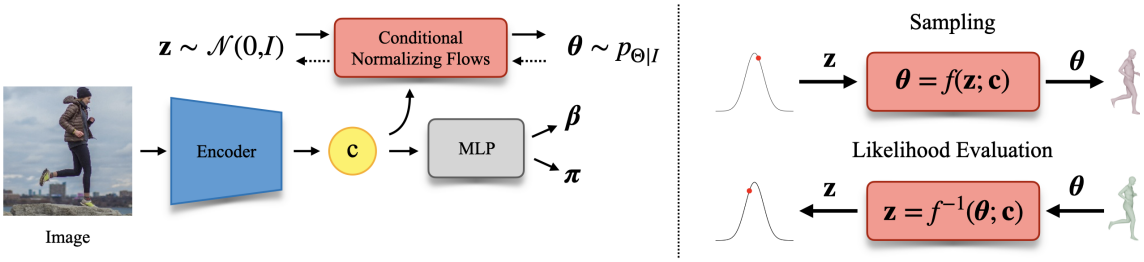

Figure 30: Architecture of the proposed probabilistic model for human mesh recovery, ProHMR. Left: Our image encoder regresses a hidden vector $\mathbf{c}$, which is used as the conditioning input to the flow model. In parallel, it is also decoded to shape parameters $\boldsymbol{\beta}$ and camera $\boldsymbol{\pi}$. Right: Our flow model learns an invertible mapping which allows for two processing directions; depending on the desired function, we can perform both sampling and fast likelihood computation.

In this Section, we present in detail our proposed approach. We start with an outline of Normalizing Flows [107] and the SMPL body model [78]. Then, we describe the model architecture and the training procedure. Finally, we show how our trained model can be used in downstream applications in a simple and straightforward manner.

4.3.1 Normalizing Flows

Let $Z \in \mathbb{R}^d$ be a random variable with distribution $p_Z(\mathbf{z})$ and $f : \mathbb{R}^d \to \mathbb{R}^d$ an invertible mapping. If we transform Z with f, then the resulting random variable $X = f(Z)$ has probability density function:

$$p_X(\mathbf{x}) = p_Z(\mathbf{z}) \left| \det \frac{\partial f}{\partial \mathbf{z}} \right|^{-1} \tag{4.1}$$

Normalizing Flow models are used to model arbitrarily complex distributions as a series of invertible transformations of a simple base distribution. Typically, the base distribution $p_Z(\mathbf{z})$ is chosen to be the standard multivariate Gaussian $\mathcal{N}(\mathbf{0}, I)$. If we write f as a composition of invertible transformations $\{f_k\}_{k=1}^K$ with $Z_0 = Z$, $Z_i = f_i(Z_{i-1})$ and $Z_K = X$, then the log-probability density of X can be computed as:

$$\ln p_X(\mathbf{x}) = \ln p_Z(\mathbf{z}) - \sum_{k=1}^{K} \ln \left| \det \frac{\partial f_i}{\partial \mathbf{z}_{i-1}} \right|. \tag{4.2}$$

Winkler *et al.* [144] extended Normalizing Flow models to model conditional distributions $p_{X|Y}(\mathbf{x}|\mathbf{y})$ by using transformations $\mathbf{x} = f(\mathbf{z}; \mathbf{y})$ that are bijective in $\mathbf{x}$ and $\mathbf{z}$.

4.3.2 SMPL model

SMPL [78] is a parametric human body model. It defines a mapping $\mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta})$ that takes as input a set of pose parameters $\boldsymbol{\theta}$ and shape parameters $\boldsymbol{\beta}$ and outputs a body mesh $M \in \mathbb{R}^{N \times 3}$, where $N = 6890$ is the number of mesh vertices. Additionally, given an output mesh, the body joints J can be expressed as a linear combination of the mesh vertices, $J = WM$, where W is a pretrained linear regressor.

4.3.3 Model design

Without loss of generality, we present our pipeline for the case where the input is an image of a person and the target output is the set of SMPL body model parameters. We call this model **ProHMR**, with the goal of *Probabilistic Human Mesh Recovery*. At the end of this

section we also show how the same method can be applied in alternative scenarios with different input and output representations.

In our setting, we are given an input image I containing a person, and our goal is to learn a distribution of plausible poses for that person conditioned on I. Since we do not have access to accurate pairs of images-shape annotations, we choose to only model the uncertainty of the SMPL pose parameters $\boldsymbol{\theta}$. Our architecture follows closely the HMR paradigm [52]. The output of our network is the conditional probability distribution $p_{\Theta|I}(\boldsymbol{\theta}|I)$ as well as point estimates for the shape and camera parameters $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ respectively.

The complete pipeline is depicted in Figure 30. Given an input image I, we encode it using a CNN g and obtain a context vector $\mathbf{c} = g(I)$. We model $p_{\Theta|I}(\boldsymbol{\theta}|\mathbf{c} = g(I))$ using Conditional Normalizing Flows. We learn a mapping $f : \mathbb{R}^d \times \mathbb{R}^c \rightarrow \mathbb{R}^d$ that is bijective in $\mathbf{z}$ and $\boldsymbol{\theta}$, *i.e.*, $\boldsymbol{\theta} = f(\mathbf{z}; \mathbf{c})$ and $\mathbf{z} = f^{-1}(\boldsymbol{\theta}; \mathbf{c})$.

We employ Normalizing Flows instead of simpler alternatives such as Mixture Density Networks (MDN) [70] because of their expressiveness and ability to model more complex distributions, as we show later in the evaluation section. In our setting, Normalizing Flows have also clear advantages over VAEs, since VAEs do not offer an easy way to compute the likelihood of a given output sample, which is crucial when using our model in downstream tasks.

Our Normalizing Flow model is based on the Glow architecture [58]. Each building block f_i is comprised of 3 basic transformations:

$$f_i = f_{coupl} \circ f_{lin} \circ f_{norm}, \tag{4.3}$$

where $f_{norm}(\mathbf{z}) = \mathbf{a} \odot \mathbf{z} + \mathbf{b}$ (Instance Normalization), $f_{lin}(\mathbf{z}) = W\mathbf{z} + \mathbf{b}$ (Linear transformation) and $f_{coupl} = [\mathbf{z}_{1:k}, \mathbf{z}_{k+1:d} + \mathbf{t}(\mathbf{z}_{1:d}, \mathbf{c})]$ (Additive coupling). To make the inversion and the Jacobian computation faster, in the linear transformation we parametrize the LU decomposition of W. The final flow model is obtained by composing four of these building

blocks.

The selected flow model allows us to perform both fast likelihood computation and fast sampling from the distribution. At the same time, a very important property is that the determinant of the Jacobian does not depend on $\mathbf{z}$, which in turn means that the mode of the output distribution is:

$$\boldsymbol{\theta}_I^* = \text{argmax}_{\boldsymbol{\theta}}\, p_{\Theta|I}(\boldsymbol{\theta}|\mathbf{c}) = f(\mathbf{0}; \mathbf{c}). \tag{4.4}$$

This result allows us to use our model as a *predictive model* in a straightforward way; in the absence of any additional side-information, we make predictions using the mode of the output distribution.

To regress the camera and the SMPL shape parameters, we use a small MLP h that takes as input the context vector $\mathbf{c}$ and outputs a single point estimate, *i.e.*, $[\boldsymbol{\beta}, \boldsymbol{\pi}] = h(\mathbf{c})$. We also experimented with having $\boldsymbol{\beta}$ and $\boldsymbol{\pi}$ depend on $\boldsymbol{\theta}$, but there was no observable improvement.

4.3.4 Training objective

Let us assume that we have a collection of images paired with SMPL pose annotations. Typically, Normalizing Flow models are trained to minimize the negative log-likelihood of the ground truth examples $\boldsymbol{\theta}_{gt}$, *i.e.* the loss function is:

$$L_{nll} = -\ln p_{\Theta|I}(\boldsymbol{\theta}_{gt}|\mathbf{c}). \tag{4.5}$$

However, for the task of 3D pose estimation, 3D annotations are generally not available except for a small number of indoor datasets captured in constrained studio environments [40, 84] and methods trained on those datasets fail to generalize in challenging in-the-wild scenes. Consequently, previous methods like [52] propose to use examples with only 2D keypoint annotations and minimize the keypoint reprojection loss jointly with an adversarial prior. To make such a mixed training possible within our framework, we propose

to minimize the expectation of the above error with respect to the learned distribution, *i.e.*,

$$L_{exp} = \mathbb{E}_{\boldsymbol{\theta} \sim p_{\Theta|I}}[L_{2D}(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\pi}) + L_{adv}(\boldsymbol{\theta}, \boldsymbol{\beta})]. \tag{4.6}$$

To make this loss differentiable we use the *Law of the Unconscious Statistician* and rewrite the expectation as:

$$L_{exp} = \mathbb{E}_{\mathbf{z} \sim p_Z}[L_{2D}(f(\mathbf{z}; \mathbf{c}), \boldsymbol{\beta}, \boldsymbol{\pi}) + L_{adv}(f(\mathbf{z}; c), \boldsymbol{\beta})]. \tag{4.7}$$

Conceptually, even though we do not have ground truth annotations, to maximize the conditional probability of these examples we can still constrain the form of the output distribution by forcing the output samples to have low reprojection error on average and lie on the manifold of valid poses. As in the case of VAEs [60], we approximate the expectation by drawing a single sample from the prior.

As mentioned previously, our goal is to use our model not only as a generative model but also as a predictive model. Thus, we propose to exploit the property that for each image I, the mode $\boldsymbol{\theta}_I^*$ of the output distribution corresponds to the transformation of $\mathbf{z} = \mathbf{0}$. We do this by explicitly supervising $\boldsymbol{\theta}_I^*$ with all the available annotations as in a standard regression framework and minimize:

$$L_{mode} = L_{3D}(\boldsymbol{\theta}_I^*, \boldsymbol{\beta}) + L_{2D}(\boldsymbol{\theta}_I^*, \boldsymbol{\beta}, \boldsymbol{\pi}) + L_{adv}(\boldsymbol{\theta}_I^*, \boldsymbol{\beta}), \tag{4.8}$$

where L_{3D} is the loss on the available 3D annotations (3D joints and/or SMPL parameters) whenever they are available. As we show in the experimental section, this explicit supervision of the mode of the output distribution helps boost the performance of our model in predictive tasks.

It is important to mention that L_{exp} is not redundant in the presence of L_{mode}; the behavior of the mode is not indicative of the full distribution, whereas L_{exp} encourages the

distribution to have certain desirable properties.

Finally, for modeling rotations we use the 6D representation proposed in [162]. One issue with this particular representation is that it is not unique. For example, for any 3D vectors x and y, $[x, y]$ and $[\alpha x, \beta x + \gamma y]$ are mapped to the same rotation matrix. Empirically we found that putting no constraints on the 6D representation results in large discrepancy between examples with full 3D SMPL parameter supervision and examples with only 2D keypoint annotations. Among other things, this caused mode collapse for the examples without 3D ground truth. Thus, we introduce another loss function L_{orth} that forces the 6D representations of the samples drawn from the distribution to be close to the orthonormal 6D representation.

Eventually, the final training objective becomes:

$$L = \lambda_{nll} L_{nll} + \lambda_{exp} L_{exp} + \lambda_{mode} L_{mode} + \lambda_{orth} L_{orth}. \tag{4.9}$$

4.3.5 Downstream applications

In this part we show how our learned conditional distribution can be used in a series of downstream applications. We highlight that all these applications refer to *test-time* processing with the same trained model without any special per-task training. Examples of such tasks are shown in Figure 29. These applications fall under the more general umbrella of Maximum a Posteriori estimation where we use all available evidence to make more informed predictions.

3D pose regression As already discussed in previous sections, we can use our model in conventional tasks such as 3D pose regression from a single image. In the absence of additional evidence, the most appropriate choice for making predictions is to pick the mode $\boldsymbol{\theta}_I^*$ of the distribution.

Body model fitting SMPLify [7] is a popular method that fits the SMPL body model to a set of 2D keypoints using a traditional optimization approach. The objective is:

$$\lambda_J E_J + \lambda_\theta E_\theta + \lambda_\alpha E_\alpha + \lambda_\beta E_\beta, \tag{4.10}$$

where E_J penalizes the weighted 2D distance between the projected model joints and the detected joints, E_θ is a Mixture of Gaussian 3D pose prior, E_α is a pose prior penalizing unnatural rotations of elbows and knees and E_β is a quadratic penalty on the shape coefficients.

Fitting a parametric body model to 2D image landmarks is a very challenging and inherently ambiguous problem. The data term E_J is purely driven by the 2D keypoints and disregards rich information contained in the input image. SPIN [63] partially addresses this issue by using an image-based regression network that provides a good initialization for the optimization, helping the fitting to converge to a better minimum. However, the image information is only used in the initialization phase, as SMPLify does not incorporate explicit image-specific priors that prevent the pose to deviate arbitrarily far from the set of plausible poses for the given image. The drifting problem is also an important limitation of [49], forcing the approach to rely on good initialization and carefully chosen stopping criteria.

Motivated by these limitations, we propose to replace the weaker generic 3D priors E_θ and E_α with an explicit pose prior $E_{\theta|I} = -\ln p_{\Theta|I}(\boldsymbol{\theta}|\mathbf{c})$ that models the likelihood of a given pose conditioned on the image evidence. Thus, the final optimization objective becomes:

$$\lambda_J E_J - \ln p_{\Theta|I}(\boldsymbol{\theta}|\mathbf{c}) + \lambda_\beta E_\beta. \tag{4.11}$$

As initialization for the fitting we use the mode $\boldsymbol{\theta}_I^*$ of the conditional distribution. In the experimental section we show that by using this learned image-based prior we are able to consistently improve the fitting results, both qualitatively and quantitatively, as reflected in the 3D metrics.

Multiple views fusion Although our model has been trained for single-image reconstruction, we can still use the learned conditional distribution to obtain refined pose estimates in the presence of multiple views of a person. Let us assume that we have a set $\{I_n\}_1^N$ of uncalibrated views of the same subject. We partition the pose vector of each frame as $\boldsymbol{\theta}_n = (\boldsymbol{\theta}_n^g, \boldsymbol{\theta}_n^b)$ where $\boldsymbol{\theta}_n^g$ corresponds to the global rotation of the model and $\boldsymbol{\theta}_n^b$ is the body pose. We propose to refine the pose by minimizing the following objective:

$$-\sum_{n=1}^{N} \ln p(\boldsymbol{\theta}_n|\mathbf{c}_n) + \lambda \sum_{n=1}^{N} ||\boldsymbol{\theta}_n^b - \bar{\boldsymbol{\theta}}^b||_2^2, \tag{4.12}$$

where $\bar{\boldsymbol{\theta}}^b = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\theta}_n^b$. The second term of the objective is equivalent to minimizing the squared distance between all pairs of poses.

4.3.6 Additional details

ProHMR. Following previous works [52, 63] we use ResNet-50 [37] as the encoder. For the Normalizing Flows we use 4 building blocks f_i. For more details about the architecture, datasets and the training hyperparameters we refer the reader to the supplementary material.

2D pose lifting. Complementary to ProHMR, we use our approach to lift 2D poses to 3D skeletons, as in Martinez *et al.* [83]. We use the same Normalizing Flow architecture as in ProHMR. In this case the input is a set of 2D Hourglass detections [91] and the output is the 3D pose coordinates. For the encoder g, instead of a CNN, we use the backbone from [83]. Since all examples have full 3D supervision, our training objective consists only of L_{nll} and L_{mode}.

Downstream tasks. For the fitting procedure employed in the downstream tasks, we found it beneficial to perform the optimization in the latent space instead of the pose space directly (similarly to SMPLify-X [97]). Thus, we leave $\mathbf{z}$ as a free variable and decode it into the pose vector $\boldsymbol{\theta} = f(\mathbf{z}; \mathbf{c})$. Also, since for our Normalizing Flow model the determinant of the Jacobian does not depend on $\mathbf{z}$, the likelihood term becomes $\ln p(\boldsymbol{\theta}|\mathbf{c}) = -||\mathbf{z}||_2^2 + \text{const.}$

4.4 Experimental evaluation

In this Section we present the experimental evaluation of our approach. First we provide an outline of the datasets used for training and evaluation and then we will present detailed quantitative and qualitative evaluation results.

4.4.1 Datasets

We report results on Human3.6M [40], MPI-INF-3DHP [84], 3DPW [140] and Mannequin Challenge [72], where we use the annotations produced by Leroy *et al*. [69]. For training, we use datasets with 3D ground truth (Human3.6M [40] and MPI-INF-3DHP [84]), as well as datasets with 2D keypoint annotations (COCO [73] and MPII [3]) augmented with pseudo ground truth SMPL parameters from SPIN [63], whenever they are available.

4.4.2 Quantitative evaluation

In this part we evaluate different aspects of our proposed approach. We compare the predictive accuracy of our model with standard regression methods and show that it achieves comparable performance with the state of the art in human mesh recovery. We also benchmark the generative capabilities of our method in multiple hypotheses scenarios, where we outperform previous approaches. Finally, we demonstrate that our learned image-conditioned prior can boost the performance in downstream applications such as model fitting and multi-view refinement.

Human mesh recovery. First, we focus on the predictive performance of our model, comparing it against other state-of-the-art methods that regress SMPL body model parameters. For the evaluation of ProHMR, we use the mode $\boldsymbol{\theta}_I^*$ of the learned distribution. For Biggs *et al*. [6] we report the metrics after quantizing to $n = 1$ sample. Based on the results of Table 11, using ProHMR as a regressor, leads to performance comparable to the state of the art. This shows that we can indeed recast the problem from point to density estimation without any significant loss in performance.

Multiple hypotheses. Next, we compare the representational power of ProHMR with

Figure 31: Samples from the learned distribution. Pink colored mesh corresponds to the mode.

	3DPW	H36M	MPI-INF-3DHP
HMR [52]	81.3	56.8	89.8
SPIN [63]	59.1	41.1	67.5
Biggs *et al.* [6]	59.9	41.6	N/A
ProHMR	**59.8**	**41.2**	**65.0**

Table 11: Evaluation on human mesh recovery. Our model achieves accuracy comparable with the state of the art. Numbers reported are PA-MPJPE in mm.

different multiple hypotheses baselines, including Biggs *et al.* [6], as well as the MDN and Conditional VAE variants explored in the same paper. Following [6], we report results for small sample sizes n. Since we are interested in measuring the representational power of the learned distribution, we also compare the minimum 3D pose error of samples drawn from each distribution as proposed in [116]. We present the detailed results for Human3.6M and 3DPW in Table 12.

	$n = 5$		$n = 10$		$n = 25$		*min*	
	[140]	[40]	[140]	[40]	[140]	[40]	[140]	[40]
[6] (MDN)	61.2	43.3	60.7	43.0	60.1	42.7	60.1	42.7
[6] (CVAE)	60.7	46.4	60.5	46.3	60.3	46.2	60.3	46.2
[6] (NF)	57.1	42.0	56.6	42.2	55.6	42.2	55.6	41.6
ProHMR	**56.5**	**39.4**	**54.6**	**38.3**	**52.4**	**36.8**	**40.8**	**29.9**

Table 12: Multiple hypotheses evalutation. Numbers are PA-MPJPE in mm. We report errors for small n and the *minimum* error over samples drawn from the distribution.

Figure 32: Model fitting results. Pink: Regression. Green: ProHMR + fitting. Grey: Regression + SMPLify

Model fitting. In this part we evaluate the accuracy of different methods that fit the SMPL body model to a set of 2D keypoints. The body model fitting baselines we compare include the standard SMPLify [7, 97], EFT [49], and our proposed fitting with the learned image-conditioned prior. For both SMPLify and EFT we use publicly available implementations and initialize the fitting process with SPIN, while for SMPLify we use two different versions for the pose prior, GMM [7] and VPoser [97]. For a fair evaluation of the performance benefit, we compare methods that are trained on the same datasets and have similar regression performance. The results are presented in Table 13. While performing SMPLify on top of regression improves the model-image alignment, it increases the 3D pose errors, especially when using OpenPose detections [9]. We hypothesize that this happens because of the generic 3D pose prior terms of SMPLify. EFT on top of regression improves the 3D pose metrics, however our method manages to push the accuracy even further. In 3DPW our approach has a 4.7mm relative error improvement vs. 2.6mm for EFT, while if we use the ground truth 2D keypoints in Human3.6M we get a 6.3mm improvement vs 3.1mm for EFT.

Multi-view refinement. We evaluate the effect of our learned image-conditioned prior at refining the pose predictions in uncalibrated multi-view scenarios. For benchmarking,

	3DPW	H36M (OP)	H36M (GT)
SPIN [63]	59.2	41.8	41.8
SPIN+SMPLify (GMM) [7]	66.5	54.6	43.3
SPIN+SMPLify (VPoser) [97]	70.9	53.5	39.9
SPIN+EFT [49]	56.6	41.6	38.7
ProHMR	59.8	41.2	41.2
ProHMR + fitting	**55.1**	**39.3**	**34.8**

Table 13: Evaluation of different model fitting methods. The fitting algorithms are initialized by the corresponding regression results. All numbers are PA-MPJPE in mm.

	H36M		Mannequin	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
ProHMR	65.1	43.7	176.0	91.9
ProHMR + rot avg	64.8	35.2	174.4	85.1
ProHMR + fitting	**62.2**	**34.5**	**171.3**	**83.9**

Table 14: Evaluation of multi-view refinement. We compare single-image 3D reconstruction with a baseline refinement using rotation averaging and the proposed optimization-based refinement scheme.

we use Human3.6M and the more challenging Mannequin Challenge dataset. We compare our fitting-based method against the individual per-view predictions and a baseline that performs rotation averaging in Table 14. For the rotation averaging we first average the per-view rotation matrices and then project them back to $SO(3)$ using SVD.

Ablation study. We also assess the significance of the term L_{mode} that we use to explicitly supervise the mode of the learned distribution. We report results for training ProHMR with and without this loss in Table 15. We can see that including L_{mode} is crucial to achieve competitive performance in conventional regression tasks.

Additional evaluations. Finally, we show that the proposed modeling is general enough to handle different input and output representations. Here, we consider the setting of lifitng

	3DPW	H36M	MPI-INF-3DHP
ProHMR (w/o L_{mode})	67.4	54.8	76.5
ProHMR	**59.8**	**41.2**	**65.0**

Table 15: Ablation for L_{mode}. Numbers are PA-MPJPE.

	MPJPE	PA-MPJPE
Martinez *et al.* [83]	62.9	47.7
Li and Lee [70] (mode)	64.5	47.8
Ours	62.9	47.6
Li and Lee [70] (min)	42.6	34.4
Ours (min)	**42.4**	**32.9**

Table 16: Evaluation of 3D pose accuracy for skeleton-based 2D pose lifting on Human3.6M. Top: Regression accuracy. Bottom: Minimum error of the distributions.

a 2D pose input to a 3D skeleton output [83] and present results in Table 16. Our model performs on par with an equivalent regression approach [83], while it outperforms the MDN method of Li and Lee [70].

4.4.3 Qualitative results

In Figure 31 we show sample reconstructions of our method. Additionally, in Figure 32 we show comparisons of our model fitting approach with SMPLify. Our method produces more realistic reconstructions overall, particularly in cases where there are missing or very low confidence keypoint detections. In cases like that (*e.g.*, example of last row), our image-based prior, unlike SMPLify, does not let the pose deviate far from the image evidence.

4.5 Summary

This work presents a probabilistic model for 3D human mesh recovery from 2D evidence. Unlike most approaches that output a single point estimate for the 3D pose, we propose to learn a mapping from the input to a distribution of plausible poses. We model this distribution using Conditional Normalizing Flows. Our probabilistic model allows for sampling of diverse outputs, efficient computation of the likelihood of each sample, and a fast and closed-form solution for the mode. We demonstrate the effectiveness of our method with empirical results in several benchmarks. Future work could consider extending our approach to other classes of articulated or non-articulated objects and potentially model other ambiguities like the depth-size trade-off.

4.6 Supplementary

In this Supplementary Material we provide additional details that were not included in the main paper due to space constraints. In Section 4.6.1 we extend the discussion about probabilistic pose models. In Section 4.6.2 we describe in detail the architecture of our model. Section 4.6.3 includes all training details whereas Section 4.6.4 describes the datasets used for training and evaluation. FInally, in Section 4.6.5 we report additional quantitative and qualitative evaluations.

4.6.1 Additional discussion

Heatmap-based methods One additional class of non-parametric probabilistic models for 3D human pose estimation is the heatmap-based methods [100]. Heatmap-based methods allow for pose sampling (for each joint, we first sample a voxel and then sample again uniformly inside each voxel), as well as likelihood evaluation for a given pose. However the main issue with heatmap-based methods is that by design the probability distribution is factorized, *i.e.*,

$$p(\boldsymbol{\theta}|I) = \prod_i p(\boldsymbol{\theta}_i|I), \tag{4.13}$$

for all joints $\boldsymbol{\theta}_i = (x_i, y_i, z_i)$. This type of model is problematic for sampling because it can lead to unrealistic poses, *e.g.*, in ambiguous cases where the output heatmaps are not unimodal, since each joint location is sampled independently of the other joints. Moreover, it is not possible to extend heatmap-based methods to non-Euclidean output spaces such as the space of SMPL parameters.

4.6.2 Architecture details

In this Section we describe in detail the architecture of the proposed model. In Figure 33 we show the design of the proposed flow model and the information flow both in forward and reverse mode. The implementation of f_{coupl} is shown in Figure 34, whereas Figure 35 depicts the architecture of the residual block used in f_{coupl}. The number of channels in the residual block is $n = 1024$. For the partitioning of $x = (x_1, x_2)$, we alternate between the odd

and even dimensions of x in each successive coupling layer to allow sufficient information propagation. The flow architecture is the same for both ProHMR and the probabilistic version of Martinez *et al.* [83]. The only difference is that for ProHMR the latent dimension is $d = 6 \cdot 24 = 144$, whereas for the 2D skeleton lifting is $d = 16 \cdot 3 = 48$. We use 16 joints instead of 17, since the pelvis location is always at $(0, 0, 0)$.

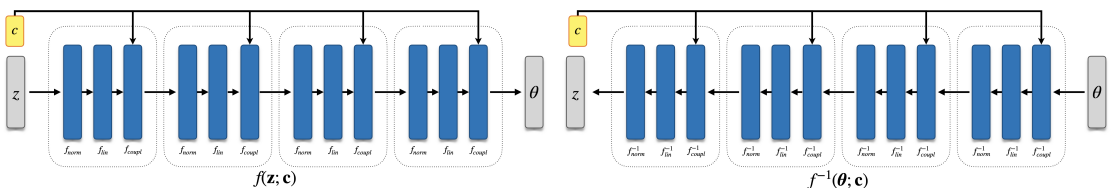

Figure 33: Normalizing Flow architecture. The figure shows the implementation of $f(\boldsymbol{\theta}; \mathbf{c})$ and its inverse using Normalizing Flows. Left: Behavior of our model in the sampling phase (map $\mathbf{z} \rightarrow \boldsymbol{\theta}$). Right: Behavior of our model in the likelihood evaluation phase (map $\boldsymbol{\theta} \rightarrow \mathbf{z}$).

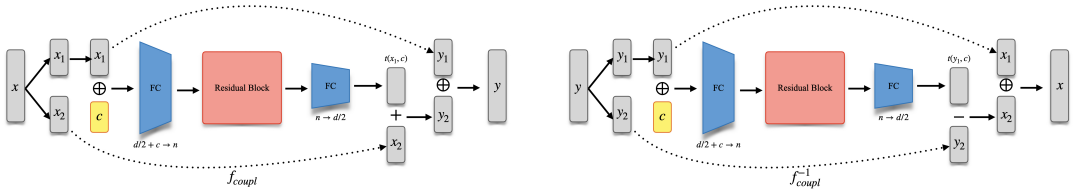

Figure 34: Coupling layer architecture. The figure shows the implementation of f_{coupl} and its inverse. Left: Behavior of our model in the forward phase. Right: Behavior of our model in the inverse phase.

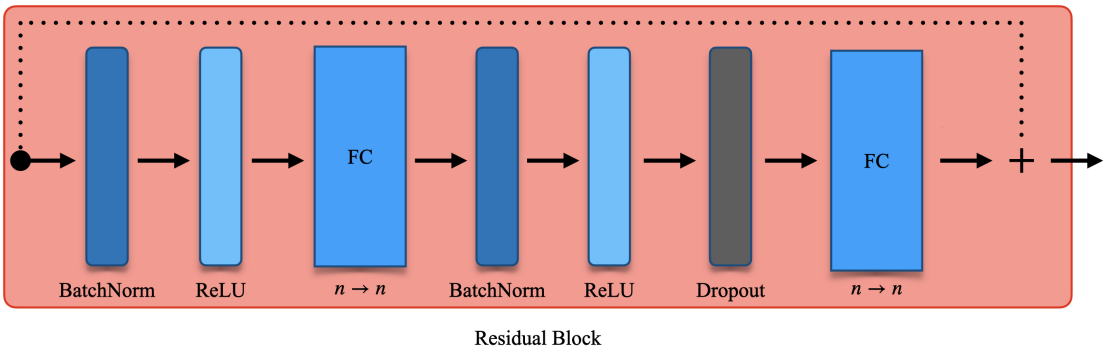

Figure 35: Residual architecture. The figure shows the implementation of the residual block used in the coupling layers f_{coupl} and its inverse.

4.6.3 Training details

We implemented our model using PyTorch [96]. Our Normalizing Flow implementation was based on the `nflows` package [22].

ProHMR We trained our model for 500K iterations with a batch size of 64 using the Adam optimizer [57] with learning rate 0.0001 and weight decay 0.0001. Training takes about 2.5 days on a NVIDIA RTX2080Ti using mixed precision. For the final loss and loss weights, we use a more fine-grained distinction than the generic described in the main manuscript. The final training loss is written as:

$$\begin{aligned} L = & \lambda_{nll} L_{nll} \\ & + \lambda_{exp,2D} L_{exp,2D} + \lambda_{exp,adv} L_{exp,adv} \\ & + \lambda_{mode,2D} L_{mode,2D} + \lambda_{mode,adv} L_{mode,adv} \\ & + \lambda_{mode,\theta} L_{mode,\theta} + + \lambda_{mode,\beta} L_{mode,\beta} \\ & + \lambda_{mode,3D} L_{mode,3D} + \lambda_{orth} L_{orth}, \end{aligned}$$

where $L_{mode,*}$ and $L_{exp,*}$ refer to the corresponding terms included in the loss functions. More specifically, $L_{mode,2D}$ and $L_{exp,2D}$ are the reprojection loss for the mode and the expected reprojection loss respectively, $L_{mode,\theta}$ and $L_{mode,\beta}$ and $L_{mode,3D}$ are penalties on the SMPL parameters and 3D joint locations and $L_{mode,adv}$ and $L_{exp,adv}$ are the adversarial priors. We use ℓ_1 penalty for the losses on the 2D and 3D keypoints and ℓ_2 penalty for the loss on the SMPL parameters. For all losses we sum over all dimensions and then divide by the batch size. The loss weights are: $\lambda_{prob} = 0.001$, $\lambda_{exp,2D} = 0.001$, $\lambda_{exp,adv} = \lambda_{mode,adv} = 0.0005$, $\lambda_{mode,2D} = 0.01$, $\lambda_{mode,3D} = 0.05$, $\lambda_{mode,\theta} = 0.001$, $\lambda_{mode,\beta} = 0.0005$ and $\lambda_{orth} = 0.1$.

2D pose lifting We trained our model for 300K iterations with a batch size of 64 using the Adam optimizer [57] with learning rate 0.0001 and no weight decay. Training takes about 7 hours on a NVIDIA RTX2080Ti. Since we have full 3D supervision, the final loss function is:

$$L = \lambda_{nll} L_{nll} + \lambda_{mode,3D} L_{mode,3D},$$

with loss weights $\lambda_{prob} = 0.001$ and $\lambda_{mode,3D} = 1.0$.

4.6.4 Datasets

In this part we give a short description of the datasets used for training and evaluation. The dataset that we use are Human3.6M [40], MPI-INF-3DHP [84], 3DPW [140], MPII [3], COCO [72] and Mannequin Challenge [72].

Human3.6M It is a studio-captured benchmark for 3D human pose estimation. It includes several different actions performed by various subjects We follow standard practices in the literature and use subjects S1, S5, S6, S7, S8 for training and subjects S9 and S11 for evaluation.

MPI-INF-3DHP It contains data captured primarily in indoor studio environments and the 3D pose data is captured using a marker-less setup. We use the predefined training and testing splits for training and evaluation respectively.

3DPW It is a dataset captured in a variety of indoor and outdoor locations and uses IMU sensors combined with a 2D pose detector to compute pose and shape ground truth. Following standard practice, we use this dataset only for evaluation in the predefined test split.

MPII It is a dataset containing images of people annotated with 2D keypoint locations. We use this dataset for training.

COCO It is a large scale dataset used among other applications for object detection, segmentation and pose estimation. We use 2D keypoint annotations from the `train2014` split to train our model.

Mannequin Challenge It is a dataset of videos of people staying frozen in diverse natural poses. We use the SMPL annotations generated by [69] and employ the entire dataset (train, test, validation) for evaluation.

4.6.5 Additional evaluations

Evaluation metrics

Here we give an outline of the metrics used for evaluation. To evaluate the 3D pose we use the Mean Per Joint Position Error (MPJPE) which computes the mean Euclidean error between the predicted and ground truth joints, after aligning the two poses at the pelvis. With PA-MPJPE we refer to the error after aligning the prediction with the ground truth pose by performing Procrustes alignment.

ProHMR Consistently with HMR [52], CMR [64] and SPIN [63], for evaluating the models that predict SMPL parameters we report the error on the 14 common LSP joints, except for MPI-INF-3DHP where we use all 17 joints provided by the dataset. For Human3.6M with the exception of the multiview experiment (Table 4 of the main manuscript) we evaluate using the frontal camera, whereas for the multiview experiment we use all available cameras.

2D pose lifting To evaluate on Human3.6M we report MPJPE and PA-MPJPE computed on the 17 body joints and use frames from all available cameras.

Additional details

We clarify that for the model fitting experiment (Table 3 of the main manuscript) "H36M (OP)" refers to fitting the SMPL model to the OpenPose detectioins, whereas for "H36M (GT)" we use the ground truth 2D keypoints. Also, when we refer to the *minimum* error of samples from the learned distribution (Table 2 of the main manuscript), we use $n = 4096$ samples. For results with smaller n, we sample $n = 4096$ random poses from the posterior and following [6], we "quantize" them using K-Means. We would also like to highlight that theoretically our model can generate an arbitrary number of samples from the posterior, whereas for the multi-head architecture of Biggs *et al*. [6], increasing the maximum number of proposals requires a linear increase in the model size as well as retraining a new model.

Quantitative evaluation

It is interesting to study how the minimum error of our method scales with the number of samples. Figure 36 shows the minimum PA-MPJPE with respect to the number of samples drawn for the 2D pose lifting network. We can see that the minimum error for our method decreases almost linearly in log-scale. At the same time we show that sampling a large number of poses from our learned posterior achieves significantly lower error than sampling the same number of poses from the training distribution.

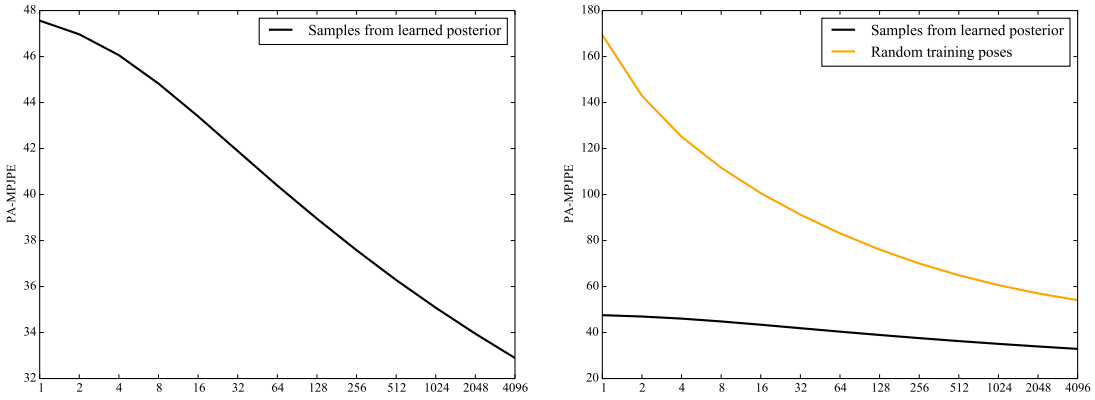

Figure 36: The effect of sampling on 3D errors. We report the minimum PA-MPJPE on Human3.6M for different number of samples. To eliminate the effect of extra data, we report results for the 2D pose lifting network [83] trained on Human3.6M. Left: Error vs number of samples from the learned posterior $p(\boldsymbol{\theta}; \mathbf{c})$. Right: Comparison with drawing an equal number of random poses from the training set.

In a similar setting, we compare the minimum error of ProHMR using 4096 samples with drawing an equal number of samples from a Gaussian distribution centered around the prediction. While ProHMR achieves a minimum PA-MPJPE of 29.9mm, drawing 4096 samples from $\mathcal{N}(\boldsymbol{\theta}^*, \sigma I)$ results in a minimum PA-MPJPE of $35.2, 41.1, 42.2$ for $\sigma = 1, 10, 20$ cm respectively. This is in part related to the fact that Gaussian posteriors are unimodal. Additionally, sampling directly from high dimensional Gaussians is known to be problematic.

Finally, we also provide another quantitative comparison. We use the AH36M dataset (ambiguous version of Human36M [40]) from Biggs *et al.* [6] and compare directly with the result of Table 1 from their paper, reporting the full results in Table 17. Again, our approach

	$n = 1$	$n = 25$
Biggs *et al.* [6]	67.8	64.2
ProHMR	67.3	60.1

Table 17: Comparison with the approach of Biggs *et al.* [6] on their AH36M dataset. Numbers are PA-MPJPE in mm.

outperforms the baseline of [6] in this comparison, particularly when increasing the number of samples n. Furthermore, we assess the effect of our method in the downstream task of fitting on AH36M. Even under the truncations of this dataset, our image-based fitting outperforms SMPLify, with SMPLify achieving a PA-MPJPE of 67.8mm, while our fitting version achieves 61.4mm for the same metric.

Qualitative Results

In Figure 40 we show additional reconstructions of ProHMR. We use pink color for the mode of the posterior distribution. Moreover, in Figure 42 we include additional comparisons between our fitting method and SMPLify initialized by our regression. An important observation is that by using our image-based prior, the body orientation after the fitting is significantly more accurate compared to SMPLify, especially in cases with truncated people.

In Figure 37 we show how the optimization-based pose refinement is able to get more accurate pose estimates by fusing information from multiple views. In the first view the hands are mostly occluded and the recovered pose is not very accurate, however after the joint pose refinement the consolidated pose captures the true pose more faithfully.

Additionally, Figure 41 depicts examples of performing interpolation in the learned latent space. Starting from $\mathbf{z} = 0$, we pick two random directions in $\mathbb{R}^d$ and then move along those directions. Please note that there are no semantics or explicit disentanglement in the latent space.

Finally, we highlight some failure cases of our method. First, in Figure 38 we show failure cases for the regression network. Remarkably though, if we have access to accurate 2D keypoint detections, then it is possible to recover from such errors using our image-based

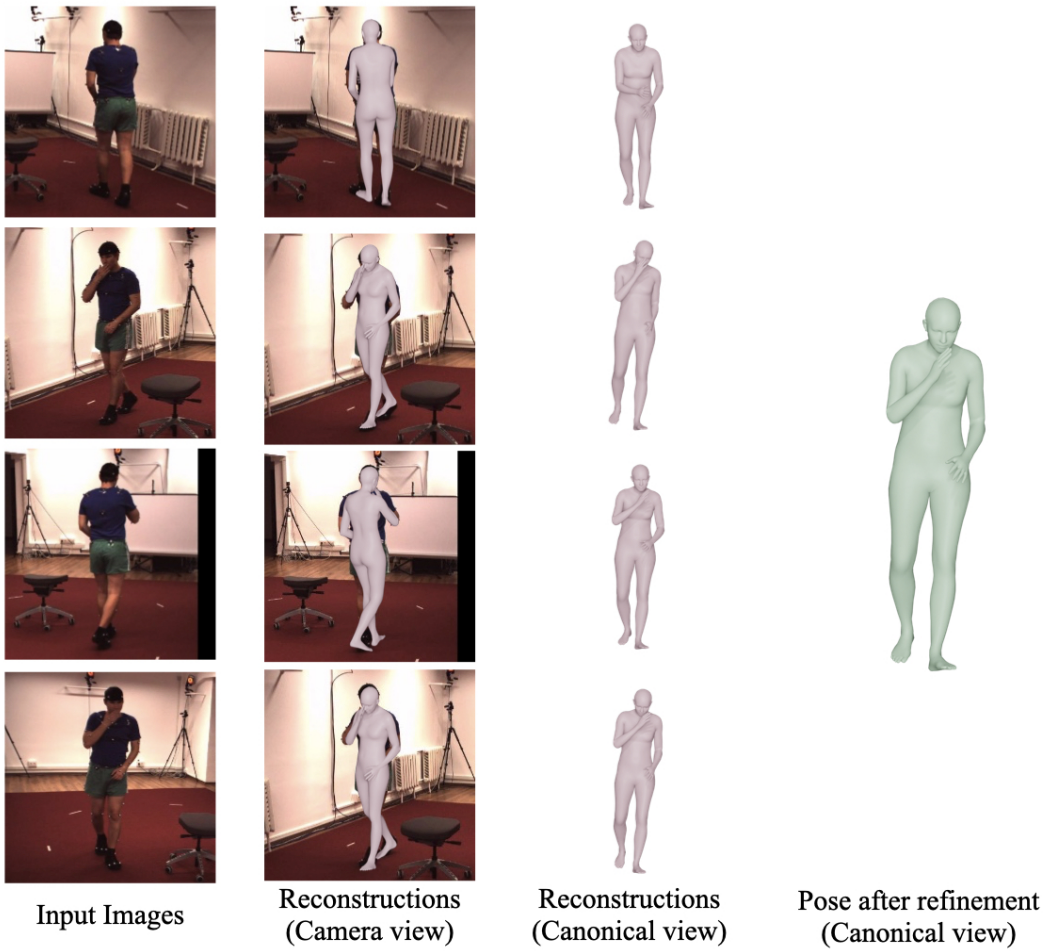

Figure 37: Multiview refinement. Pink: Regression. Green: Multiview refinement. Fitting with multiple views fixes the position of the right hand.

model fitting. However, the model fitting can in turn fail when there are wrong keypoint detections with high confidence, or when there are too few detected keypoints as we show in Figure 39.

Figure 38: Failure cases of pose regression. Some failure cases of the regression (pink mesh) in challenging poses. In these examples, the model fitting (green mesh) is able to improve the pose reconstruction

Figure 39: Failure cases for the model fitting. The optimization can fail if there are wrong keypoint detections with high confidence (rows 1 and 2) or very few detected keypoints (row 3).

Figure 40: Samples from the learned distribution. The pink colored mesh corresponds to the mode whereas we use purple and yellow for the additional samples.

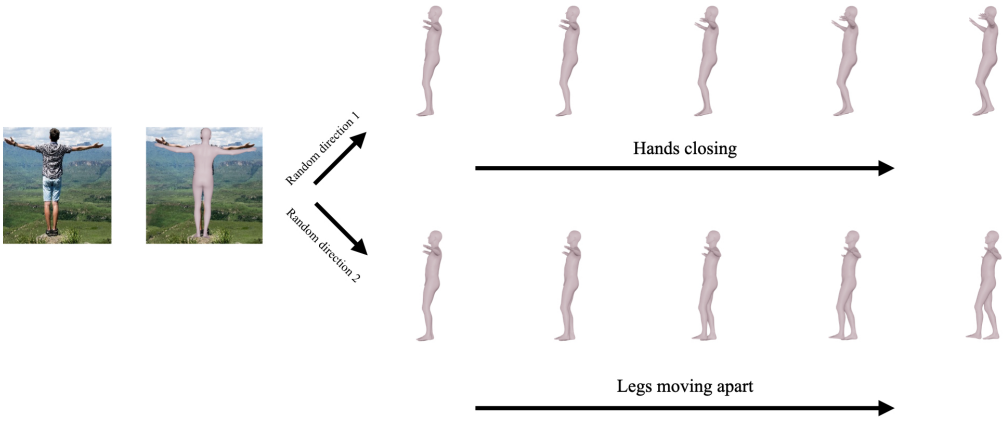

Figure 41: Interpolation in the latent space. We pick two random directions in the latent space and visualize the transformed samples on each direction from a side view.

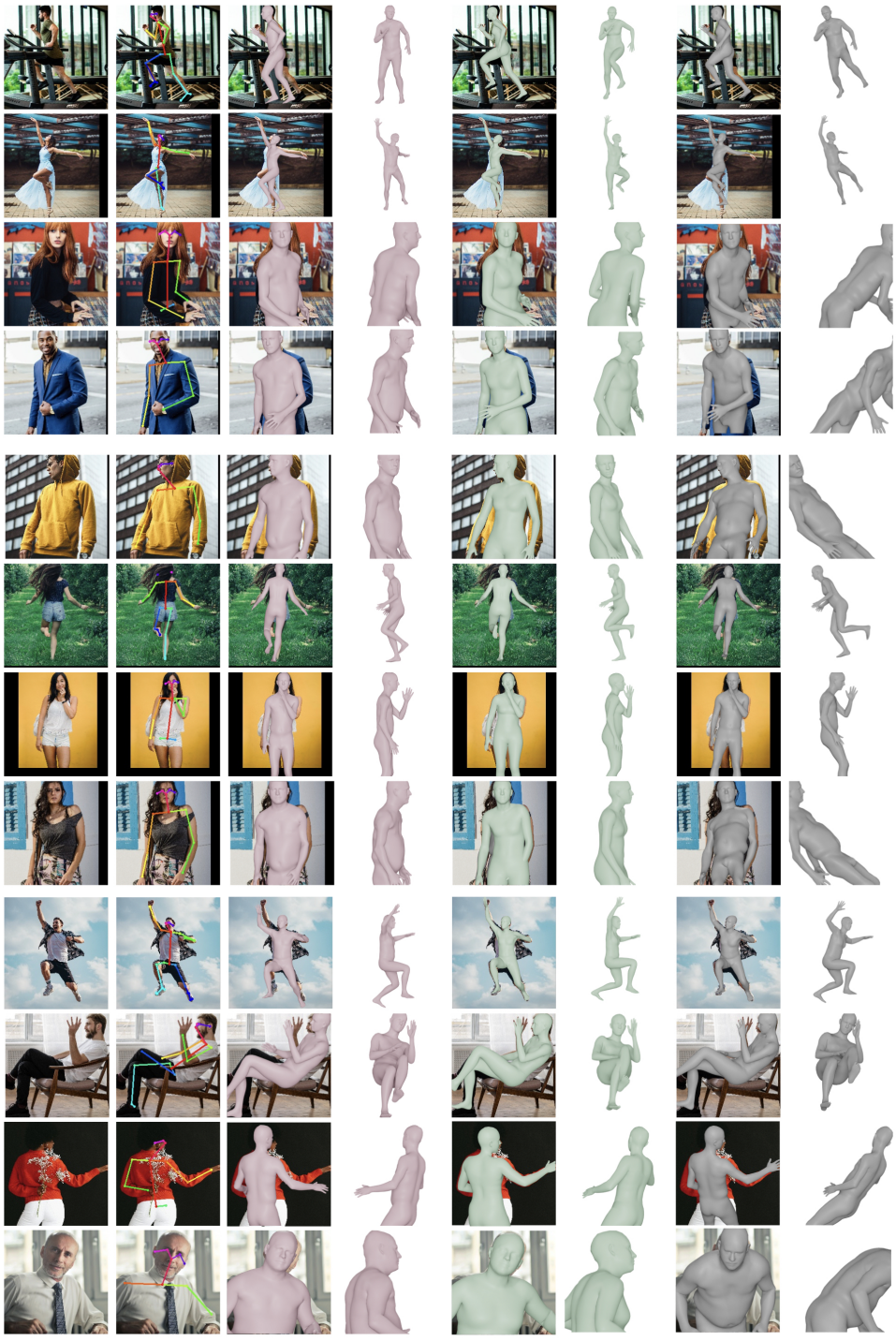

Figure 42: Model Fitting results. Pink: Regression. Green: ProHMR + fitting. Grey: Regression + SMPLify

5.1 Introduction

Figure 43: Coherent reconstruction of pose and shape for multiple people. Typical top-down regression baselines (center) suffer from predicting people in overlapping positions, or in inconsistent depth orderings. Our approach (right) is trained to respect all these constraints and recover a coherent reconstruction of all the people in the scene in a feedforward manner.

Recent work has achieved tremendous progress on the frontier of 3D human analysis tasks. Current approaches have established impressive performance for 3D keypoint estimation [83, 126], 3D shape reconstruction [25, 138], full-body 3D pose and shape recovery [32, 52, 63, 98], or even going beyond that and estimating more detailed and expressive reconstructions [97, 147]. However, as we progress towards more holistic understanding of scenes and people interacting in them, a crucial step is the coherent 3D reconstruction of multiple people from single images.

Regarding multi-person pose estimation, on one end of the spectrum, we have bottom-up approaches. The works following this paradigm, first detect all body joints in the scene and then group them, i.e., assigning them to the appropriate person. However, it is not straightforward how bottom-up processing can be extended beyond joints (e.g., use it for shape estimation, or mesh recovery). Different from bottom-up, top-down approaches first detect all people in the scene, and then estimate the pose for each one of them. Although they take a hard decision early on (person detection), they typically rely on state-of-the-art methods for person detection and pose estimation which allows them to achieve very compelling results, particularly in the 2D pose case, e.g., [15, 124, 148]. However, when reasoning about the pose of multiple people in 3D, the problems can be more complicated than in 2D. For example, the reconstructed people can overlap each other in the 3D space, or be estimated at depths that are inconsistent with the actual depth ordering, as is demon-

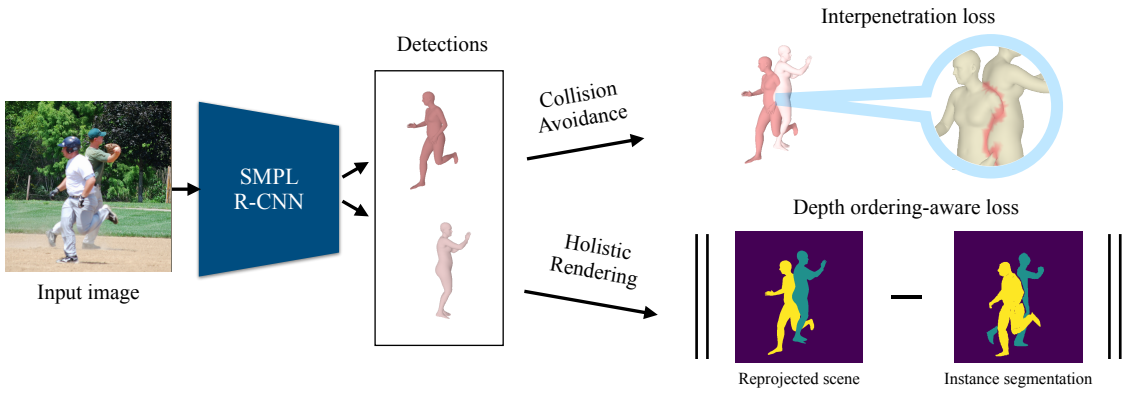

Figure 44: Overview of the proposed approach. We design an end-to-end framework for 3D pose and shape estimation of multiple people from a single image. An R-CNN-based architecture [36] detects all people in the image and estimates their SMPL parameters [78]. During training we incorporate constraints to promote a coherent reconstruction of all the people in the scene. First, we use an interpenetration loss to avoid people overlapping each other. Second, we apply a depth ordering-aware loss by rendering the meshes of all the people to the image and encouraging the rendered instance segmentation to match with the annotated instance masks.

strated in Figure 43. This means that it is crucial to go beyond just predicting a reasonable 3D pose for each person individually, and instead estimate a coherent reconstruction of all the people in the scene.

This coherency of the holistic scene is the primary goal of this work. We adopt the typical top-down paradigm, and our aim is to train a deep network that learns to estimate a coherent reconstruction of all the people in the scene. Starting with a framework that follows the R-CNN pipeline [106], a key decision we make is to use of the SMPL parametric model [78] as our representation, and add a SMPL estimation branch to the R-CNN. The mesh representation provided by SMPL allows us to reason about occlusions and interpenetrations enabling the incorporation of two novel losses towards coherent 3D reconstruction. First, a common problem of predictions from regression networks is that the reconstructed people often overlap each other, since the feedforward nature does not allow for holistic feedback on the potential intersections. To train a network that learns to avoid this type of collisions, we introduce an interpenetration loss that penalizes intersections among the

reconstructed people. This term requires no annotations and relies on a simple property of natural scenes, i.e., that people cannot intersect each other. Besides collisions, another source of incoherency in the results is that the estimated depths of the meshes are not respecting the actual depth ordering of the humans in the scene. Equipped with a mesh representation, we render our holistic scene prediction on the 2D image plane and penalize discrepancies of this rendering from the annotated instance segmentation. This loss enables reasoning about occlusion, encouraging the depth ordering of the people in the scene to be consistent with the annotated instance masks. Our complete framework (Figure 44) is evaluated on various benchmarks and outperforms previous multi-person 3D pose and shape approaches, while the proposed losses improve coherency of the holistic result both qualitatively and quantitatively.

To summarize, our main contributions are:

- We present a complete framework for coherent regression of 3D pose and shape for multiple people.
- We train with an interpenetration loss to avoid regressing meshes that intersect each other.
- We train with a depth ordering-aware loss to promote reconstructions that respect the depth ordering of the people in the scene.
- We outperfrom previous approaches for multi-person 3D pose and shape, while recovering significantly more coherent results.

5.2 Related work

In this Section we provide a short description of prior works that are more relevant to ours.

Single-person 3D pose and shape: Many recent works estimate 3D pose in the form of a skeleton, e.g., [83, 87, 99, 104, 126, 133, 134, 159], or 3D shape in a non-parmetric way, e.g., [25, 120, 138]. However, here we focus on full-body pose and shape reconstruction in

the form of a mesh, typically using a parametric model, like SMPL [78]. After the early works on the problem [31, 117], the first fully automatic approach, SMPLify, was proposed by Bogo *et al*. [7]. SMPLify iteratively fits SMPL on the 2D joints detected by a 2D pose estimation network [103]. This optimization approach was later extended in multiple ways; Lassner *et al*. [67] use silhouettes for the fitting, Varol *et al*. [138] use voxel occupancy grids, while Pavlakos *et al*. [97] fit a more expressive parametric model, SMPL-X.

Despite the success of the aforementioned fitting approaches, recently we have observed an increased interest in approaches that regress the pose and shape parameters directly from images, using a deep network for this task. Many works focus on first estimating some form of intermediate representation before regressing SMPL parameters. Pavlakos *et al*. [101] use keypoints and silhouettes, Omran *et al*. [92] use semantic part segmentation, Tung *et al*. [137] append heatmaps for 2D joints to the RGB input, while Kolotouros *et al*. [64] regress the mesh vertices with a Graph CNN. Regressing SMPL parameters directly from RGB input is more challenging, but it avoids any hand-designed bottleneck. Kanazawa *et al*. [52] use an adversarial prior to penalize improbable 3D shapes during training. Arnab *et al*. [5] use temporal context to improve the regression network. Güler *et al*. [32] incorporate a test-time post-processing based on 2D/3D keypoints and DensePose [33].

Multi-person 3D pose: For the multi-person case, the top-down paradigm is quite popular for 3D pose estimation, since it capitalizes on the success of the R-CNN works [29, 106, 36]. The LCR-Net approaches [110, 111] first detect each person, then classify its pose in a pose cluster and finally regress an offset for each joint. Dabral *et al*. [17] first estimate 2D joints inside the bounding box and then regress 3D pose. Moon *et al*. [88] contribute a root network to give an estimate of the depth of the root joint. Zanfir *et al*. [153] rely on scene constraints to iteratively optimize the 3D pose and shape of the people in the scene. Alternatively, there are also approaches that follow the bottom-up paradigm. Mehta *et al*. [86] propose a formulation based on occlusion-robust pose-maps, where Part Affinity Fields [11] are used for the association problem. Follow-up work [85], improves, among others, the

robustness of the system. Finally, Zanfir *et al*. [154] solve a binary integer linear program to perform skeleton grouping.

In the context of pose and shape estimation in particular, there is a limited number of works that estimate full-body 3D pose and shape for multiple people in the scene. Zanfir *et al*. [153] optimize the 3D shape of all the people in the image using multiple scene constraints. Our approach draws inspiration from this work and shares the same goal, in the sense of recovering a coherent 3D reconstruction. In contrast to them, instead of optimizing for this coherency at test-time, we train a feedforward regressor and use the scene constraints at training time to encourage it to produce coherent estimates at test-time. Using a feedforward network to estimate pose and shape for multiple people has been proposed by the work of Zanfir *et al*. [154]. However, in that case, 3D shape is regressed based on 3D joints, which are the output of a bottom-up system. In contrast, our approach is top-down, and SMPL parameters are regressed directly from pixels, instead of using an intermediate representation, like 3D joints. In fact, it is non-trivial to design a framework for SMPL parameter regression in a bottom-up manner.

Coherency constraints: An important aspect of our work is the incorporation of loss terms that promote coherent 3D reconstruction of the multiple humans. Regarding our interpenetration loss, Bogo *et al*. [7] and Pavlakos *et al*. [97] use a relevant objective to avoid self-interpenetrations of the human under consideration. In a more similar spirit to us, Zanfir *et al*. [153] use a volume occupancy loss to avoid humans intersecting each other. In different applications, Hasson *et al*. [35] penalize interpenetrations between the object and the hand that interacts with it, while Hassan *et al*. [34] penalize interpenetrations between humans and their environment. The majority of the above works uses the interpenetration penalty to iteratively refine estimates at test-time. With the exception of [35], our work is the only one that uses an interpenetration term to guide the training of a feedforward regressor and promote colliding-free reconstructions at test time.

Regarding our depth ordering-aware loss, we follow the formulation of Chen *et al*. [14],

which was also used in the context of 3D human pose by Pavlakos *et al*. [99]. In contrast to them, we do not use explicit depth annotations, but instead, we leverage the instance segmentation masks to reason about occlusion and thus, depth ordering. The work of Rhodin *et al*. [108] is also relevant, where inferring depth ordering is used as an intermediate abstraction for scene decomposition from multiple views. Our work also aims to estimate a coherent depth ordering, but we do so from a single image with the guidance of instance segmentation, while we retain a more explicit human representation in terms of meshes. Finally, using instance segmentation via render and compare has also been proposed by Kundu *et al*. [66]. However, their multi-instance evaluation includes only rigid objects, specifically cars, whereas we investigate the, significantly more complex, non-rigid case.

5.3 Technical approach

In this Section, we describe the technical approach followed in this work. We start with providing some information about the SMPL model (Subsection 5.3.1) and the baseline regional architecture we use (Subsection 5.3.2). Then we describe in detail our proposed losses promoting interpenetration-free reconstruction (Subsection 5.3.3) and consistent depth ordering (Subsection 5.3.4). Finally, we provide more implementation details (Subsection 5.3.5).

5.3.1 SMPL parametric model

For the human body representation, we use the SMPL parametric model of the human body [78]. What makes SMPL very appropriate for our work, in comparison with other representations, is that it allows us to reason about occlusion and interpenetration enabling the use of the novel losses we incorporate in the training of our network. The SMPL model defines a function $\mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta})$ that takes as input the pose parameters $\boldsymbol{\theta}$, and the shape parameters $\boldsymbol{\beta}$, and outputs a mesh $M \in \mathbb{R}^{N_v \times 3}$, consisting of $N_v = 6890$ vertices. The model also offers a convenient mapping from mesh vertices to k body joints J, through a linear regressor W, such that joints can be expressed as a linear combination of mesh vertices, $J = WM$.

5.3.2 Baseline architecture

In terms of the architecture for our approach, we follow the familiar R-CNN framework [106], and use a structure that is most similar to the Mask R-CNN iteration [36]. Our network consists of a backbone (here ResNet50 [37]), a Region Proposal Network, as well as heads for detection and SMPL parameter regression (SMPL branch). Regarding the SMPL branch, its architecture is similar to the iterative regressor proposed by Kanazawa *et al.* [52], regressing pose and shape parameters, $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ respectively, as well as camera parameters $\boldsymbol{\pi} = \{s, t_x, t_y\}$. The camera parameters are predicted per bounding box but we later update them based on the position of the bounding box in the full image (details in the Sup.Mat.). Although there is no explicit feedback among bbox predictions, the receptive field of each proposal includes the majority of the scene. Since each bounding box is aware of neighboring people and their poses, it can make an *informed pose prediction* that is consistent with them.

For our baseline network, the various components are trained jointly in an end-to-end manner. The detection task is trained according to the training procedure of [36], while for the SMPL branch, the training details are similar to the ones proposed by Kanazawa *et al.* [52]. In the rare cases that 3D ground truth is available, we apply a loss, L_{3D}, on the SMPL parameters and the 3D keypoints. In the most typical case that only 2D joints are available, we use a 2D reprojection loss, L_{2D}, to minimize the distance between the ground truth 2D keypoints and the projection of the 3D joints, J, to the image. Additionally, we also use a discriminator and apply an adversarial prior L_{adv} on regressed pose and shape parameters, to encourage the output bodies to lie on the manifold of human bodies. Each of the above losses is applied independently to each proposal, after assigning it to the corresponding ground truth bounding box. More details about the above loss terms and the training of the baseline model are included in the Sup.Mat.

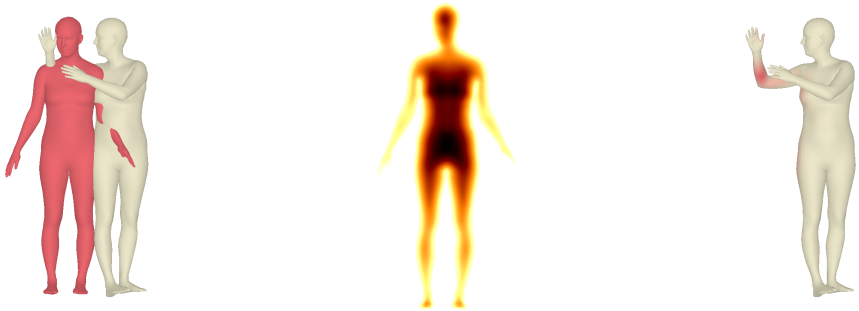

Figure 45: Illustration of interpenetration loss. Left: Collision between person i (red) and j (beige). Center: Distance field ϕ_i for person i, Right: Mesh M_j of person j. The vertices of M_j that collide with person i, i.e., located in non-zero areas of ϕ_i and visualized with soft red, are penalized by the interpenetration loss.

5.3.3 Interpenetration loss

A critical barrier towards coherent reconstruction of multiple people from a single image is that the regression network can often predict the people to be in overlapping locations. To promote prediction of non-colliding people, we introduce a loss that penalizes interpenetrations among the reconstructed people. Our formulation draws inspiration from [34]. An important difference is that instead of a static scene and a single person, our scene includes multiple people and it is generated in a dynamic way during training.

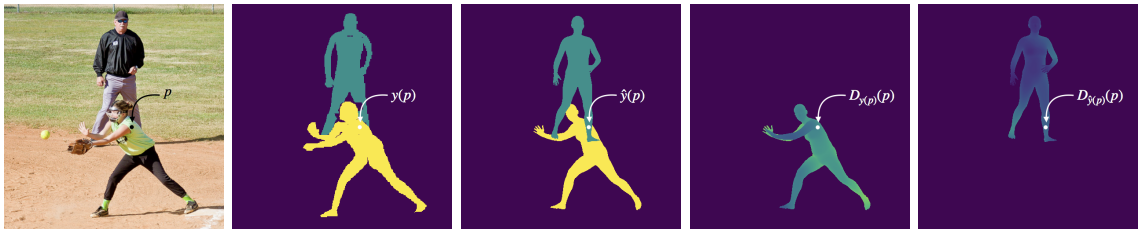

Figure 46: Illustration of depth ordering-aware loss. For an RGB image (first image), we consider the annotated instance segmentation (second image), and the instances based on the rendering of the estimated meshes on the image plane (third image). In case that there is a disagreement between the person index, e.g., for pixel p, where $y(p) \neq \hat{y}(p)$, we penalize the corresponding depth estimates at this pixel with an ordinal depth loss. The pixel depths $D_{y(p)}(p)$ and $D_{\hat{y}(p)}(p)$ are estimated by rendering the depth map independently for each person mesh (fourth and fifth image). This allows gradients to be backpropagated even to the non-visible vertices.

Let ϕ be a modified Signed Distance Field (SDF) for the scene that is defined as follows:

$$\phi(x, y, z) = -\min\left(\mathrm{SDF}(x, y, z), 0\right), \tag{5.1}$$

According to the above definition, inside each human, ϕ takes positive values, proportional to the distance from the surface, while it is simply 0 outside of the human. Typically, ϕ is defined on a voxel grid of dimensions $N_p \times N_p \times N_p$. The naïve generation of a single voxelized representation for the whole scene is definitely possible. However, we often require a very fine voxel grid, which depending on the extend of the scene, might make processing intractable in terms of memory and computation. One critical observation here is that we can compute a separate ϕ_i function for each person in the scene, by calculating a tight box around the person and voxelizing it. This allows us to ignore empty scene space that is not covered by any person and we can instead use a fine spatial resolution to get a detailed voxelization of the body. Using this formulation, the collision penalty of person j for colliding with person i is defined as:

$$\mathcal{P}_{ij} = \sum_{v \in M_j} \tilde{\phi}_i(v), \tag{5.2}$$

where $\tilde{\phi}_i(v)$ samples the ϕ_i value for each 3D vertex v in a differentiable way from the 3D grid using trilinear interpolation (Figure 49). The ϕ_i computation for person i is performed by a custom GPU implementation. This computation does not have to be differentiable; ϕ_i only defines a distance field from which we sample values in a differentiable way. By definition, $\mathcal{P}_{ij}$ is non-negative. It takes value 0 if there is no collision between person i and j and increases as the distance of the surface vertices for person j move farther from the surface of person i. In theory, $\mathcal{P}_{ij}$ can be used by itself as an optimization objective for interpenetration avoidance. However, in practice, we observed that it results in very large gradients for the person translation, leading to training instabilities when there are heavy collisions. Instead of the typical term, we use a robust version of this objective. More

specifically, our final interpenetration loss for a scene with N people is defined as follows:

$$L_{\mathcal{P}} = \sum_{j=1}^{N} \rho \left(\sum_{i=1, i \neq j}^{N} \mathcal{P}_{ij} \right) \tag{5.3}$$

where ρ is the Geman-McClure robust error function [26]. To avoid penalizing intersections between boxes corresponding to the same person, we use only the most confidence box proposal assigned to a ground truth box.

5.3.4 Depth ordering-aware loss

Besides interpenetration, another common problem in multi-person 3D reconstruction is that people are often estimated in incorrect depth order. This problem is more evident in cases where people overlap on the 2D image plane. Although it is obvious to the human eye which person is closer (due to the occlusion), the network predictions can still be incoherent. Fixing this depth ordering problem would be easy if we had access to pixel-level depth annotations. However, this type of annotations is rarely available. Our key idea here is that we can leverage the instance segmentation annotations that are often available, e.g., in the large scale COCO dataset [73]. Rendering the meshes of all the reconstructed people on the image plane can indicate the person corresponding to each pixel and optimize based on the agreement with the annotated instance annotation.

Although this idea sounds straightforward, its realization is more complicated. An obvious implementation would be to use a differentiable renderer, e.g., the Neural Mesh Renderer (NMR) [56], and penalize inconsistencies between the actual instance segmentation and the one produced by rendering the meshes to the image. The practical problem with [56] is that it backpropagates errors only to visible mesh vertices; if there is a depth ordering error, it will not promote the invisible vertices to move closer to the camera. In practice, we observed that this tends to move most people farther away, collapsing our training. Liu *et al.* [76] attempt to address this problem, but we observed that their softmax operation across the depths can result in vanishing gradients, while we also faced numerical instabilities.

Instead of rendering only the semantic segmentation of the scene, we also render the depth image D_i for each person independently using NMR [56]. Assuming the scene has N people, we assign a unique index $i \in \{1, 2, \ldots, N\}$ to each one of them. Let $y(p)$ be the person index at pixel location p in the ground truth segmentation, and $\hat{y}(p)$ be the predicted person index based on the rendering of the 3D meshes. We use 0 to indicate background pixels. If for a pixel p the two estimates indicate a person (no background) and disagree, i.e., $y(p) \neq \hat{y}(p)$, then we apply a loss to the depth values of both people for this pixel, $y(p)$ and $\hat{y}(p)$, to promote the correct depth ordering. The loss we apply is an ordinal depth loss, similar in spirit to [14]. More specifically, the complete loss expression is:

$$L_{\mathcal{D}} = \sum_{p \in \mathcal{S}} \log\left(1 + \exp\left(D_{y(p)}(p) - D_{\hat{y}(p)}(p)\right)\right) \tag{5.4}$$

where $\mathcal{S} = \{p \in I : y(p) > 0, \hat{y}(p) > 0, y(p) \neq \hat{y}(p)\}$ represents the set of pixels for image I where we have depth ordering mistakes (Figure 46). The key detail here is that the loss is backpropagated to the mesh (and eventually the model parameters) of both people, instead of backpropagating gradients only to the visible person, as a conventional differentiable renderer would do. This promotes a more symmetric nature to the loss (and the updates), and eventually makes this loss practical.

5.3.5 Implementation details

Our implementation is done using PyTorch and the publicly available mmdetection library [12]. We resize all input images to 512x832, keeping the same aspect ratio as in the original COCO training. For the baseline model we train only with the losses specified in Subsection 5.3.2, while for our full model we include in our training the losses proposed in Subsections 5.3.3 and 5.3.4. Our training uses 2 1080Ti GPUs and a batch size of 4 images per GPU.

For the SDF computation, we reimplemented [122, 123] in CUDA. Voxelizing a single mesh in a $32 \times 32 \times 32$ voxel grid requires about 45ms on an 1080Ti GPU. For efficiency, we perform 3D bounding box checks to detect overlapping 3D bounding boxes, and voxelize only the

relevant meshes. Additionally, we reimplemented parts of NMR [56] to make rendering large images more efficient. This allowed us to have more than an order of magnitude of speedup since the forward pass complexity dropped from $O(Fwh)$ to $O(F + wh)$ on average, where F is the number of faces and w and h the image width and height respectively.

5.4 Experiments

In this Section, we present the empirical evaluation of our approach. First, we describe the datasets used for training and evaluation (Subsection 5.4.1). Then, we focus on the quantitative evaluation (Subsections 5.4.2 and 5.4.3), and finally we present more qualitative results (Subsection 5.4.4).

5.4.1 Datasets

Human3.6M [40]: It is an indoor dataset where a single person is visible in each frame. It provides 3D ground truth for training and evaluation. We use Protocol 2 of [52], where Subjects S1,S5,S6,S7 and S8 are used for training, while Subjects S9 and S11 are used for evaluation.

MuPoTS-3D [86]: It is a multi-person dataset providing 3D ground truth for all the people in the scene. We use this dataset for evaluation using the same protocol as [86].

Panoptic [48]: It is a dataset with multiple people captured in the Panoptic studio. We use this dataset for evaluation, following the protocol of [153].

MPI-INF-3DHP [84]: It is a single person dataset with 3D pose ground truth. We use subjects S1 to S8 for training.

PoseTrack [2]: In-the-wild dataset with 2D pose annotations. Includes multiple frames for each sequence. We use this dataset for training and evaluation.

LSP [46], **LSP Extended** [47], **MPII** [3]: In-the-wild datasets with annotations for 2D joints. We use the training sets of these datasets for training.

COCO [73]: In-the-wild dataset with 2D pose and instance segmentation annotations. We

Method	HMR [52]	Arnab *et al.* [5]	Ours
Reconst. Error	56.8	54.3	**52.7**

Table 18: Results on Human3.6M. The numbers are mean 3D joint errors in mm after Procrustes alignment (Protocol 2). The results of all approaches are obtained from the original papers.

use the 2D joints for training as we do with the other in the-wild datasets, while the instance segmentation masks are employed for the computation of the depth ordering-aware loss.

5.4.2 Comparison with the state-of-the-art

For the comparison with the state-of-the-art, as a sanity check, we first evaluate the performance of our approach on a typical single person baseline. Our goal is always multi-person 3D pose and shape, but we expect our approach to achieve competitive results, even in easier settings, i.e., when only one person is in the image. More specifically, we evaluate the performance of our network on the popular Human3.6M dataset [40]. The most relevant approach here is HMR by Kanazawa *et al.* [52], since we share similar architectural choices (iterative regressor, regression target), training practices (adversarial prior) and training data. The results are presented in Table 18. Our approach outperforms HMR, as well as the approach of Arnab *et al.* [5], that uses the same network with HMR, but is trained with more data.

Having established that our approach is competitive in the single person setting, we continue the evaluation with multi-person baselines. In this case, we consider approaches that also estimate pose and shape for multiple people. The most relevant baselines are the works of Zanfir *et al.* [153, 154]. We compare with these approaches in the Panoptic dataset [48, 50], using their evaluation protocol (assuming no data from the Panoptic studio are used for training). The full results are reported in Table 19. Our initial network (baseline), trained without our proposed losses, achieves performance comparable with the results reported by the previous works of Zanfir *et al.* More importantly though, adding the two proposed losses (full), improves performance across all subsequences and overall, while we also outperform

Method	Haggling	Mafia	Ultim.	Pizza	Mean
Zanfir *et al.* [153]	140.0	165.9	150.7	**156.0**	153.4
Zanfir *et al.* [154]	141.4	152.3	**145.0**	162.5	150.3
Ours (baseline)	141.2	140.3	160.7	156.8	149.8
Ours (full)	**129.6**	**133.5**	153.0	156.7	**143.2**

Table 19: Results on the Panoptic dataset. The numbers are mean per joint position errors after centering the root joint. The results of all approaches are obtained from the original papers.

the previous baselines. These results demonstrate both the strong performance of our approach in the multi-person setting, as well as the benefit we get from the losses we propose in this work.

Another popular benchmark for multi-person 3D pose estimation is the MuPoTS-3D dataset [84]. Since no multi-person 3D pose and shape approach reports results on this benchmark, we implement two strong top-down baselines, based on state-of-the-art approaches for single-person 3D pose and shape. Specifically, we select a regression approach, HMR [52], and an optimization approach, SMPLify-X [97], and we apply them on detections provided by OpenPose [10] (as is suggested by their public repositories), or by Mask-RCNN [36] (for the case of HMR). The full results are reported in Table 20. As we can see, our baseline model performs comparably to the other approaches, while our full model trained with the proposed losses improves significantly over the baseline. Similarly with the previous results, this experiment further justifies the use of our coherency losses. Besides this, we also demonstrate that naïve baselines trained with a single person in mind are suboptimal for the multi-person setting of 3D pose. This is different from the 2D case, where a single-person network can perform particularly well in multi-person top-down pipelines as well, e.g., [15, 124, 148]. For the 3D case though, when multiple people are involved, making the network aware of occlusions and interpenetrations during training, can actually be beneficial at test-time too.

Method	All	Matched
OpenPose + SMPLify-X [97]	62.84	68.04
OpenPose + HMR [52]	66.09	70.90
Mask-RCNN + HMR [52]	65.57	68.57
Ours (baseline)	66.95	68.96
Ours (full)	**69.12**	**72.22**

Table 20: Results on MuPoTS-3D. The numbers are 3DPCK. We report the overall accuracy (All), and the accuracy only for person annotations matched to a prediction (Matched).

Method	MuPoTS-3D	PoseTrack
Our baseline	114	653
Our baseline + $L_{\mathcal{P}}$	34	202

Table 21: Ablative for interpenetration loss. The results indicate the number of collisions on MuPoTS-3D and PoseTrack.

5.4.3 Ablative studies

For this work, our interest in multi-person 3D pose estimation extends beyond just estimating poses that are accurate under the typical 3D pose metrics. Our goal is also to recover a coherent reconstruction of the scene. This is important, because in many cases we can improve the 3D pose metrics, e.g., get a better 3D pose for each detected person, but return incoherent results holistically. For example, the depth ordering of the people might be incorrect, or the reconstructed meshes might be positioned such that they overlap each other. To demonstrate how our proposed losses improve the network predictions under these coherency metrics even if they are only applied during training, we perform two ablative studies for more detailed evaluation.

First, we expect our interpenetration loss to naturally eliminate most of the overlapping people in our predictions. We evaluate this on MuPoTS-3D and PoseTrack, reporting the number of collisions with and without the interpenetration loss. The results are reported in Table 21. As we expected, we observe significant decrease in the number of collisions when we train the network with the $L_{\mathcal{P}}$ loss.

Moreover, our depth ordering-aware loss should improve the translation estimates for the

Method	Moon *et al.* [88]	Our baseline	Our baseline + $L_{\mathcal{D}}$
Accuracy	90.85%	92.17%	93.68%

Table 22: Ablative for depth-ordering-aware loss. Depth ordering results on MuPoTS-3D. We consider all pairs of people in the image, and we evaluate whether the approaches recovered the ordinal depth relation between the two people correctly. The numbers are percentages of correctly estimated ordinal depth relations.

people in the scene. Since for monocular methods it is not meaningful to evaluate metric translation estimates, we propose to evaluate only the returned depth ordering. More specifically, we consider all pairs of people in the scene, and we evaluate whether our method predicted the ordinal depth relation for this pair correctly. In the end, we report the percentage of correctly estimated ordinal relations in Table 22. As expected, the depth ordering-aware loss improves upon our baseline. In the same Table, we also report the results of the approach of Moon *et al.* [88] which is the state-of-the-art for 3D skeleton regression. Although [88] is skeleton-based and thus, not directly comparable to us, we want to highlight that even a state-of-the-art approach (under 3D pose metric evaluation) can still suffer from incoherency in the results. This provides evidence that we often might overlook the coherency of the holistic reconstruction, and we should also consider this aspect when we evaluate the quality of our results.

Finally, we underline that we do not apply these coherency losses at test time. Instead, during training, our losses act as constraints to the reconstruction and ultimately provide better supervision to the network, for images that no explicit 3D annotations are available. The improved supervision leads to more coherent results *at test time too.*

5.4.4 Qualitative evaluation

In this Subsection, we present more qualitative results of our approach. In Figure 47 we compare our baseline with our full model trained with the proposed losses. As expected, our full model generates more coherent reconstructions, improving over the baseline as far as interpenetration and depth ordering mistakes are concerned. Errors can happen when there is significant scale difference among the people and there is no overlap on the image

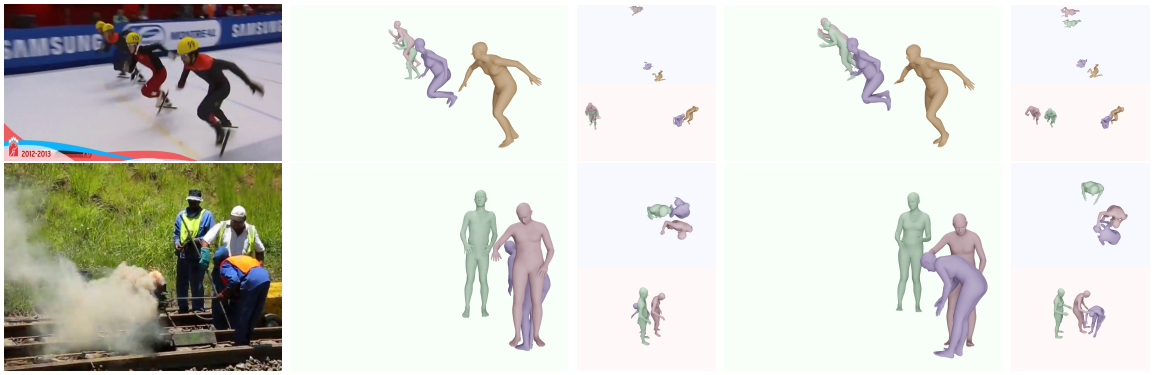

Figure 47: Qualitative effect of proposed losses. Results of our baseline model (center) and our full model trained with our proposed losses (right). As expected, we improve over our baseline in terms of coherency in the results (i.e., fewer interpenetrations, more consistent depth ordering for the reconstructed meshes).

Figure 48: Qualitative evaluation. We visualize the reconstructions of our approach from different viewpoints; front (green background), top (blue background) and side (red background). More qualitative results can be found in the Sup.Mat.

plane (last row of Figure 48). More results can be found in the Sup.Mat.

5.5 Summary

In this work, we present an end-to-end approach for multi-person 3D pose and shape esti-

mation from a single image. Using the R-CNN framework, we design a top-down approach that regresses the SMPL model parameters for each detected person in the image. Our main contribution lies on assessing the problem from a more holistic view and aiming on estimating a coherent reconstruction of the scene instead of focusing only on independent pose estimation for each person. To this end, we incorporate two novel losses in our framework that train the network such that a) it avoids generating overlapping humans and b) it is encouraged to position the people in a consistent depth ordering. We evaluate our approach in various benchmarks, demonstrating very competitive performance in the traditional 3D pose metrics, while also performing significantly better both qualitatively and quantitatively in terms of coherency of the reconstructed scene. In future work, we aim to more explicitly model interactions between people (besides the overlap avoidance), so that we can achieve a more accurate and detailed reconstruction of the scene at a finer level as well. In a similar vein, we can incorporate further information towards a holistic reconstruction of scenes. This can include constraints from the ground plane [153], background [34], or the objects that humans interact with [35, 131].

5.6 Supplementary Material

This supplementary material includes additional details that were not included in the main manuscript due to space constraints. We start with more implementation details (Section 5.6.1). We continue with a short discussion about the effect of the model and the losses in our approach (Section 5.6.2). Then, we provide further results from our quantitative experiments (Section 5.6.3). Finally we extend our qualitative evaluation, including more examples of our approach, including successes, failures and comparisons with the baseline model (Section 5.6.4).

5.6.1 Implementation details

Architecture

Our architecture follows the typical Faster-RCNN pipeline [106], where we add an additional branch for SMPL parameter regression. This branch follows the architecture choices of the

iterative regressor proposed by Kanazawa *et al.* [52]. Ultimately, the output of the SMPL branch includes the estimated pose and shape parameters for the corresponding bounding box, θ and β respectively, as well as the camera parameters $\boldsymbol{\pi} = \{s, x, y\}$. In the original HMR formulation [52], the camera parameters include a scaling factor s, as well as a 2D translation t_x, t_y for a weak perspective camera. However, in order to produce a coherent scene we need to move away from the original weak perspective camera assumption. To do that, we propose a way of converting the camera parameters $\boldsymbol{\pi}$ to the actual translation of each person in the scene.

Let us represent with M_i, $\boldsymbol{\pi}_i = \{s_i, x_i, y_i\}$, the regressed mesh and camera parameters respectively for the ith bounding box B_i in an image I with width w and height h. For each image, we assume we have a single camera located at the origin of the coordinate system with focal length f and its principal point at the center of the image. We underline that the camera parameters we regress are not for weak perspective projection. Instead, we assume a fully perspective camera model, where the focal length f is known. Let $B_i = [x_{min}, y_{min}, x_{max}, y_{max}]$, with center $c_i = [(x_{min} + x_{max})/2, (y_{min} + y_{max})/2]$ and size $\alpha_i = \max(x_{max} - x_{min}, y_{max} - y_{min})$. Given these parameters, the depth of the person is calculated as:

$$d_i = \frac{2f}{s_i \alpha_i} \tag{5.5}$$

Using the computed depth, we then define the person translation as:

$$t_i = \begin{bmatrix} d_i \left(x_i \alpha_i + c_{i,x} - w/2 \right) / f \\ d_i \left(y_i \alpha_i + c_{i,y} - h/2 \right) / f \\ d_i \end{bmatrix} \tag{5.6}$$

The above transformation performs a "coordinate change" from the local, per-bounding box camera to the single global scene camera. This choice ensures that the projection of $\hat{t}_i = [x_i, y_i, d_i]$ given a camera with principal point at the center of the bounding box, projects to the same point, as t_i given a camera with principal point in the image center.

Intuitively, the SMPL branch predicts camera parameters for each box independently. These parameters are relative to the bounding box size, because the input to the SMPL head is the 14×14 output of the ROI Align, so they have to be scaled accordingly.

Interpenetration loss

Here we will elaborate more on how the interpenetration loss works in cases where there are collisions between different people. In the main text we defined the interpenetration loss for a scene as:

$$L_{\mathcal{P}} = \sum_{j=1}^{N} \rho \left(\sum_{i=1, i \neq j}^{N} \mathcal{P}_{ij} \right). \tag{5.7}$$

As explained in the main text, the loss for each person is applied at the vertex level; for person j, we penalize all the vertices that lie inside another person i and that penalty specifically is:

$$\mathcal{P}_{ij} = \sum_{v \in M_j} \tilde{\phi}_i(v). \tag{5.8}$$

Because the loss is applied at the vertex level, $\mathcal{P}_{ij}$ is not symmetric. This is depicted with an example showing a collision between two people in Figure 49.

Training strategy

We observed that the tasks of detection and 3D shape reconstruction behave quite differently during training, with the reconstruction branch needing significantly more training iterations than the detection branch. For this reason, before training the full network end-to-end, we pretrained the SMPL head with cropped images for roughly 350K iterations. The pretraining was done using single-person examples from Human3.6M [40], MPI-INF-3DHP [84], COCO [73], LSP [46], LSP Extended [47] and MPII [3]. After this step, training continues with multi-person images for 400k more iterations. For our full model, our proposed losses are also active in this second step, while for the results reported as "baseline", they are not. We trained our full model in 2 1080Ti GPUs with a learning rate of $1e-4$ using the Rectified Adam optimizer [75]. Regarding the weights for the different loss functions, we use 4 for the keypoint reprojection loss, 4 for the 3D keypoint loss, 1 for the loss

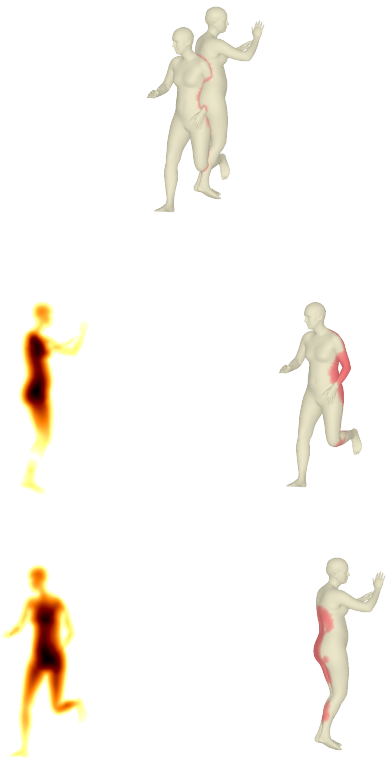

Figure 49: Illustration of interpenetration loss. Top: Collision between two people. Center: Distance field ϕ_2 for person 2 and penalized vertices of person 1. Bottom: Distance field ϕ_1 for person 1 and penalized vertices of person 2.

on the SMPL θ parameters, 1/100 for the loss on the SMPL β parameters, 1/60 for the adversarial prior, 1/100 for the collision loss, 100 for the depth ordering loss, and 1 for the detection and RPN losses.

Before applying the 2D keypoint loss, we normalize the keypoints inside each box proposal by subtracting the box center and dividing by the box width. The other losses of the SMPL branch (3D keypoint loss, loss on the SMPL parameters, loss on the adversarial prior) are computed the same way as in HMR [52].

5.6.2 Effect of model and losses

For the ResNet50 backbone, each neuron has a receptive field of size 483×483 pixels. This means that for a $h \times w$ bounding box, the receptive field is $(h+482) \times (w+482)$. Considering that the input images have resolution 512×832 pixels, for a bounding box, we expect most

of its neighboring people to be within its receptive field. The most interesting scenario is when we have three people, A, B and C, where A overlaps with B, B overlaps with C, but A does not overlap with C. This is challenging, since C might not be "visible" from A and vice versa. In that case, it would be hard to get a coherent prediction for the whole group. To examine how often this occurs, we investigated the statistics of the datasets we used. Specifically we focused in this (A,B,C), scenario, where B can correspond to more than one person, i.e., we can have a longer chain. Considering the receptive field of our architecture, we observed that in most cases C is "visible" from A. Particularly, across all the cases where this (A,B,C) scenario happens, in 88% of cases for Panoptic, in 92% of cases for MuPoTS, and in 91% of cases for PoseTrack, person C is included in the receptive field of A. We expect that with a deeper network, that has a larger receptive field, e.g., ResNet152, these percentages will be even higher.

Regarding our proposed losses, they belong in the category of cross-instance supervision. Cross-instance losses have also been applied successfully in recent works, e.g., [35, 66]. Effectively, during training, to decrease these losses, the network needs to develop features that help avoid coherency errors. The learned features can be related to depth, occlusion, segmentation of the person, etc. Since the losses decrease during training, the network does generate helpful features. More importantly, this translates also to improvement at test time (Tables 4,5 in main manuscript present improvements in *unseen* datasets particularly for collisions and depth ordering). This is a strong indication that the network is not overfitting, but it is indeed learning features that generalize across scenes, and encourage it to make coherent predictions *at test time too*.

5.6.3 Quantitative results

First we provide a more detailed evaluation of our proposed losses on the MuPoTS-3D dataset [86]. We have already reported the results of our baseline and our full model in Table 3 of the main manuscript, but here we extend to a more fine-grained ablative study. The complete results for different versions of our model are presented in Table 23. Based

Method	All	Matched
Our baseline	66.95	68.96
Our baseline + $L_{\mathcal{P}}$	67.84	70.00
Our baseline + $L_{\mathcal{D}}$	66.59	68.43
Our baseline + $L_{\mathcal{P}}$ + $L_{\mathcal{D}}$	**69.12**	**72.22**

Table 23: Ablative on MuPoTS-3D. The numbers are 3DPCK. We report the overall accuracy (All), and the accuracy only for person annotations matched to a prediction (Matched).

on the results, we see that the use of the interpenetration loss alone improves slightly the results over the baseline, while with the depth ordering-aware loss alone we observe a small decrease in the accuracy. However, when we combine the two losses together, we achieve better results, both compared to the baseline, as well as compared to the versions using only one of the two losses alone.

Regarding the comparison with the state-of-the-art in the main manuscript, our evaluation has focused on approaches that estimate 3D pose and shape in the form of the SMPL parametric model [78]. This is common in the literature, where SMPL-based approaches, e.g., [52, 92, 101] do not directly compare with skeleton-based approaches, e.g. [83, 126], and vice versa, because of the different output they provide. Typically, skeleton-based approaches report better quantitative results when compared on metrics using 3D joints, but SMPL-based approaches still output a more informative representation in the form of 3D rotations for each part, making the task harder than only estimating 3D joint locations. Although we are not directly comparable with skeleton-based approaches, we observe that on MuPoTS-3D [86] our approach still performs better than [110, 86], it is competitive to [111] and is underperforming only when it is compared to the most recent baseline, i.e., [88]. However, this comparison is done under the traditional 3D pose metrics, which are computed on 3D joints of individual people only. When the evaluation is performed on a metric that requires coherent estimation for all the people in the scene, e.g., on depth ordering, we observed that even the state-of-the-art approach of Moon *et al*. [88] performs worse than our approach. Concerning the evaluation with the single-person pose and shape

	Method	TS1	TS2	TS3	TS4	TS5	TS6	TS7	TS8	TS9	TS10	TS11	TS12	TS13	TS14	TS15	TS16	TS17	TS18	TS19	TS20	Avg
All	Ours (baseline)	76.42	65.75	**71.59**	66.26	76.89	32.89	**74.01**	67.68	**60.52**	**78.88**	57.81	55.55	64.38	59.68	70.87	75.01	69.84	69.60	**75.19**	70.18	66.95
	Ours (full)	**80.60**	**68.65**	67.02	**68.19**	**77.78**	**38.99**	**74.01**	**67.88**	54.69	77.11	**63.77**	**64.73**	**64.40**	**60.37**	**72.71**	**83.68**	**75.53**	**76.91**	74.40	**70.67**	**69.12**
Matched	Ours (baseline)	76.42	71.91	**71.77**	66.48	79.16	32.92	74.30	68.93	**60.52**	**78.88**	57.81	55.55	66.80	70.80	70.87	75.71	69.95	73.08	76.55	80.89	68.96
	Ours (full)	**80.60**	**76.59**	67.19	**68.42**	**80.24**	**40.33**	**74.71**	**70.77**	54.69	77.11	**64.88**	**64.73**	**67.92**	**72.91**	**72.84**	**85.03**	**75.97**	**81.89**	**78.52**	**89.04**	**72.22**

Table 24: Full results on MuPoTS-3D. The numbers are 3DPCK. We report the overall accuracy (All), and the accuracy only for person annotations matched to a prediction (Matched).

baselines our comparison focuses primarily on HMR [52], that is more similar to us in terms of architecture, training details, and training data. Some more recent approaches, e.g., [32, 63, 98, 150] report improved results on single-person datasets, but they rely on improved training techniques or architectures. These improvements are orthogonal to ours, since we focus on improving the multi-person results, and not the single-person case as they do.

For the evaluation on MuPoTS-3D, we only presented the mean accuracy over all sequences. Here we also provide a more detailed evaluation for each sequence. The complete results are included in Table 24. As we can see, for most sequences, and overall, the version of our model trained with the proposed losses outperforms our baseline.

Besides the above experiments, we also present additional ablatives to clarify the effect of using different datasets to train our system. Similar to Kanazawa *et al.* [52], we use a large set of datasets to train our network, since we observed that this diverse set of images is helpful for better generalization to in-the-wild settings. However, for simpler indoor settings, like Human3.6M [40] and Panoptic [48, 50], using only COCO [73] and Human3.6M [40] for training provides comparable results. To focus on the effect of the data specifically on pose reconstruction, we investigate a simpler setting where we train only the ResNet backbone and the SMPL head, providing ground truth bounding boxes during testing. As we can see in Table 25 for Panoptic and Table 26 for Human3.6M, for these indoor datasets, training with all the data achieves similar performance with training only with Human3.6M and COCO. It is also interesting to observe that using ground truth bounding boxes instead of detections improves performance for Human3.6M, but it hurts performance on Panoptic.

Data	MoSh	Haggling	Mafia	Ultim.	Pizza	Mean
All data	Yes	155.4	178.6	179.7	186.1	175.0
COCO+H36M	Yes	157.5	180.3	178.3	191.7	177.0
COCO+H36M	No	158.7	176.4	175.0	190.4	175.1

Table 25: Ablative on the Panoptic dataset. We focus on the ResNet backbone and the SMPL head (i.e., we use ground truth bounding boxes) and we ablate different training strategies; using all training data (first row), reducing the training data to COCO and Human3.6M datasets only (second row), and abandoning MoSh parameters (third row). All the different versions have comparable results.

Data	MoSh	Reconst. Error
All data	Yes	48.6
COCO + H36M	Yes	50.5
COCO + H36M	No	51.4

Table 26: Ablative on Human3.6M dataset. We focus on the ResNet backbone and the SMPL head (i.e., we use ground truth bounding boxes) and we ablate different training strategies; using all training data (first row), reducing the training data to COCO and Human3.6M datasets only (second row), and abandoning MoSh parameters (third row). The different versions have comparable results. The numbers are mean 3D joint errors in mm after Procrustes alignment (Protocol 2).

This can be attributed to the fact that Panoptic has many truncated human instances, so learning to jointly crop the most informative bounding box along with reconstructing the person can be beneficial compared to be given an arbitrary bounding box at test time.

Additionally, we also ablate the type of supervision we use for Human3.6M. Similar to [52], we use SMPL parameters provided by fitting SMPL to surface markers through MoSh [77]. To see if we can relax this constraint, we also use SMPL parameters provided by fitting SMPL to Human3.6M 3D keypoints, using a procedure similar to SMPLify [7]. Again the results are comparable (Tables 25 and 26), which means that our performance does not rely explicitly on the availability of MoSh parameters.

5.6.4 Qualitative results

For our qualitative evaluation, in Figure 50, we provide more comparisons between our baseline model and our full model trained with our proposed losses. Then, in Figures 51 and 52 we provide more successful reconstructions from the datasets we use in our evalua-

tion. Finally, in Figure 53 we present some representative failure cases of our approach.

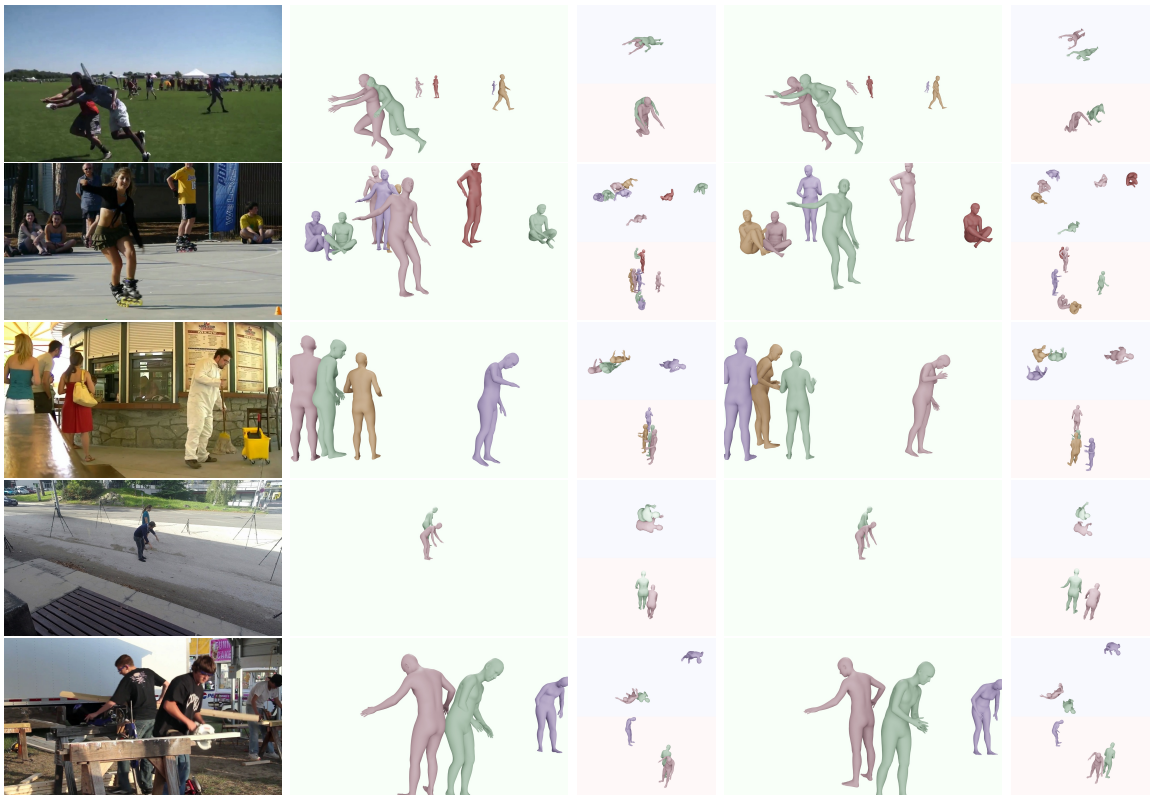

Figure 50: Qualitative effect of our proposed losses. Given an input image (first column), we provide results of the baseline model (second and third column) and our full model trained with our proposed losses (fourth and fifth column). As expected, we improve over our baseline in terms of coherency in the results (i.e., fewer interpenetrations, more consistent depth ordering for the reconstructed meshes). For the first image, the visualization focuses only on the two people in the foreground and the rest are ignored.

Figure 51: Successful reconstructions (1). We visualize the reconstructions of our approach from different viewpoints.

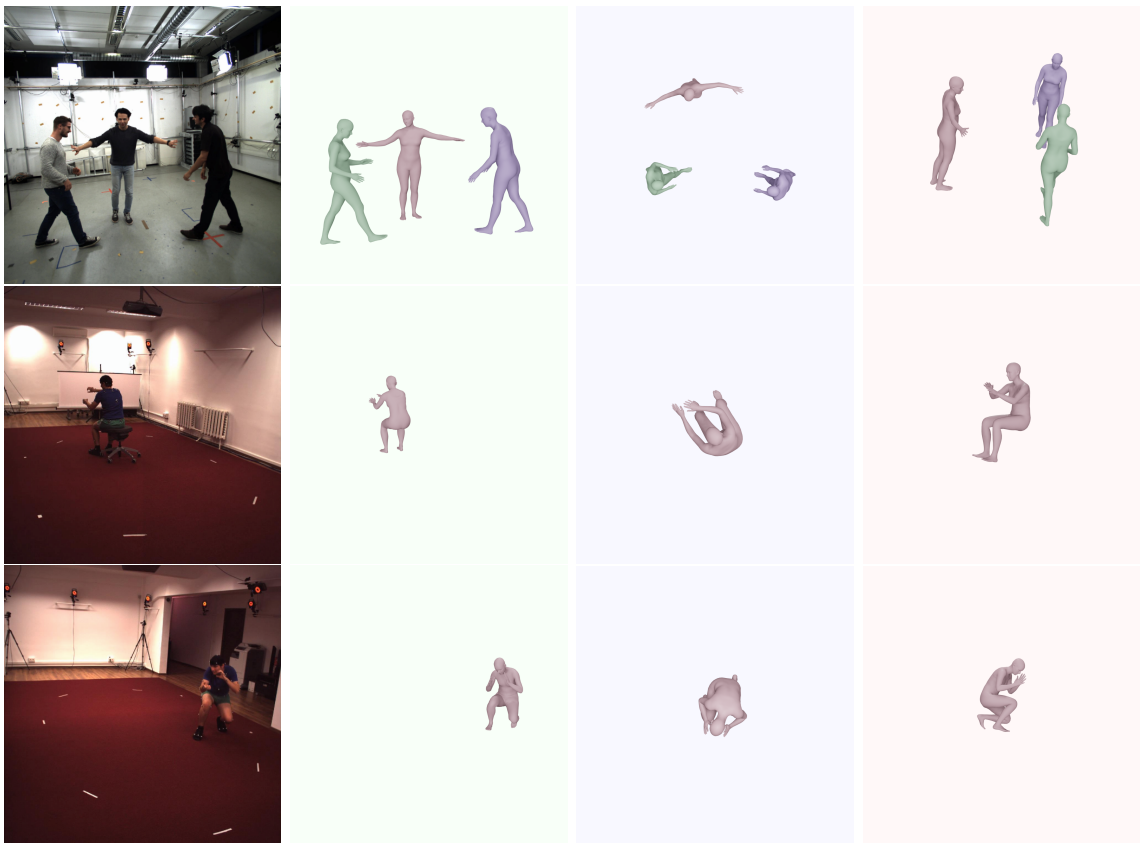

Figure 52: Successful reconstructions (2). We visualize the reconstructions of our approach from different viewpoints.

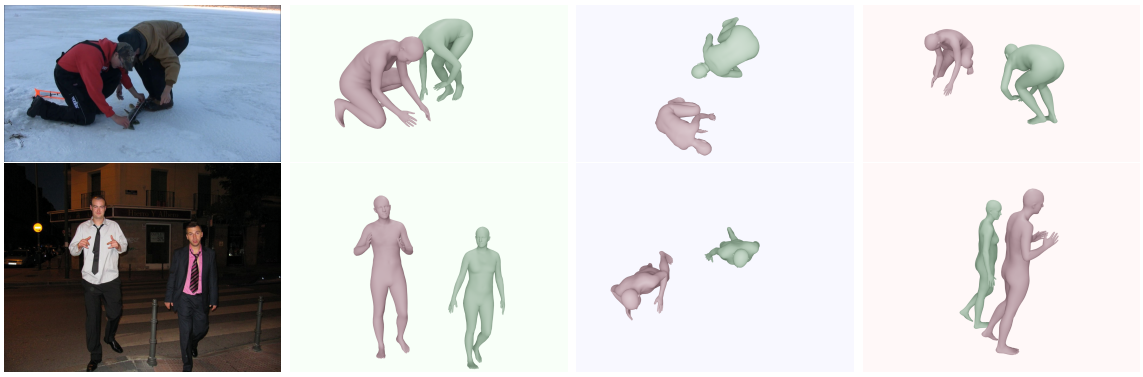

Figure 53: Failure cases. We visualize the reconstructions of our approach from different viewpoints. For the first image, the person on the right is slightly shorter than the person on the left, but this is hard to perceive by our model, that estimates roughly the same height for both people and positions the person on the right to be farther away from the camera. For the second image, our model estimates the depth ordering correctly, but clearly overestimates the distance between the two people, which are almost in contact.

BIBLIOGRAPHY

[1] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3D people models. In *CVPR*, 2018.

[2] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018.

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

[4] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: shape completion and animation of people. In *ACM transactions on graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[5] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3D human pose estimation in the wild. In *CVPR*, 2019.

[6] Benjamin Biggs, Sébastien Ehrhart, Hanbyul Joo, Benjamin Graham, Andrea Vedaldi, and David Novotny. 3D multibodies: Fitting sets of plausible 3D models to ambiguous image data. In *NeurIPS*, 2020.

[7] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016.

[8] Judith Butepage, Michael J. Black, Danica Kragic, and Hedvig Kjellstrom. Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *PAMI*, 43(1):172–186, 2019.

[10] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using part affinity fields. *PAMI*, 2019.

[11] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017.

[12] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2018.

[14] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *NIPS*, 2016.

[15] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018.

[16] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *ECCV*, 2020.

[17] Rishabh Dabral, Nitesh B Gundavarapu, Rahul Mitra, Abhishek Sharma, Ganesh Ramakrishnan, and Arjun Jain. Multi-person 3D human pose estimation from monocular images. In *3DV*, 2019.

[18] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. Learning 3D human pose from structure and motion. In *ECCV*, 2018.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[20] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In *ICLR*, 2015.

[21] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR*, 2017.

[22] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. nflows: normalizing flows in PyTorch, Nov. 2020.

[23] Jialue Fan, Wei Xu, Ying Wu, and Yihong Gong. Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks*, 21(10):1610–1623, 2010.

[24] Michela Farenzena, Loris Bazzani, Alessandro Perina, Vittorio Murino, and Marco Cristani. Person re-identification by symmetry-driven accumulation of local features. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2360–2367. IEEE, 2010.

[25] Valentin Gabeur, Jean-Sébastien Franco, Xavier Martin, Cordelia Schmid, and Gregory Rogez. Moulding humans: Non-parametric 3D human shape estimation from single images. In *ICCV*, 2019.

[26] Stuart Geman and Donald E McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 4:5–21, 1987.

[27] Georgios Georgakis, Ren Li, Srikrishna Karanam, Terrence Chen, Jana Košecká, and Ziyan Wu. Hierarchical kinematic human mesh recovery. In *ECCV*, 2020.

[28] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: masked autoencoder for distribution estimation. In *ICML*, 2015.

[29] Ross Girshick. Fast R-CNN. In *ICCV*, 2015.

[30] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[31] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *CVPR*, 2009.

[32] Rıza Alp Güler and Iasonas Kokkinos. HoloPose: Holistic 3D human reconstruction in-the-wild. In *CVPR*, 2019.

[33] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense human pose estimation in the wild. In *CVPR*, 2018.

[34] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *ICCV*, 2019.

[35] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019.

[36] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

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

[38] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V. Gehler, Javier Romero, Ijaz Akhter, and Michael J. Black. Towards accurate marker-less human shape and pose estimation over time. In *3DV*, 2017.

[39] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015.

[40] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *PAMI*, 36(7):1325–1339, 2013.

[41] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. In *ICCV*, 2017.

[42] Aaron S Jackson, Chris Manafas, and Georgios Tzimiropoulos. 3D human body reconstruction from a single image via volumetric regression. In *ECCVW*, 2018.

[43] Ehsan Jahangiri and Alan L Yuille. Generating multiple diverse hypotheses for human 3D pose consistent with 2D joint detections. In *ICCVW*, 2017.

[44] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013.

[45] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *CVPR*, 2020.

[46] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010.

[47] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, 2011.

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

[49] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3D human pose fitting towards in-the-wild 3D human pose estimation. *arXiv preprint arXiv:2004.03686*, 2020.

[50] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *PAMI*, 41(1):190–204, 2017.

[51] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018.

[52] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[53] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.

[54] Angjoo Kanazawa, Jason Zhang, Panna Felsen, and Jitendra Malik. Learning 3D human dynamics from video. In *CVPR*, 2019.

[55] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[56] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *CVPR*, 2018.

[57] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[58] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.

[59] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *NIPS*, 2016.

[60] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[61] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[62] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5253–5263, 2020.

[63] 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 *ICCV*, 2019.

[64] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019.

[65] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *ICCV*, 2021.

[66] Abhijit Kundu, Yin Li, and James M Rehg. 3D-RCNN: Instance-level 3D object reconstruction via render-and-compare. In *CVPR*, 2018.

[67] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *CVPR*, 2017.

[68] Hsi-Jian Lee and Zen Chen. Determination of 3D human body postures from a single view. *CVIU*, 30(2):148–168, 1985.

[69] Vincent Leroy, Philippe Weinzaepfel, Romain Brégier, Hadrien Combaluzier, and Grégory Rogez. SMPLy benchmarking 3D human pose estimation in the wild. In *3DV*, 2020.

[70] Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3D human pose estimation with mixture density network. In *CVPR*, 2019.

[71] Sijin Li and Antoni B Chan. 3D human pose estimation from monocular images with deep convolutional neural network. In *ACCV*, 2014.

[72] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T. Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019.

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

[74] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *CVPR*, 2018.

[75] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[76] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *ICCV*, 2019.

[77] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: Motion and shape capture from sparse markers. *ACM Transactions on Graphics (ToG)*, 33(6):1–13, 2014.

[78] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015.

[79] Diogo C Luvizon, David Picard, and Hedi Tabia. 2D/3D pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018.

[80] Siddharth Mahendran, Haider Ali, and Rene Vidal. A mixed classification-regression framework for 3D pose estimation from 2D images. In *BMVC*, 2018.

[81] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019.

[82] Julieta Martinez, Michael J. Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[83] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3D human pose estimation. In *ICCV*, 2017.

[84] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D human pose estimation in the wild using improved CNN supervision. In *3DV*, 2017.

[85] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. XNect: Real-time multi-person 3D human pose estimation with a single RGB camera. *arXiv preprint arXiv:1907.00837*, 2019.

[86] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3D pose estimation from monocular RGB. In *3DV*, 2018.

[87] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. VNect: Real-time 3D human pose estimation with a single RGB camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017.

[88] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3D multi-person pose estimation from a single RGB image. In *ICCV*, 2019.

[89] Arsalan Mousavian, Dragomir , John Flynn, and Jana Košecká. 3D bounding box estimation using deep learning and geometry. In *CVPR*, 2017.

[90] Lea Muller, Ahmed AA Osman, Siyu Tang, Chun-Hao P Huang, and Michael J Black. On self-contact and human pose. In *CVPR*, 2021.

[91] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[92] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *3DV*, 2018.

[93] Ahmed AA Osman, Timo Bolkart, and Michael J Black. STAR: Sparse trained articulated human body regressor. In *ECCV*, 2020.

[94] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *NIPS*, 2017.

[95] Despoina Paschalidou, Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3D reconstruction with ray potentials. In *CVPR*, 2018.

[96] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani,

Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

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

[98] Georgios Pavlakos, Nikos Kolotouros, and Kostas Daniilidis. TexturePose: Supervising human mesh estimation with texture consistency. In *ICCV*, 2019.

[99] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *CVPR*, 2018.

[100] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *CVPR*, 2017.

[101] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3D human pose and shape from a single color image. In *CVPR*, 2018.

[102] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

[103] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. DeepCut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016.

[104] Alin-Ionut Popa, Mihai Zanfir, and Cristian Sminchisescu. Deep multitask architecture for integrated 2D and 3D human sensing. In *CVPR*, 2017.

[105] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3D faces using convolutional mesh autoencoders. In *ECCV*, 2018.

[106] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[107] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.

[108] Helge Rhodin, Victor Constantin, Isinsu Katircioglu, Mathieu Salzmann, and Pascal Fua. Neural scene decomposition for multi-person motion capture. In *CVPR*, 2019.

[109] Gregory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *Advances in Neural Information Processing Systems*, 2016.

[110] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net: Localization-classification-regression for human pose. In *CVPR*, 2017.

[111] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net++: Multi-person 2D and 3D pose detection in natural images. *PAMI*, 2019.

[112] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6):245, 2017.

[113] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[114] Alexander G Schwing and Raquel Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015.

[115] Akash Sengupta, Ignas Budvytis, and Roberto Cipolla. Probabilistic 3D human shape and pose estimation from multiple unconstrained images in the wild. In *CVPR*, 2021.

[116] Saurabh Sharma, Pavan Teja Varigonda, Prashast Bindal, Abhishek Sharma, and Arjun Jain. Monocular 3D human pose estimation by generation and ordinal ranking. In *ICCV*, 2019.

[117] Leonid Sigal, Alexandru Balan, and Michael J Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *NIPS*, 2008.

[118] L. Sigal, A. Balan, and M. J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1):4–27, Mar. 2010.

[119] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[120] David Smith, Matthew Loper, Xiaochen Hu, Paris Mavroidis, and Javier Romero. FACSIMILE: Fast and accurate scans from an image in less than a second. In *ICCV*, 2019.

[121] Jie Song, Xu Chen, and Otmar Hilliges. Human body model fitting by learned gradient descent. In *ECCV*, 2020.

[122] David Stutz. *Learning shape completion from bounding boxes with CAD shape priors.* PhD thesis, Master's thesis, RWTH Aachen University, 2017.

[123] David Stutz and Andreas Geiger. Learning 3D shape completion from laser scan data with weak supervision. In *CVPR*, 2018.

[124] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[125] Xiao Sun, Bin Xiao, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018.

[126] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018.

[127] Yu Sun, Wu Liu, Qian Bao, Yili Fu, Tao Mei, and Michael J Black. Putting people in their place: Monocular regression of 3d people in depth. In *CVPR*, 2022.

[128] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *ECCV*, 2020.

[129] Vince Tan, Ignas Budvytis, and Roberto Cipolla. Indirect deep structured learning for 3D human body shape and pose prediction. In *BMVC*, 2017.

[130] Camillo J Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *CVIU*, 80(3):349–363, 2000.

[131] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+O: Unified egocentric recognition of 3D hand-object poses and interactions. In *CVPR*, 2019.

[132] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3D human pose with deep neural networks. In *BMVC*, 2016.

[133] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2D and 3D image cues for monocular body pose estimation. In *ICCV*, 2017.

[134] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3D pose estimation from a single image. In *CVPR*, 2017.

[135] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014.

[136] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[137] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *NIPS*, 2017.

[138] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. BodyNet: Volumetric inference of 3D human body shapes. In *ECCV*, 2018.

[139] Nitika Verma, Edmond Boyer, and Jakob Verbeek. FeaStNet: Feature-steered graph convolutions for 3D shape analysis. In *CVPR*, 2018.

[140] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using IMUs and a moving camera. In *ECCV*, 2018.

[141] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[142] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[143] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[144] Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

[145] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[146] Yuxin Wu and Kaiming He. Group normalization. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[147] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *CVPR*, 2019.

[148] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018.

[149] 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 *CVPR*, 2021.

[150] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7760–7770, 2019.

[151] Andrei Zanfir, Eduard Gabriel Bazavan, Hongyi Xu, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Weakly supervised 3D human pose and shape reconstruction with normalizing flows. In *ECCV*, 2020.

[152] Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Zanfir, William T. Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Neural descent for visual 3D human pose and shape. In *CVPR*, 2021.

[153] 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 *CVPR*, 2018.

[154] Andrei Zanfir, Elisabeta Marinoiu, Mihai Zanfir, Alin-Ionut Popa, and Cristian Sminchisescu. Deep network for the integrated 3D sensing of multiple people in natural images. In *NIPS*, 2018.

[155] Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3D people in scenes without people. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6194–6204, June 2020.

[156] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3586–3593, 2013.

[157] Ce Zheng, Wenhan Wu, Taojiannan Yang, Sijie Zhu, Chen Chen, Ruixu Liu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *arXiv preprint arXiv:2012.13392*, 2020.

[158] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015.

[159] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3D human pose estimation in the wild: a weakly-supervised approach. In *ICCV*, 2017.

[160] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3D human pose estimation from monocular video. In *CVPR*, 2016.

[161] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G. Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):901–914, 2019.

[162] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019.