# Keypoint-aligned 3D Human Shape Recovery from A Single Imagewith Bilayer-Graph

Yu, Xin; van Baar, Jeroen; Chen, Siheng; Sullivan, Alan

TR2021-143 January 08, 2022

## Abstract

The ability to estimate 3D human shape and pose from images can be useful in many contexts. Recent approaches have explored using graph convolutional networks, and achieved promising results. The fact that the 3D shape is represented by a mesh, an undirected graph, makes graph convolutional networks a natural fit for this problem. However, graph convolutional networks have limited representation power. Information from nodes in the graph is passed to connected neighbors, and propagation of information requires successive graph convolutions. To overcome this limitation, we propose a dual-scale graph approach. We use a coarse graph, derived from a dense graph, to estimate the human's 3D pose, and the dense graph to estimate the 3D shape. Information in coarse graphs can be propagated over longer distances compared to dense graphs. In addition, information about pose can guide to recover local shape detail, and vice versa. We recognize that the connection between coarse and dense is itself a graph, and introduce graph fusion blocks to exchange information between graphs with different scales. We train our model end-to-end and show that we can achieve state of the art results for several evaluation datasets.

*International Conference on 3D Vision (3DV) 2021*

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

# Joint 3D Human Shape Recovery and Pose Estimation from a Single Image with Bilayer Graph

Xin Yu*
School of Computing, University of Utah
Salt Lake City, Utah, USA
xiny@cs.utah.edu

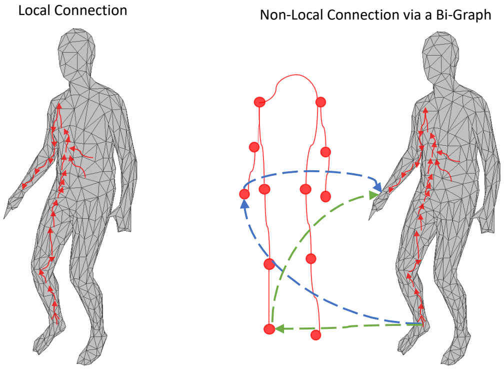
Jeroen van Baar†, Siheng Chen
Mitsubishi Electric Research Laboratories
Cambridge, MA, USA
{jeroen, schen}@merl.com

## Abstract

*The ability to estimate the 3D human shape and pose from images can be useful in many contexts. Recent approaches have explored using graph convolutional networks and achieved promising results. The fact that the 3D shape is represented by a mesh, an undirected graph, makes graph convolutional networks a natural fit for this problem. However, graph convolutional networks have limited representation power. Information from nodes in the graph is passed to connected neighbors, and propagation of information requires successive graph convolutions. To overcome this limitation, we propose a dual-scale graph approach. We use a coarse graph, derived from a dense graph, to estimate the human's 3D pose, and the dense graph to estimate the 3D shape. Information in coarse graphs can be propagated over longer distances compared to dense graphs. In addition, information about pose can guide to recover local shape detail and vice versa. We recognize that the connection between coarse and dense is itself a graph, and introduce graph fusion blocks to exchange information between graphs with different scales. We train our model end-to-end and show that we can achieve state-of-the-art results for several evaluation datasets. The code is available at the following link, https://github.com/yuxwind/BiGraphBody.*

## 1. Introduction

Recovering 3D human shapes and poses from 2D images is a fundamental task for numerous real-world applications, such as animation and dressing 3D people [4, 17]. Some recent approaches restrict themselves to only estimate 3D poses [45, 50, 52], while other approaches need multiple images to achieve reliable shape recovery [19, 25]. Here we consider joint 3D human shape recovery and pose estimation from a single image.



(a) Mesh-only graph (b) Bilayer graphs

Figure 1: By associating the mesh graph to the input image with a skeleton graph, the bilayer graph structure will shorten the paths between remote mesh nodes (1723 nodes here), when we connect a joint with the mesh nodes it controls. With the body parts are correlated, such as the ankle and the wrist, this bilayer graph implicitly learn the interaction between joints and mesh vertices and further shorten the path among the remote body mesh vertices.

As an undirected graph, a 3D mesh can represent a human shape, making graph-based techniques a natural fit to this task. For example, graph CMR [25] deforms a template human mesh in a neutral pose to a desired shape through a graph convolutional network. Graph convolutional layers then propagate the node features over the mesh. However, this mesh-graph only approach suffers from the issue node feature propagation will be extremely slow when the mesh has dense vertices, such as 1723 nodes used for human in general. We illustrate this limitation of mesh based graph in Fig. 1. The recent work [29] use transformer to reduce

[*] Work mainly done when Xin Yu was an intern at MERL.
[†] Corresponding author.

the distance between any two nodes to 1 via self-attention mechanisms. However, self-attention over a down-sampled 423 mesh nodes is still not efficient; positional encoding may maintain the base coordinates information in the sequential ordering of the mesh nodes, still ignores the structured correlations between body parts.

To resolve the above issues, we propose a bilayer graph structure, where one layer is a mesh graph for human shapes, and the other layer is a newly added skeleton graph for body joints. As shown in Fig. 1, the newly added skeleton graph can associate the 2D body joints estimated from an image with their coordinates in 3D space. [1] This body-joint-based correspondence allows us to attach detailed local image features to each body joint in the skeleton graph. In previous mesh-only graph approaches, as shown in Fig. 1a, the ankle and wrist nodes in the mesh graph can only connect to each other via multiple iterations of aggregation-and-combine operations in GCN. Image feature propagation will be extremely slow when the mesh template has 1723 nodes. This bilayer graph structure (see Fig. 1b) use sparse skeleton graph to guide the mesh nodes to exchange information in a more efficient way. It further shortens the paths between remote mesh nodes when connecting them via joints. We thus leverage the spatial nonlocality of the mesh graph.

An added benefit of this two-layer graph structure is multi-tasking: achieving shape recovery and pose estimation at the same time. Two layers naturally model a human body from mesh and skeleton scales, handling shape recovery and pose estimation, respectively. The cross or fusion layer is a trainable bipartite graph that connects body joints and mesh nodes. Instead of imposing any fixed connections, such a bipartite graph can adaptively adjust the relationships between the mesh nodes and body joints. It enables the feature fusion between two scales of a human body, mutually enhancing two tasks. This is related to linear blend skinning [22], however, where skinning provides an analytical transformation, our cross layer learns a data-adaptive transformation between body joints and mesh nodes in the high-dimensional feature space.

In summary, our main contributions are:

- We are the first to propose a neural network based on a two-layer graph structure that jointly achieves 3D human shape and pose recovery. The skeleton graph module propagates pose (coarser-scale) information, the mesh graph module propagates detailed shape (finer-scale) information, and the fusion graph module allows us to exchange information across the two modules.
- We propose an adaptive graph fusion block to learn the trainable correspondence between body joints and mesh nodes, promoting information exchange across two scales.
- We validate our method on several datasets (H36M, UP-3D, LSP), and show that exchanging local and global image information from different scales provides a significant improvement and speedup over single graph methods.

## 2. Related Work

**Human Shape Recovery** Over the years there have been many approaches to recover 3D human shapes from images. Several methods propose to recover clothed humans from either single or multi-view images [1, 10, 17, 35, 38, 40, 41, 49, 53]. These approaches rely on well segmented humans in the images, and do not emphasize accurate 3D shape and pose. Our goal instead is to capture the 3D shape and pose accurately without relying on any prior segmentation. Other methods rely on the video input to recover 3D human shapes [21, 36, 43, 51]. While Our goal is to recover accurate 3D shape and pose from a single image only. To handle the alignment issue between neutral and deformed poses, [5] proposes an optimization procedure to iteratively refine the estimate. The authors in [8, 33] introduce a sequential and iterative approach from 2D poses. In this work, we propose a trainable two-layer graph structure to resolve the alignment issue which does not require iterations.

**Graph CNNs for 3D Reconstruction** Recently, graph convolutional neural networks (GCNs) have been used to recover 3D objects from images. In this method, objects are represented as meshes [46, 47]. Meshes are the de facto representation of 3D objects in computer graphics, and a mesh can be considered an undirected (3D) graph. The initial mesh before refinement may be a mesh obtained from a volumetric estimation [11]. The authors in [9] state the limitations of GCN for 3D pose estimation. To overcome this limitation, they propose learnable weights for the structure of the graph. To address the limited representation capability of GCN, we propose a two-layer-graph neural networks with adaptive edge weights to share information between two graph layers. Our work is an extension to the regression based approach called Graph CMR [25]. The input to Graph CMR is a human mesh in neutral pose along with global image features. Graph CMR then relies solely on graph convolutions to propagate information between nodes, and finally provide a 3D estimate. The mesh is refined by estimating SMPL parameters. We propose a two-layer graph to more efficiently propagate information. Multi-scale graphs have been explored in some other applications. The authors in [12] use multiple graph scales and exchange information via connectivity between different scales for the purpose of human parsing. The authors in [28] use multi-scale graphs for the purpose of joint human action recognition and motion prediction. The information between scales is fused according to feature replication and concatenation between the different graphs. In comparison, our fusion approach relies on a learnable graph adjacency matrix, which exchanges information between two body scales.

[1] 2D body joints can be well estimated from an image, e.g., [7, 42].

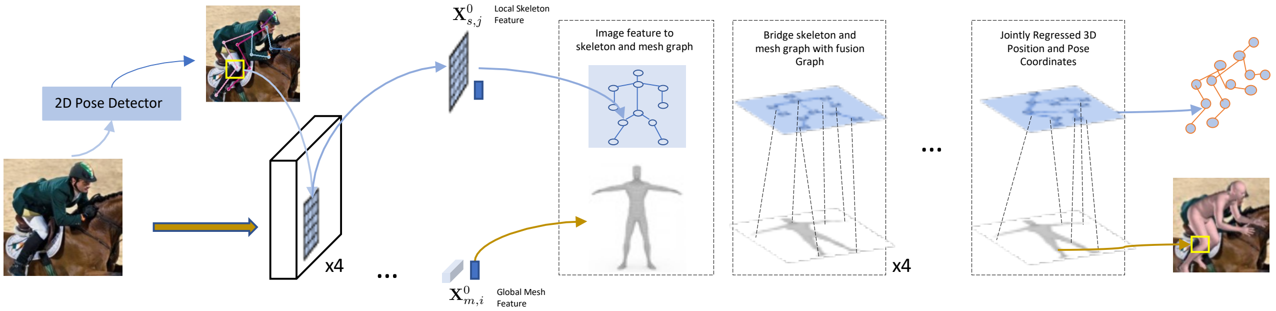

2D Pose Detector
x4
...
$\mathbf{X}^0_{s,j}$
Local Skeleton Feature
$\mathbf{X}^0_{m,i}$
Global Mesh Feature
Image feature to skeleton and mesh graph
Bridge skeleton and mesh graph with fusion Graph
x4
...
Jointly Regressed 3D Position and Pose Coordinates

Figure 2: Our proposed bilayer graph architecture. Given an input image, the mesh graph module (Mesh-GCN) is a regression which outputs 3D vertex coordinates, and the skeleton graph module (Skeleton-GCN) estimates a skeleton with twelve 3D joint locations. The input to the Mesh-GCN is a template mesh together with a global perceptual feature extracted using a CNN, from the bounding box around the person in the image. Each global perceptual feature is attached to the XYZ coordinates of the vertices in the mesh. For clarity, we omit the SMPL part. The input to each joint node in the Skeleton-GCN is a local perceptual feature extracted from the image regions around the 2D joints estimated by HRNet. The two modules exchange information via so-called fusion graph, which is a bipartite graph between all mesh nodes and joints.

## 3. Problem Formulation

As shown in Fig. 1, to resolve the issues of lacking detailed local information and inefficient long-range interactions, we use a bilayer graph structure to jointly estimate a 3D human pose and recover a complete 3D mesh based on a single input RGB image (without knowing camera parameters). Mathematically, let $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ be an RGB image with the height $H$ and the width $W$. Both 3D human pose and 3D mesh structure can be represented as a graph with a set of node coordinates and an adjacency matrix indicating their connecting relations. For the 3D pose, we denote the skeleton pose coordinates as $\mathbf{V_s} \in \mathbb{R}^{N_\mathbf{s}\times 3}$, where $N_s$ is the total number of body joints, thus the corresponding skeleton adjacency matrix is $\mathbf{A_s} \in \mathbb{R}^{N_\mathbf{s}\times N_\mathbf{s}}$. For the 3D mesh structure, we denote the mesh node coordinates as $\mathbf{V_m} \in \mathbb{R}^{N_\mathbf{m}\times 3}$, where $N_m$ is the number of mesh nodes. Then the adjacency matrix for the mesh structure is $\mathbf{A_m} \in \mathbb{R}^{N_\mathbf{m}\times N_\mathbf{m}}$. We aim to propose a model $\mathcal{F}(\cdot)$:

$$\widehat{\mathbf{V}}_\mathbf{s}, \widehat{\mathbf{V}}_\mathbf{m} = \mathcal{F}(\mathbf{I}, \mathbf{A}_m, \mathbf{A}_s), \tag{1}$$

to estimate human pose $\widehat{\mathbf{V}}_\mathbf{s}$ and the recovered human mesh $\widehat{\mathbf{V}}_\mathbf{m}$, which precisely approximate the targets $\mathbf{V_s}, \mathbf{V_m}$, respectively.

This joint task naturally requires to model a human body at two scales: a sparse graph at the skeleton scale and a dense graph at the mesh scale. To explicitly model the vertex correlations at two scales, we introduce a fusion graph to learn how the joints control the deformation of the body mesh vertices, and vice versa. Thus we propose a *two-layer graph structure* that consists of a skeleton graph $G_s(\mathcal{V}_s, \mathbf{A}_s)$, a mesh graph $G_m(\mathcal{V}_m, \mathbf{A}_m)$ and a fusion graph $G_f(\mathcal{V}_s \cup V_m, \mathbf{A}_f)$, where $\mathbf{A}_f \in \mathcal{R}^{(N_m+N_s)\times(N_m+N_s)}$ is the adjacency matrix for the fusion graph. Note that $\mathbf{A}_s, \mathbf{A}_m$ are fixed and given based on the human body prior; see the predefined graph topology in Fig. 1; while $\mathbf{A}_f$ is data-adaptive during training.

The graph-based formulation makes Graph CNN (GCN) a natural fit for this task. We propose a Bilayer-Graph GCN to address this task effectively and efficiently. As a core of the proposed system, it brings two benefits: First, it naturally models a human body from both mesh and skeleton aspects, promoting local and non-local topology learning, which will speedup the convergence of training and improve the joint pose and shape recovery. Second, a fusion graph enables information exchange between two scales of a human body, mutually enhancing feature extraction at two scales and further improving the performances in two tasks.

## 4. Two-scale Graph Neural Network

To model the two-scale skeleton and mesh graph, and a fusion graph connecting them, we propose a bilayer graph neural network. Fig. 2 shows an overview of the this architecture. In this section, we introduce the detailed implementation of each building block in our proposed method.

### 4.1. Architecture Overview

As shown in Fig. 2, given a single input image, an image encoder will be firstly used to extract the features from it, and a 2D-pose detector will processes the image into a skeleton graph. Then on the top part, skeleton graph module attaches local joint features, and propagate the features in skeleton graph layer. While at the bottom part, mesh graph module attach the global image features and models the mesh graph layer. Between them, fusion graph module connects between all the skeleton joints and mesh nodes, and exchange dual-scale information in a structured way. Finally, the learned joint and mesh node representation will be used to regress the 3D pose and mesh coordinates.

## 4.2. Image Encoder

The functionality of an image encoder module is to extract informative visual features from an RGB image, which would be the input for the subsequent modules. Given an input RGB image $\mathbf{I}$, we use a multi-layer CNN to obtain a collection of intermediate image features from the output of each layer $l$, $\{\mathbf{X}_{\text{im}}^{(\ell)}\}_{\ell=1}^{L} = \mathcal{F}_{\text{im}}(\mathbf{I})$, where $\mathcal{F}_{\text{im}}(\cdot)$ is a CNN whose architecture follows ResNet50 [14]. [2]

## 4.3. Bilayer Graph Module

We propose to employ Graph CNN to jointly regress the 3D coordinates of the mesh and skeleton vertices. It consists of three sub-graphs: the mesh graph module, the skeleton graph and the fusion graph module.

For each sub-graph, we employ the same basic graph convolutions to formulate them [23], which is defined as:

$$\mathbf{X}^{\ell+1} = \mathbf{A}\mathbf{X}^{\ell}\mathbf{W} \in \mathbb{R}^{N\times d_{\ell+1}} \tag{2}$$

where $\ell$ indicates the $\ell$-th convolutional layer, $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a graph adjacency matrix for the (sub)graph, $\mathbf{W} \in \mathbb{R}^{d_\ell \times d_{\ell+1}}$ is a convolution weight matrix, $\mathbf{X}^{\ell} \in \mathbb{R}^{N\times d_\ell}$ is the input feature vector. As shown in Eq. (2), given a set of vertices initialized with **input features** ($X^0$) and their **adjacency matrices** ($\mathbf{A}$), the graph convolution layer allows feature propagating and updating over the (sub)graph so that each vertex can aggregate information from its neighbors. In the following, we will introduce the functions of the three sub-graphs separately, and comparatively study the input and adjacency matrix of them.

### 4.3.1 Mesh Graph Module

The functionality of a mesh graph module is to regress the posed 3D mesh conditioned on the input image features. It is identical to graph layers of GraphCMR [25]. We start from the template mesh in neutral (T-) pose introduced by SMPL and deform them to the shaped and posed mesh with the graph convolutions.

**Inputs** We employ the template coordinates as the position embedding of the mesh vertices, and attach it with the 2048-D global feature vector of ResNet-50 [14] to feed in the mesh graph module.

Let $\mathbf{x}^{\text{g}}_{\text{im}} \in \mathbb{R}^{D_g}$ be the global image feature after the average pooling layer and $\mathbf{v}^T_{m,i} \in \mathbb{R}^3$ is the 3D coordinate of a $i$-th template mesh vertex. For each mesh vertex, we have an initialized feature defined as:

$$\mathbf{X}^0_{m,i} = \mathcal{F}^0_m(\mathbf{v}^T_{m,i} \oplus \mathbf{x}^{\text{g}}_{\text{im}}) \in \mathbb{R}^{d_0} \tag{3}$$

Where $\oplus$ denotes feature vector concatenation and $\mathcal{F}^0_m$ denotes the linear layer to reduce the dimension (the typical reduced dimension is 512) of the concatenated features whose weights shared among all mesh vertices.

**Mesh adjacency matrix** $A_m$ is initialed as a binary matrix to indicate the connectivity among the vertices as shown in Fig. 1 and further row-normalized.

### 4.3.2 Skeleton Graph Module

The functionality of a skeleton graph module is to lift the 2D pose estimated from an input RBG image to a 3D pose. As shown in Fig. 1b, the sparse skeleton graph can promote the non-local topology features of the dense mesh graph and enhance the correlations between different body parts. Furthermore, we extracted the local features around the joints for precise pose.

**Inputs** Instead of using global image features for all the joint nodes, we use joint-aware local features for joint nodes. Given the image, we use HRNet [42] off-the-shelve to estimate the 2D positions of body joints in this image. For each body joint, we crop a patch centered at the estimated 2D positions with the size of the average estimated bone length from the joints per image, using RoI Align [13] from the $k$-th image feature map ($K$ layers in total) $\mathbf{x}^k_{\text{im}}$ of ResNet-50 [14]. This feature patch reflects the local visual information around the corresponding body joint. We concatenate the image feature patches with the positional embedding as the initial skeleton features as:

$$\mathbf{X}^0_{s,i} = \mathcal{F}^0_s(\hat{\mathbf{v}}_{s,i} \oplus \mathbf{RoI}(\hat{\mathbf{v}}_{s,i}, \mathbf{x}^1_{\text{im}}, \ldots \mathbf{x}^K_{\text{im}})) \in \mathbb{R}^{d_0} \tag{4}$$

where $i$-th body joint estimated by HRNet as $\widehat{\mathbf{s}}_i \in \mathbb{R}^2$, $\mathbf{RoI}(\cdot)$ returns image feature patches from $\mathbf{x}^k_{\text{im}}$ using RoI Align [13] with the patch centered at $\widehat{\mathbf{s}}_i$. Similarly, $\mathcal{F}^0_s$ is a linear layer and share the weights among the skeleton vertices. We also experimented with the skeleton template coordinates as the positional embedding but we did not observe quantitative improvement in the results and thus keep the 2D embedding for all experiments.

**Skeleton adjacency matrix** We use fixed adjacency matrix for $\mathbf{A}_s$. The element is initialized as the reciprocal of the Euclidean distance between two template joint vertices.

### 4.3.3 Fusion Graph Module

The functionality of a fusion graph is to correlate the sparse skeleton graph and the dense mesh graph and enable information exchange between them and mutually enhance both tasks of 3D shape recovery and pose estimation. As shown in Fig. 1b, the fusion graph connection can shorten the path of two remote mesh vertices dramatically and will speedup the non-local information propagation of the mesh graph.

**Inputs** The fusion graph consists of the vertex from both the Mesh Graph and Skeleton Graph and applies the same

[2] Any "typical" CNN auto-encoder can be used for the image feature extraction.

initial feature as those two sub-graphs. The intermediate input of this module are the intermediate features from the Skeleton Graph and Mesh Graph, which fuses those intermediate features and populates them back to both Skeleton-GCN and Mesh-GCN.

**Fusion adjacency matrix** To fuse features from the skeleton and mesh graph, we leverage a trainable fusion graph to reflect the data-driven connectivity between body joints and mesh nodes. Besides defining a fixed connection part, denoted as $A_{f,s}$, we allow an extra dynamic connection, denoted as $W_f$, to be trainable to capture the connectivity in the hidden feature space in a data-driven manner. We define the final adjacency matrix as:

$$A_f = \mathrm{RowNorm}(A_{f,s} \odot W_f) \tag{5}$$

where $A_{f,s}, W_f \in \mathcal{R}^{(N_m+N_s)\times(N_m+N_s)}$, $\mathrm{RowNorm}()$ indicate a row normalization, $\odot$ denotes element wise product. $W_f$ is learnable and its element is initialized as 1 for vertex-joint correlation and 0 for joint-joint and vertex-vertex (the Skeleton and Mesh Graphs have cover those connections). The element of $A_{f,s}$ for a connection between a skeleton vertex and mesh vertex is fixed to the reciprocal of their Euclidean distance; otherwise, it is zero.

We also experiment the $\mathrm{RowNorm}()$ with a softmax on each row and apply additive optation between $A_{f_s}$ and $W_f$, both of which bring minor change to the performance. We adopt the row normalization and element-wise product for their simplification of computation.

### 4.3.4 Architecture Implementation

**Bilayer-Graph Block** At the heart of this approach, we propose a Bilayer-Graph block as an elementary computational unit for feature learning and propagation based on the bilayer-graph structure. As illustrated in Fig. 3, the Skeleton-GCN Block, Fusion-GCN Block and Mesh-GCN Block apply graph convolutions on skeleton graph, fusion graph and mesh graph respectively. The Fusion-GCN Block collects features from both Mesh-GCN and Skeleton-GCN and distributes the updated features back to the other module. Each block consists of a sequence of a graph linear layer, a graph convolution layer, and another graph linear layer, with a residual connection from the input directly to the output of this block. Each layer follows a group normalization layer [48] and ReLU. Please note that the graph linear layer is a special graph convolution layer, which simply substitutes the graph adjacency matrix $\mathbf{A}$ in the graph convolution layer ( see Eq. (2)) to an identity matrix.

In this network, we stack five Bilayer-Graph blocks for feature propagation, followed by two graph linear layers to regress the skeleton vertices and joint vertices separately. The first linear layers also follows a group normalization [48] and ReLU.

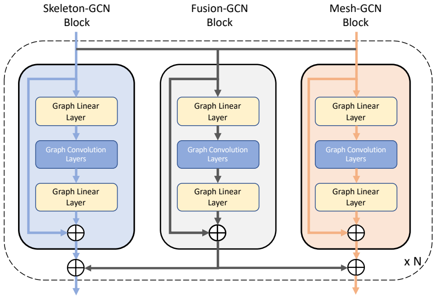


Figure 3: A Bilayer-Graph Block consists of a Fusion-GCN block, a Mesh-GCN block and a Fusion-GCN block. The fusion block depicted here has its input from the previous Skeleton-GCN and Mesh-GCN blocks prior to this Bilayer-Graph block, and adds its output back to each branch after their respective blocks.

**SMPL regressor** As the parametric representation of the human body can be very useful for down-stream tasks (e.g., body manipulation), we follow [25] to train a MLP module to regress pose ($\widehat{\theta}$) and shape ($\widehat{\beta}$) parameters for a SMPL model [32] from the predicted mesh $\widehat{\mathbf{V}}_m$.

## 4.4. Training

**Losses** To train the Bilayer GCN, we apply loss functions on the output of the Bilayer GCN and SMPL regressor and minimize the errors between the predictions and ground truths. Firstly, we use the a per-vertex $L_1$ loss between the ground truth $\mathbf{V}_m$ and predicted mesh vertices $\hat{\mathbf{V}}_m$ from Mesh-GCN, denoted as $\mathcal{L}_m$, and between the GT and predicted joint vertices $\hat{\mathbf{V}}_s$ from Skeleton-GCN, denoted as $\mathcal{L}_s$.

We follow [20, 25] to multiply the predicted mesh $\hat{\mathbf{V}}_m$ by a predefined matrix to get 3D joints, denoted as $\hat{\mathbf{V}}_m^{j3d}$. $L_1$ loss is also applied to it and its GT $\mathbf{V}_s$, denoted as $\mathcal{L}_m^{j3d}$.

As we trained on mixed datasets consisting of both 3D and 2D data, we have additional supervision on the predict a weak perspective camera parameters from the intermediate features of the Mesh-GCN with two graph linear layers. Apply this camera parameters to $\hat{\mathbf{V}}_m^{j3d}$ and $\hat{\mathbf{V}}_s$, we get two sets of 2D pose and use a $\mathcal{L}_1$ loss on them and the 2D GT pose, denoted as $\mathcal{J}_m^{j2d}$ and $\mathcal{J}_s^{j2d}$ respectively.

Finally, we apply MSE loss on the predicted SMPL shape ($\hat{\theta}$) and pose ($\hat{\beta}$) parameters, denoted as $\mathcal{L}_\theta$ and $\mathcal{L}_\beta$ respectively. And we have the final loss as below:

$$\mathcal{L} = \mathcal{L}_{\mathrm{m}} + \mathcal{L}_m^{j3d} + \mathcal{L}_m^{j2d} + \mathcal{L}_s + \mathcal{L}_s^{j2d} + \mathcal{L}_\theta + \lambda\mathcal{L}_\beta, \tag{6}$$

**Focal loss for regression** We observe that in the above losses on 3D vertices, the error caused by each body part varies a lot. For example, the joints on legs and arms usually have much larger error than the other parts. The intuition is that the variation for body limbs is much larger compared to torso and head. We generalizes the focal loss [30], which addresses class imbalance by down-weighting the loss for well-classified samples, to this regression tasks to addresses the imbalanced vertex error. We modify it based on the $L_1$ loss of the target, i.e.,

$$\mathcal{L}_{fl} = -(\alpha\mathcal{L})^{\gamma} \log(1 - \max(\tau, \alpha\mathcal{L})),$$

where $\mathcal{L}$ is the $L1$ loss, $\alpha$ is a factor to scale $\mathcal{L}$ to (0,1), $\tau < 1$ is a threshold that truncates $\alpha\mathcal{L}$ with a maximum value to avoid unreasonably large loss when $\alpha\mathcal{L}$ approaches 1, $(\alpha\mathcal{L})^{\gamma}$ is a factor to reduce the relative loss for well-regressed vertices with $\gamma > 0$.

# 5. Empirical Evaluations

We have evaluated our proposed method and present the results in this section. The datasets have different 2D annotations. We have selected the 12 joint annotations in common for the skeleton graph to define its graph structure.

## 5.1. Datasets and Evaluation metrics

**Human 3.6M** This indoor 3D dataset [15, 16] comprises eleven subjects performing 17 common scenarios, e.g. sitting down, talking on the phone. The training data contains ground truth 2D joints, 3D joints, and SMPL (pose and shape) parameters. The entire dataset contains 3.6M images. For training we only have access to subjects 1, 5, 6, 7, and 8 (about 1.55M images). Subjects 9 and 11 are held out for evaluation (about 0.5M images).

**UP-3D** Unite the People 3D [27] consists of images with annotations by humans doing sports and other miscellaneous activities. Besides ground truth 2d keypoints, SMPL fits have been performed on the 2D keypoints to produce ground truth SMPL parameters. About 7K images are used for training, and 639 held out for evaluation.

**LSP** Leeds Sports Pose [18] contains 2K images with 2D joint annotations of people playing sports. We use 1000 images for training, and 1000 for evaluation.

**COCO** Common Object in Context [31] also contains images of people annotated 2D keypoints. About 28K images are used for training. We do not evaluate for this dataset.

**MPII** MPII Human Pose dataset contains images with annotated body joints of people performing 410 different activities [2]. We use about 15K training images from this dataset, and do not evaluate on this dataset.

**Evaluation metrics** For H36M we report the mean Euclidean distance (**mm**) between the predicted and ground truth 3D joints after root joint alignment (**MPJPE**), and rigid alignment error (**PA-MPJPE**) as in [54]. For UP-3D we report **MPVE**, which is a mean per-vertex error between the predicted and ground truth shape, and for LSP we report accuracy (**Acc.**) and **F1** score on foreground-background (**FB seg**) and part segmentation (**Parts seg**). We report non-parametric (**np**) and SMPL parametric (**p**) predictions for H36M P1, P2 and UP-3D datasets.

| Methods | H36M P1 | | H36M P2 | |
|---|---|---|---|---|
| | MPJPE↓ | PA-MPJPE↓ | MPJPE↓ | PA-MPJPE↓ |
| SMPLify [5] | - | - | - | 82.3 |
| Lassner [27] | - | - | - | 93.9 |
| HMR [19] | 88.5 | 58.1 | - | 56.8 |
| NBF [34] | - | - | - | 59.5 |
| Pavlakos [37] | - | - | - | 75.9 |
| Kanazawa [21] | - | - | - | 56.9 |
| Arnab [3] | - | - | 77.8 | 54.3 |
| GraphCMR [25] | 75.0 | 51.2 | 72.7 | 49.3 |
| SPIN [24] | - | - | - | 41.1 |
| I2L-MeshNet [33] | - | - | 55.7 | 41.1 |
| METRO [29] | - | - | **54.0** | 36.7 |
| Ours | **61.2** | **35.4** | 58.5 | **34.0** |

Table 1: Comparison with the state-of-the-art on Human3.6M (Protocal 1 and 2) for estimated 3D poses (see suppl. mat. for per-activities results).

| Methods | FB seg | | Parts seg | |
|---|---|---|---|---|
| | Acc.↑ | F1↑ | Acc.↑ | F1↑ |
| SMPLify oracle [6] | 92.17 | 0.88 | 88.82 | 0.67 |
| SMPLify [6] | 91.89 | 0.88 | 87.71 | 0.64 |
| SMPLify on [37] | 92.17 | 0.88 | 88.24 | 0.64 |
| Bodynet [44] | 92.75 | 0.84 | - | - |
| HMR [19] | 91.67 | 0.87 | 87.12 | 0.60 |
| SPIN [24] | 91.83 | 0.87 | 89.41 | 0.68 |
| GraphCMR [25] | 91.46 | 0.87 | 88.69 | 0.66 |
| Ours | **93.15** | **0.89** | **90.96** | **0.73** |

Table 2: Comparison with the state-of-the-art on LSP for 2D projection from the predicted non-parametric mesh.

| Methods | MPVE (np) ↓ | MPVE(p) ↓ |
|---|---|---|
| GraphCMR [25] | 104.5 | 122.9 |
| Ours | **59.0** | **61.1** |

Table 3: Comparison with the mesh-only graph method [25] on UP-3D for estimated 3D mes (MPVE is in mm).

## 5.2. Experiment Details

We use a pre-trained ResNet-50 to extract perceptual features. Our model is trained end-to-end with a batch size of 64 and learning rate of $2.5e^{-4}$. Mini-batches during training are assembled by selecting images from the five training datasets. The composition is 30%, 20%, 10%, 20% and 20% for Human3.6M, UP-3D, LSP, COCO and MPII respectively. The Adam optimizer is used to determine the weight updates. We train our model for fifty epochs, but we observed fewer epochs could suffice (see Section 5.3). During training, we apply the focal loss $\mathcal{L}_{fl}$ (with $\alpha = 1$, $\gamma = 1$) to the estimated 3D pose from Mesh-GCN ($\hat{\mathbf{V}}_s$) and the coefficient before this loss term is 5.0. We mixed the ground truth and the estimated 2D joint location to get feature patches during training with a mixture ratio which gradually decreases to zero at the last epoch during training and only use the estimated 2D joints during inference.

## 5.3. Main Results and Analysis

We compare our method to state-of-the-art methods on H36M, UP-3D and LSP datasets, which evaluates 3D poses, 3D mesh, and 2D projections of the mesh respectively.

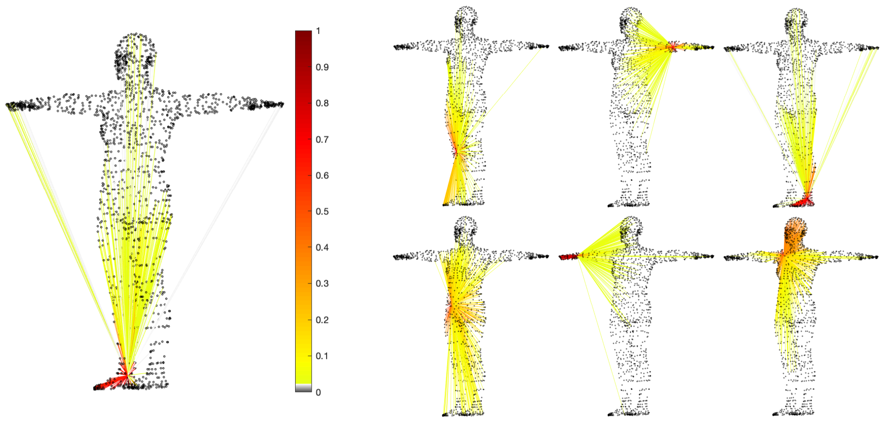


Figure 4: A visualization of the learned fusion adjacency matrix ($A_f$). **Left**: the connections between the right ankle and the mesh vertices. **Top row on the right**: right knee, left elbow, left ankle; **Bottom row on the right**: right hip, right wrist, right shoulder. Red, yellow and gray color indicate strong, weak and trivial connections.

The results are shown in Tables 1, 3 and 2 for those three datasets. Our method either outperforms or achieves comparable performance as the prior methods on those datasets.

First of all, we aim to investigate how the bi-layer graph performs for body recovery. To this end, we first focus on the Human3.6M dataset and UP-3D dataset. The rich human activities in their images is a natural target to study the correlation between body parts, which requires long-range interactions. We evaluate the regressed mesh by our bi-layer graph through 3D pose accuracy, in comparison to the mesh-only graph method [26] and the self-attention in transformer [29] as shown in Table 1. In both cases, we outperform them in reconstruction error (PA-MPJPE), indicating that our proposed bi-layer graph uses the non-local interactions efficiently for body recovery. We also evaluate the regressed mesh and the mesh calculated from the regressed SMPL on the UP-3D dataset in Table 3, which demonstrates that our method can promote the fine-grained interactions between mesh vertices for improved body shape. We also evaluate 3D shape through silhouette projection on the LSP dataset in Table 2. Our proposed bi-layered graph again outperforms prior methods.

Our model aim to jointly model local vertex-vertex (defined by mesh neighbourhood), non-local vertex-joint, and joint-joint interactions. We get insight of vertex-joint intersections by the learned fusion adjacency matrix ($A_f$) in Fig. 4. Firstly, strong interactions between a joint and its nearby mesh vertices are encouraged, thus the joint will guide the mesh recovery. We achieve this by setting the initial values of $A_f, s$ (see Eq. (5)) as the reciprocal of the Euclidean vertex-join distance in T-pose mesh. This intersections cover a range larger than the fine-grained vertex-vertex interactions predefined by mesh neighborhood. Secondly, long-range joint-vertex interactions are learnt between a joint and remote vertices near another joint, when the body part correlation happens. Please see the example of the right ankle and right wrist in the left sub-figure of Fig. 4. Our intersections differ from the transformer-based METRO [29] in two ways: the local vertex-vertex intersections avoid huge computation of the brute-force self-attention; and joint-vertex intersections learnt from the Fusion Graph efficiently model the most important topology knowledge between body mesh and joints. Together with the localized image features, our model achieves comparable performance to METRO [29] of the strong representation ability for the fully connected intersections. We believe that attention and knowledge-aware bi-layer graph network can be integrated to learn the interactions.

Finally, we evaluate the efficiency of the bi-layer graph structure through the convergence speed of the training compared to the mesh-only method [25] (see Fig. 5). Our model (in blue) achieves a lower, more stable loss much earlier compared to the baseline [25] (in orange), which indicates that the speedup of information propagation along this bi-layer graph can potentially reduce the training time.

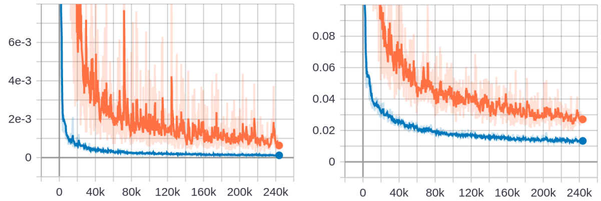


Figure 5: Comparison of training loss on $\hat{\mathcal{V}}_m^{j3d}$(left) and $\hat{\mathcal{V}}_m$(right) from the common mesh graph structure of the baseline [25] and our proposed model. We trained for 50 epochs and one epoch takes about 5,000 steps.

**Qualitative Results** Fig. 6 and 7 show four successful examples and two failure cases due to challenging poses and occlusions (see suppl. mat. for more examples).

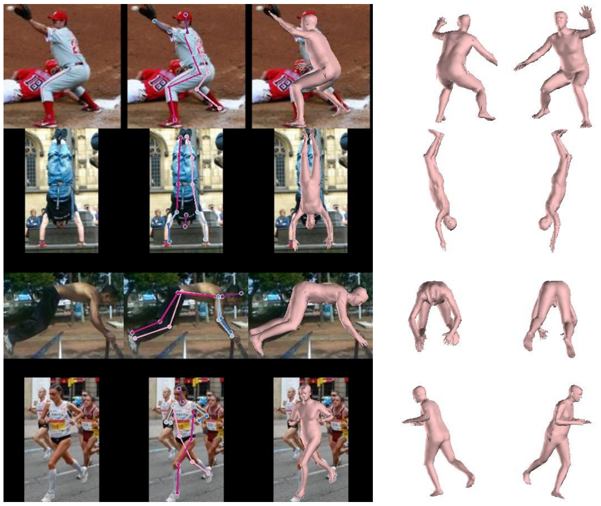


Figure 6: Qualitative results (left to right: input image, non-parametric pose and shape, two other view perspectives).

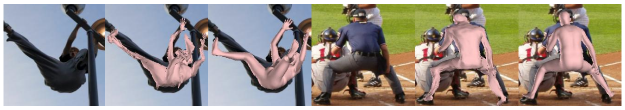


Figure 7: Two failure case examples (input, non-parametric, parametric).

### 5.4. Ablation Studies

**Benefit of bi-layer graph vs. localized features** As shown in Table 4, instead of single-mesh only graph (**A**), bi-layer graph only (**B**) or localized image features only (**C**) , it is the combination (**D**) of Fusion Graph and localized features that jointly contribute most to the performance gain. This combination is the core difference from the transformer-based METRO [29] and other hierarchical structures, such as CoMA [39] (see suppl. mat. for more discussion).

**Benefit of Fusion Graph** Since we design a bi-layer graph structure connected with fusion graph for mesh recovery, one interesting question is that whether the fusion graph is useful. In Table 5, We study the fusion graph by limiting its usage in the bi-layer graph network: replacing with a simple fusion by pooling (**avgpool-as-fusion** and **maxpool-as-fusion**) and restricting it applied to only the first or the last graph layer(**fusion-at-first** and **fusion-at-last** respectively). The simple fusion strategy will loss the individual interaction between a joint vertex and a mesh vertex as each joint (mesh) vertex apply an identical feature from pooling the mesh (joint) vertices feature. We compare those strategies to our fusion graph and observe the performance increase significantly with allowing more fusion connections in the network and our fusion graph works on best with the full connections in all graph layers.

|  | Skeleton Graph | Fusion Graph | Localized features | H36M P2 | UP-3D |
|---|---|---|---|---|---|
|  |  |  |  | MPJPE ↓ | MPVE ↓ |
| A | ✗ | ✗ | ✗ | 54.0 | 104.5 |
| B | ✓ | ✓ | ✗ | 47.5 | 96.3 |
| C | ✓ | ✗ | ✓ | 48.8 | 100.7 |
| D | ✓ | ✓ | ✓ | **34.0** | **59.0** |

Table 4: Evaluation of bi-layer graph components and localized features. All has Mesh Graph with global image features as input as GraphCMR and has the same training settings.

| Methods | H36M P2 | UP-3D |
|---|---|---|
|  | MPJPE ↓ | MPVE ↓ |
| Ours | **34.0** | **59.0** |
| avgpool-as-fusion | 47.3 | 81.1 |
| maxpool-as-fusion | 42.9 | 77.0 |
| fusion-at-first | 36.7 | 64.3 |
| fusion-at-last | 38.3 | 71.3 |
| shared weight | 34.2 | 61.7 |
| no FL | 34.7 | 59.8 |

Table 5: Evaluation results for ablation studies. See text for details. FL is for focal loss, shared weight indicates the model shares weights between skeleton and mesh graph.

**Weight sharing** We exploit the property of GCN to share weights between skeleton-GCN and Mesh-GCN for compact model size. In Table 5, it is interesting to observe only marginal performance loss, indicating the strong representation ability of GCN on the body and skeleton topology.

**Focal loss** To demonstrate the benefit of the focal loss for regression, we trained the model with $L_1$ loss on $\widehat{\mathbf{V}}_s$ instead of the proposed focal loss and keep the other losses the same. In Table 5, we see that $L_1$ loss works a bit inferior to the focal loss. We will explore the use of focal loss in future work for improving the overall performance.

## 6. Conclusion

We have proposed a dual-scale graph-based method for 3D human shape and pose recovery from a single image. A skeleton graph estimates 3D pose, and a mesh graph estimates 3D shape. A fusion graph promotes the exchange of local and global information between the two graphs. And Fusion Graph employs an adaptive adjacency matrix to learn which nodes between the two scales influence one another most. Our results show that we can outperform state-of-the-art methods. Some poses, and partial occlusions remain challenging. For future work, We would like to extend our work to take both single and multi-view images as input, which may help improve performance. In addition, 3D reconstruction of objects from images in general, not only humans, is an interesting research direction.

**Acknowledgements** This work was supported by MERL. Xin Yu was partially funded by the NSF grant IIS 1764071. We thank Srikumar Ramalingam for valuable discussions. We also thank the anonymous reviewers for their constructive feedback that helped in shaping the final manuscript.

# References

[1] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Int. Conf. Comput. Vis.*, October 2019. 2

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2014. 6

[3] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3395–3404, 2019. 6

[4] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *Int. Conf. Comput. Vis.*, pages 5420–5430, 2019. 1

[5] 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 *Eur. Conf. Comput. Vis.*, pages 561–578. Springer, 2016. 2, 6

[6] 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 Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Eur. Conf. Comput. Vis.*, pages 561–578, Cham, 2016. Springer International Publishing. 6

[7] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*, 2018. 2

[8] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[9] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Int. Conf. Comput. Vis.*, October 2019. 2

[10] Valentin Gabeur, Jean-Sebastien Franco, Xavier Martin, Cordelia Schmid, and Gregory Rogez. Moulding humans: Non-parametric 3d human shape estimation from single images. In *Int. Conf. Comput. Vis.*, October 2019. 2

[11] Justin Johnson Georgia Gkioxari, Jitendra Malik. Mesh r-cnn. *Int. Conf. Comput. Vis.*, 2019. 2

[12] Ke Gong, Yiming Gao, Xiaodan Liang, Xiaohui Shen, Meng Wang, and Liang Lin. Graphonomy: Universal human parsing via graph transfer learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 2

[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Int. Conf. Comput. Vis.*, pages 2980–2988, Oct 2017. 4

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 4

[15] Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu. Latent structured models for human pose estimation. In *2011 International Conference on Computer Vision*, pages 2220–2227. IEEE, 2011. 6, 11

[16] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1325–1339, jul 2014. 6, 11

[17] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. *Eur. Conf. Comput. Vis.*, 2020. 1, 2

[18] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Brit. Mach. Vis. Conf.*, 2010. doi:10.5244/C.24.12. 6

[19] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1, 6

[20] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 5

[21] Angjoo Kanazawa, Jason Y. Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 2, 6

[22] Ladislav Kavan. Part i: direct skinning methods and deformation primitives. In *ACM SIGGRAPH*, volume 2014, pages 1–11, 2014. 2

[23] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017. 4

[24] 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 *Int. Conf. Comput. Vis.*, October 2019. 6

[25] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019. 1, 2, 4, 5, 6, 7

[26] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[27] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, July 2017. 6

[28] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction. arXiv, 2019. 2

[29] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021. 1, 6, 7, 8, 11

[30] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *Int. Conf. Comput. Vis.*, pages 2999–3007, 2017. 5

[31] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 5

[33] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. *Eur. Conf. Comput. Vis.*, 2020. 2, 6

[34] 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 *IEEE Int. Conf. on 3D Vision (3DV)*, pages 484–494, 2018. 6

[35] Rohit Pandey, Anastasia Tkach, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Ricardo Martin-Brualla, Andrea Tagliasacchi, George Papandreou, Philip Davidson, Cem Keskin, Shahram Izadi, and Sean Fanello. Volumetric capture of humans with a single rgbd camera via semi-parametric learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 2

[36] Georgios Pavlakos, Nikos Kolotouros, and Kostas Daniilidis. Texturepose: Supervising human mesh estimation with texture consistency. In *Int. Conf. Comput. Vis.*, October 2019. 2

[37] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 459–468, 2018. 6

[38] Albert Pumarola, Jordi Sanchez-Riera, Gary P. T. Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3dpeople: Modeling the geometry of dressed humans. In *Int. Conf. Comput. Vis.*, October 2019. 2

[39] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 704–720, 2018. 8, 11

[40] Shunsuke Saito, , Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. *Int. Conf. Comput. Vis.*, 2019. 2

[41] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. 2

[42] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5693–5703, 2019. 2, 4

[43] Yu Sun, Yun Ye, Wu Liu, Wenpeng Gao, Yili Fu, and Tao Mei. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *Int. Conf. Comput. Vis.*, October 2019. 2

[44] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–36, 2018. 6

[45] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 1

[46] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. *Eur. Conf. Comput. Vis.*, 04 2018. 2

[47] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Int. Conf. Comput. Vis.*, October 2019. 2

[48] Yuxin Wu and Kaiming He. Group normalization. In *Eur. Conf. Comput. Vis.*, September 2018. 5

[49] Tao Yu, Zerong Zheng, Yuan Zhong, Jianhui Zhao, Qionghai Dai, Gerard Pons-Moll, and Yebin Liu. Simulcap : Single-view human performance capture with cloth simulation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 2

[50] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 1

[51] Jason Y. Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. In *Int. Conf. Comput. Vis.*, October 2019. 2

[52] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019. 1

[53] Zerong Zheng, Tao Yu, Yixuan Wei, Qionghai Dai, and Yebin Liu. Deephuman: 3d human reconstruction from a single image. In *Int. Conf. Comput. Vis.*, October 2019. 2

[54] 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 Trans. Pattern Anal. Mach. Intell.*, 41(4):901–914, Apr. 2019. 6

# Additional Studies and Results

## Per-Activity Evaluation for Human 3.6M

The Human 3.6M dataset [15, 16] contains people performing 15 activities, such as sitting down and walking. In the main paper, we reported our best model (shown in Table 2 of the main paper), which applies No Weight Sharing . Table 6 and 7 shows the evaluation of non-parametric and SMPL parametric predictions on Human 3.6M for each activity separately. The activities represent Providing Directions, Having a Discussion, Eating, Greeting, Making a Phone Call, Taking a Photo, Posing, Making a Purchase, Sitting, Sitting Down, Smoking, Waiting, Walking a Dog, Walking Together, and Walking respectively. It is clear from the table that the performance for certain activities (highlighted in red), e.g., Walking, compares favorably to others, e.g., Sitting Down, as those poses are much more challenging and consequently have higher errors.

| | P1 | | P2 | |
|---|---|---|---|---|
| Act. | MPJPE↓ | PA-MPJPE↓ | MPJPE↓ | PA-MPJPE↓ |
| Directions | 61.29 | 31.61 | 55.24 | **28.14** |
| Discussion | 59.62 | 33.84 | 56.19 | 32.30 |
| Eating | 58.43 | 34.46 | 55.34 | 33.51 |
| Greeting | 62.68 | 34.81 | 58.36 | 33.53 |
| Phoning | 60.32 | 35.48 | 58.14 | 33.50 |
| Photo | 68.23 | 39.27 | 63.66 | 38.23 |
| Posing | 61.22 | 33.43 | 57.86 | 29.98 |
| Purchases | 60.59 | 32.18 | 59.39 | 31.75 |
| Sitting | 66.63 | 40.51 | 64.92 | 42.21 |
| SittingDown | 72.74 | 49.57 | 72.18 | 45.44 |
| Smoking | 57.77 | 34.09 | 54.06 | 33.21 |
| Waiting | 61.80 | 34.23 | 57.93 | 31.51 |
| WalkDog | 58.09 | 34.50 | 59.16 | 35.31 |
| WalkTogether | 57.36 | 31.41 | 56.49 | 30.79 |
| Walking | **52.72** | **29.46** | **52.03** | 28.66 |
| Overall | 61.17 | 35.36 | 58.45 | 33.96 |

Table 6: Evaluation of non-parametric predictions on Human 3.6M per activity. Certain activities (in red) result in better performance, compared to others. Numbers are MPJPE and PA-MPJPE in mm.

## Non-Parametric vs Parametric

**Result comparison** As shown in Table 6 and 7, the non-parametric shape is the regression result of all mesh vertices from the Bi-layer Graph and is generally able to learn the pose better than the SMPL parametric predictions on Human 3.6M dataset. Please note the evaluation on SMPL parametric predictions is still comparable to the state-of-the-arts as shown in Table 1 in the main paper. As introduced in the main paper, the Bi-layer Graph and SMPL regressor forms a pipeline, and the input of the later module depends on the output of the former module. The good performance of the later module further illustrates that our proposed Bi-layer Graph has addressed the dense regression of body mesh vertices well. Comparing the third and last rows of Figure 6 and Figure **??**, we can observe that some keypoints of the SMPL prediction, e.g. ankles, are slightly off the ground truth while the non-parametric poses are more accurate.

| | P1 | | P2 | |
|---|---|---|---|---|
| Act. | MPJPE↓ | PA-MPJPE↓ | MPJPE↓ | PA-MPJPE↓ |
| Directions | 62.65 | 34.72 | 56.83 | 32.11 |
| Discussion | 62.00 | 37.43 | 58.27 | 36.44 |
| Eating | 63.35 | 38.59 | 59.95 | 37.32 |
| Greeting | 64.49 | 38.63 | 59.96 | 37.30 |
| Phoning | 65.71 | 40.39 | 63.45 | 37.92 |
| Photo | 74.37 | 45.42 | 69.97 | 44.86 |
| Posing | 64.06 | 37.97 | 59.08 | 34.96 |
| Purchases | 65.84 | 37.47 | 62.52 | 37.46 |
| Sitting | 74.62 | 47.14 | 70.46 | 46.33 |
| SittingDown | 79.76 | 53.81 | 79.31 | 51.85 |
| Smoking | 62.57 | 40.04 | 59.10 | 38.79 |
| Waiting | 64.18 | 37.97 | 59.42 | 35.74 |
| WalkDog | 63.57 | 39.57 | 63.71 | 40.44 |
| WalkTogether | 59.60 | 34.73 | 58.83 | 34.15 |
| Walking | **55.09** | **32.87** | **54.02** | **32.05** |
| Overall | 65.35 | 39.91 | 62.16 | 38.56 |

Table 7: Evaluation of SMPL parametric prediction on Human 3.6M per activity. Certain activities (in red) result in better performance, compared to others. Numbers are MPJPE and PA-MPJPE in mm.

## More discussion about other networks

**METRO** Both the transformer-based METRO [29] and our model aim to jointly model vertex-vertex, vertex-joint, and joint-joint interactions. But they differ in the ways of representing each vertex and joint and learning those interactions. METRO uses self-attention to brute-force learn all interactions. Although powerful, the self-attention has a well-known issue of the quadratic time and memory complexity. METRO has to down-sample the mesh to 431 vertices and train on very large mixed datasets for a long time (200 epochs) to learn all the interactions. Rather than the brute-force self-attention, we inject prior knowledge of the mesh topology into the Mesh Graph (1723 vertices), whose adjacency matrix is naturally sparse. In this way, our model trains on a smaller amount of data for 50 epochs and converges faster: the accuracy increases rapidly in the first 12 epochs and becomes stable after 32 epochs as shown in Figure 5 in the main paper. Together with the localized image features, our model achieves comparable performance to self-attention based METRO. We believe that attention and knowledge-aware bi-layer graph network can be integrated to learn the interactions.

**CoMA** Compared to the hierarchy GCN capturing face shape and expression at multiple scales in CoMA [39], our bi-layer graph is simple and efficient to represent non-local body mesh with the additional skeleton-scale graph. Firstly, we use the prior knowledge that the body mesh highly depends on the joint motion. Secondly, extra intermediate-scale body mesh representations by down-sampling (as in CoMA) doesn't help based on our trials. Additionally, our bi-layer graph uses both vertex and joint inputs, rather than just vertices as CoMA does. Our Fusion Graph further learns dynamic vertex-joint correlations, while the transform matrices of down-sampling and up-sampling layers in CoMA are predefined and fixed.

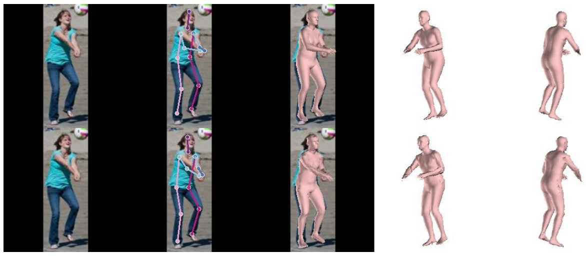


Figure 8: One example of bad pose in different 3D views. The two rows show parametric and non-parametric results respectively.

**Qualitative comparison** On the other side, we note that the shape of the parametric (SMPL) prediction is smooth but the shape of the non-parametric prediction may exhibit non-smooth artifacts. To demonstrate it, we shows examples of the rendered mesh for non-parametric prediction in Figure 9, and the ones for SMPL predictions in Figure 10 for the evaluation images from the Human 3.6M, UP-3D and LSP datasets. To avoid some of the noise artifacts, we can apply some surface constraints, like vertex normal loss, to smooth the predicted surface for non-parametric methods in the future.

## Qualitative results on occlusion

The body is always self-occluded in a single image. It is especially challenging to predict the occluded limbs because of their rich poses from a variety of activities. This requires the methods to deduce the missing limbs from the other visible parts. Our model explicitly embed the whole joints and mesh in the bi-layer structure, enabling to learn the occluded part in a data-driven way. This bi-layer structure further learn the interactions between the body joints and mesh vertices by the fusion graph and thus guide the dense mesh with the sparse joints, which is less challenging to learn from the data. In Figure 9 and 10, We render the predicted meshes from different views and demonstrate that our methods can always learn the occlusion well from the data. We also show a failure case of local hands pose in Figure 8. Although the shape and pose seem correct from the camera view of the image, when viewed from different angles, we can see that the hands are separated rather than joined.

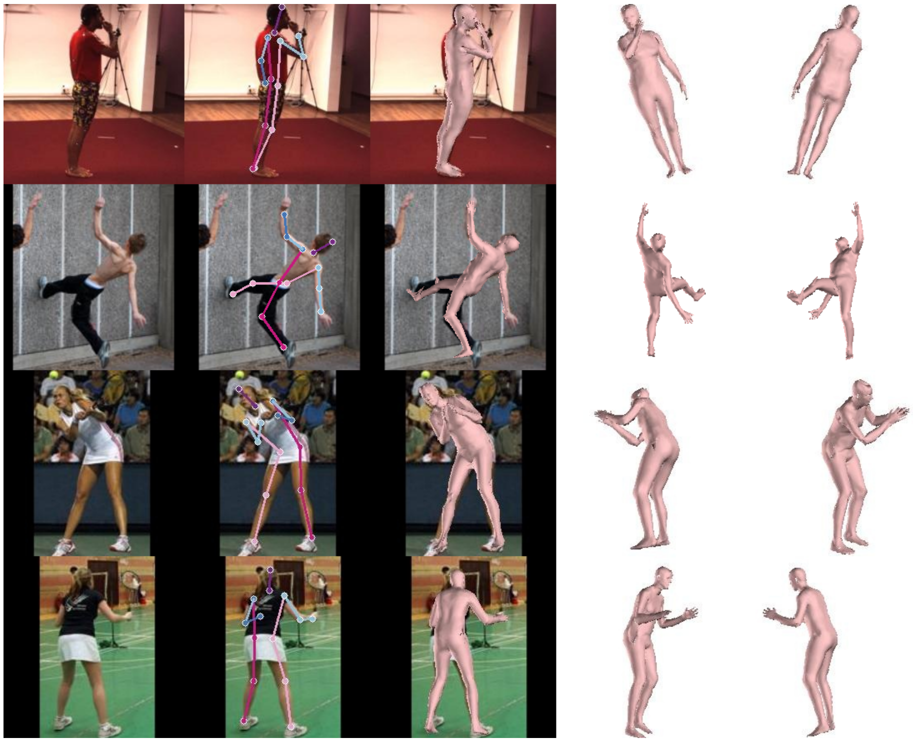


Figure 9: Qualitative non-parametric results. From left to right: input image, pose and shape, two different view perspectives.

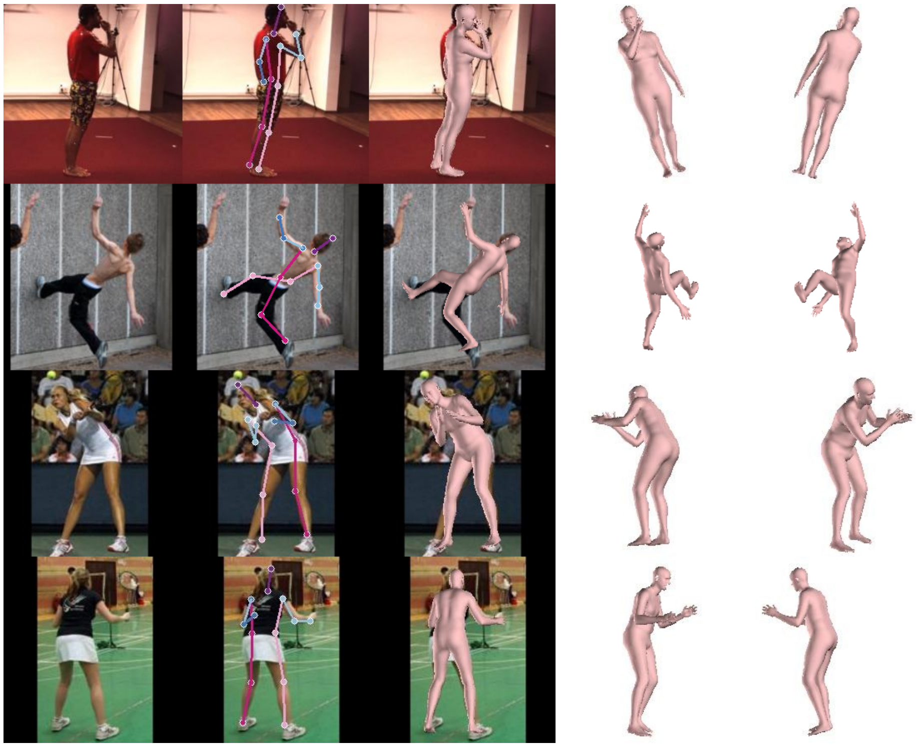


Figure 10: Qualitative SMPL parametric results. From left to right: input image, pose and shape, two different view perspectives.