Learning 3D Human Shape and Pose from Dense Body Parts

Hongwen Zhang, Jie Cao, Guo Lu, Wanli Ouyang, Senior Member, IEEE, and Zhenan Sun, Senior Member, IEEE

Abstract—Reconstructing 3D human shape and pose from monocular images is challenging despite the promising results achieved by the most recent learning-based methods. The commonly occurred misalignment comes from the facts that the mapping from images to the model space is highly non-linear and the rotation-based pose representation of body models is prone to result in the drift of joint positions. In this work, we investigate learning 3D human shape and pose from dense correspondences of body parts and propose a Decompose-and-aggregate Network (DaNet) to address these issues. DaNet adopts the dense correspondence maps, which densely build a bridge between 2D pixels and 3D vertices, as intermediate representations to facilitate the learning of 2D-to-3D mapping. The prediction modules of DaNet are decomposed into one global stream and multiple local streams to enable global and fine-grained perceptions for the shape and pose predictions, respectively. Messages from local streams are further aggregated to enhance the robust prediction of the rotation-based poses, where a position-aided rotation feature refinement strategy is proposed to exploit spatial relationships between body joints. Moreover, a Part-based Dropout (PartDrop) strategy is introduced to drop out dense information from intermediate representations during training, encouraging the network to focus on more complementary body parts as well as neighboring position features. The efficacy of the proposed method is validated on both indoor and real-world datasets including Human3.6M, UP3D, COCO, and 3DPW, showing that our method could significantly improve the reconstruction performance in comparison with previous state-of-the-art methods. Our code is publicly available at https://hongwenzhang.github.io/dense2mesh

Index Terms—3D human shape and pose estimation, decompose-and-aggregate network, position-aided rotation feature refinement, part-based dropout.

1 INTRODUCTION

Reconstructing human shape and pose from a monocular image is an appealing yet challenging task, which typically involves the prediction of the camera and parameters of a statistical body model (e.g. the most commonly used SMPL [1] model). Fig. 1(a) shows an example of the reconstructed result. The challenges of this task come from the fundamental depth ambiguity, the complexity and flexibility of human bodies, and variations in clothing and viewpoint, etc. Classic optimization-based approaches [2], [3] fit the SMPL model to 2D evidence such as 2D body joints or silhouettes in images, which involve complex non-linear optimization and iterative refinement. Recently, regression-based approaches [4], [5], [6], [7] integrate the SMPL model within neural networks and predict model parameters directly in an end-to-end manner.

Though great progress has been made, the direct prediction of the body model from the image space is still complex and difficult even for deep neural networks. In this work, we propose to adopt IUV maps as intermediate representations to facilitate the learning of the mapping from images to models. As depicted in Fig. 1(b), compared with other 2D representations [4], [6], [7], the IUV map could provide more rich information, because it encodes the dense correspondence between foreground pixels on 2D images and vertices on 3D meshes. Such a dense semantic map not only contains essential information for shape and pose estimation from RGB images, but also eliminates the interference of unrelated factors such as appearance, clothing, and illumination variations.

The representation of 3D body model [1], [8] can be factorized into the shape and pose components, depicting the model at different scales. The body shape gives an identity-dependent description about the model, while

- H. Zhang, J. Cao, and Z. Sun are with CRIPEC, NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, and also with the University of Chinese Academy of Sciences, Beijing 101408, China. E-mail: hongwen.zhang@cripac.ia.ac.cn; jie.cao@cripac.ia.ac.cn; znsun@nlpr.ia.ac.cn. (Corresponding author: Zhenan Sun.)
- G. Lu is with the Beijing Institute of Technology, Beijing 100081, China. E-mail: guo.lu@bit.edu.cn.
- W. Ouyang is with the University of Sydney, NSW 2006, Australia. E-mail: wanli.ouyang@sydney.edu.au.
the body pose provides more detailed descriptions about the rotation of each body joint. Previous regression-based methods [5], [7] typically predict them simultaneously using global information from the last layer of the neural network. We observe that the detailed pose of body joints should be captured by local visual cues instead of global information. As shown in Fig. [6], we can estimate the rotations of those visible body joints only based on local visual cues, while the information from other body joints and background regions would be irrelevant.

For the rotation-based pose representation of commonly used body models [1], [8], small rotation errors accumulated along the kinematic chain could lead to large drift of position at the leaf joint. Moreover, the rotation estimation is error-prone for those occluded body joints since their perceptions are less reliable under occlusions. Hence, it is crucial to utilize information from visible body joints and the prior about the structure of human bodies. As shown in previous work [3], [10], the structural information at the feature level is helpful for more robust and accurate pose estimation. However, it is non-trivial to apply these feature refinement methods to our case due to the weak correlation between rotation-based poses of different joints. For instance, the shoulder, elbow, and wrist are three consecutive body joints, and one can hardly infer the relative rotation of wrist w.r.t. the elbow given the relative rotation of elbow w.r.t. the shoulder. On the other hand, we observe that the 3D locations of body joints have stronger correlations than the rotation of body joints. For instance, the positions of shoulder, elbow, and wrist are strongly constrained by the length of the arm.

Based on the observations above, we propose a Decompose-and-aggregate Network (DaNet) to learn 3D human shape and pose from dense correspondences of body parts. As illustrated in Fig. [1(d)], DaNet utilizes IUV maps as the intermediate information for more efficient learning, and decomposes the prediction modules into multiple streams considering that the prediction of different parameters requires the receptive fields with different sizes. To robustly predict the rotations of body joints, DaNet aggregates messages from different streams and refines the rotation features via an auxiliary position feature space to exploit the spatial relationships between body joints. For better generalization, a Part-based Dropout (PartDrop) strategy is further introduced to drop out dense information from intermediate representations during training, which could effectively regularize the network and encourage it to learn features from complementary body parts and leverage information from neighboring body joints. As will be validated in our experiments, all the above new designs could contribute to better part-based learning and improve the reconstruction performance. To sum up, the main contributions in this work are listed as follows.

- We comprehensively study the effectiveness of adopting the IUV maps in both global and local scales, which contains densely semantic information of body parts, as intermediate representations for the task of 3D human pose and shape estimation.
- Our reconstruction network is designed to have decomposed streams to provide global perception for the camera and shape prediction while detailed perception for pose prediction of each body joint.
- A part-based dropout strategy is introduced to drop dense information from intermediate representations during training. Such a strategy can encourage the network to learn features from complementary body parts, which also has the potential for other structured image understanding tasks.
- A position-aided rotation feature refinement strategy is proposed to aggregate messages from different part features. It is more efficient to exploit the spatial relationship in an auxiliary position feature space since the correlations between position features are much stronger.

An early version of this work appeared in [11]. We have made significant extensions to our previous work in three main aspects. First, the methodology is improved to be more accurate and robust thanks to several new designs, including the part-based dropout strategy for better generalization performance and the customized graph convolutions for more efficient and better feature mapping and refinement. Second, more extensive evaluations and comparisons are included to validate the effectiveness of our method, including evaluations on additional datasets and comparisons of the reconstruction errors across different human actions and model surface areas. Third, more discussions are provided in our ablation studies, including comprehensive evaluations on the benefit of adopting IUV as intermediate representations and in-depth analyses on the refinement upon the rotation feature space and position feature space.

The remainder of this paper is organized as follows. Section 2 briefly reviews previous work related to ours. Section 3 provides preliminary knowledge about the SMPL model and IUV maps. Details of the proposed network are presented in Section 4. Experimental results and analyses are included in Section 5. Finally, Section 6 concludes the paper.

2 RELATED WORK

2.1 3D Human Shape and Pose Estimation

Early pioneering work on 3D human model reconstruction mainly focuses on the optimization of the fitting process. Among them, [12], [13] fit the body model SCAPE [8] with the requirement of ground truth silhouettes or manual initialization. Bogo et al. [2] introduce the optimization method SMPLify and make the first attempt to automatically fit the SMPL model to 2D body joints by leveraging multiple priors. Lassner et al. [5] extend this method and improve the reconstruction performance by incorporating silhouette information in the fitting procedure. These optimization-based methods typically rely on accurate 2D observations and the prior terms imposed on the shape and pose parameters, making the procedure time-consuming and sensitive to the initialization. Alternatively, recent regression-based methods employ neural networks to predict the shape and pose parameters directly and learn the priors in a data-driven manner. These efforts mainly focus on several aspects including intermediate representation leveraging, architecture designs, structural information modeling, and re-projection loss designs, etc. Our work makes contributions to the first three aspects above and is also complementary to the work focusing on the re-projection loss designs [4], [14].
The recovery of the 3D human pose from a monocular image is challenging. Common strategies use intermediate estimations as the proxy representation to alleviate the difficulty. These methods can benefit from existing state-of-the-art networks for lower-level tasks. For the recovery of 3D human pose or human model, 2D joint positions, silhouette, segmentation, depth maps, joint heatmaps, volumetric representation, and 3D orientation fields are adopted in literature as intermediate representations to facilitate the learning task. Though the aforementioned representations are helpful for the task, detailed information contained within body parts is missing in these coarse representations, which becomes the bottleneck for fine-grained prediction tasks. Recently, DensePose regresses the IUV maps directly from images, which provides the dense correspondence mapping from the image to the human body model. However, the 3D model cannot be directly retrieved from such a 2.5D projection. In our work, we propose to adopt such a densely semantic map as the intermediate representation for the task of 3D human shape and pose estimation. To the best of our knowledge, we are among the first attempts to leverage IUV maps for 3D human model recovery. In comparison, the major differences between concurrent efforts and ours lie in three aspects: 1) obtain IUV predictions from a pretrained network of DensePose, while our work augments the annotations of 3D human pose datasets with the rendered ground-truth IUV maps and imposes dense supervisions on the intermediate representations; 2) only leverage global IUV maps, while our work exploits using IUV maps in both global and local scales; 3) resorts to involving more synthetic IUV maps as additional training data while our work introduces the part-based dropout upon IUV maps to improve generalization. We believe these concurrent work complement each other and enrich the research community.

2.1.2 Architecture Design
Existing approaches to 3D human shape and pose estimation have designed a number of network architectures for more effective learning of the highly nonlinear image-to-model mapping. Tan et al. develop an encoder-decoder based framework where the decoder learns the SMPL-to-silhouette mapping from synthetic data and the encoder learns the image-to-SMPL mapping with the decoder frozen. Kanazawa et al. present an end-to-end framework HMR to reconstruct the SMPL model directly from images using a single CNN with an iterative regression- and optimization-based methods. Kolotouros et al. enhance HMR with the fitting process of SMPLify to incorporate regression- and optimization-based methods. Pavlakos et al. propose to predict the shape and pose parameters from the estimated silhouettes and joint locations respectively. Sun et al. also leverage joint locations and further involve deep features into the prediction process. Instead of regressing the shape and pose parameters directly, Kolotouros et al. employ a Graph CNN to regress the 3D coordinates of the human mesh vertices, while Yao et al. regress the 3D coordinates in the form of an unwrapped position map. All aforementioned regression-based methods predict the pose in a global manner. In contrast, our DaNet predicts joint poses from multiple streams, hence the visual cues could be captured in a fine-grained manner. Recently, Güler et al. also introduce a part-based reconstruction method to predict poses from the deep features pooled around body joints. In comparison, the pooling operation of our DaNet is performed on intermediate representations, enabling detailed perception for better pose feature learning. Moreover, existing approaches for rotation-based pose estimation do not consider feature refinement, while DaNet includes an effective rotation feature refinement scheme for robust pose predictions.

2.1.3 Structural Information Modeling
Leveraging the articulated structure information is crucial for human pose modeling. Recent deep learning-based approaches to human pose estimation incorporate the structured feature learning in their network architecture designs. All these efforts exploit the relationship between the pose features of body joints and their feature refinement strategies are only validated on the position-based pose estimation problem. Our work is complementary to them by investigating the refinement for rotation features under the context of the rotation-based pose representation, which paves a new way to impose structural constraints upon rotation features. Our solution aggregates the rotation features into a position feature space, where the aforementioned structural feature learning approaches could be easily applied.

For more geometrically reasonable pose predictions, different types of pose priors are also employed as constraints in the learning procedure. For instance, Akhter and Black learn the pose prior in the form of joint angle constraints. Sun et al. design handcrafted constraints such as limb-lengths and their proportions. Similar constraints are exploited in under the weakly-supervised setting. For the rotation-based pose representation in the SMPL model, though it inherently satisfies structure constraints such as limb proportions, the pose prior is still essential for better reconstruction performance. SMPLify imposes several penalizing terms on predicted poses to prevent unnatural results. Kanazawa et al. introduce an adversarial prior for guiding the prediction to be realistic. All these methods consider the pose prior at the output level. In our work, we will exploit the relationship at the feature level for better 3D pose estimation in the SMPL model.

2.2 Regularization in Neural Networks
Regularization is important to neural networks for better generalization performance. A number of regularization techniques have been proposed to remove features from neural networks at different granularity levels. Among them, dropout is commonly used at the fully connected...
layers of neural networks to drop unit-wise features independently. The introduction of dropout has inspired the development of other dropping out strategies with structured forms. For instance, SpatialDropout [59] drops channel-wise features across the entire feature map, while DropBlock [60] drops block-wise features in contiguous regions. Different from these techniques, our PartDrop strategy drops part-wise features at the granularity level of semantic body parts. Such a part-wise dropping strategy could remove patterns from these techniques, our PartDrop strategy drops part-wise features at the granularity level of semantic body parts. This way, the pixels on the foreground could be projected back to vertices on the template mesh according to a predefined bijective mapping between the 3D surface space and the 2D UV space. For the human body model, the correspondence could have finer granularity by introducing the Index from 1 and 24 are also normalized into the [0, 1] interval. (a) Generation of ground truth IUV Maps for 3D human body models.

3 SMPL Model and IUV Maps

SMPL Model. The Skinned Multi-Person Linear model (SMPL) [11] is one of the widely used statistical human body models, which represent the body mesh with two sets of parameters, i.e., the shape and pose parameters. The shape indicates the model’s height, weight and limb proportions while the pose indicates how the model deforms with the rotated skeleton joints. Such decomposition of shape and pose makes it convenient for algorithms to focus on one of these two factors independently. In the SMPL model, the shape parameters \( \beta \in \mathbb{R}^{10} \) denote the coefficients of the PCA basis of the body shape. The pose parameters \( \theta \in \mathbb{R}^{3K} \) denote the axis-angle representations of the relative rotation of \( K \) skeleton joints with respect to their parents in the kinematic tree, where \( K = 24 \) in the SMPL model. For simplicity, the root orientation is also included as the pose parameters of the root joint in our formulation. Given the pose and shape parameters, the model deforms accordingly and generates a triangulated mesh with \( N = 6890 \) vertices \( M(\theta, \beta) \in \mathbb{R}^{3 \times N} \). The deformation process \( M(\theta, \beta) \) is differentiable with respect to the pose \( \theta \) and shape \( \beta \), which means that the SMPL model could be integrated within a neural network as a typical layer without any learnable weights. After obtaining the final mesh, vertices could be further mapped to sparse 3D keypoints by a pretrained linear regressor.

IUV Maps. Reconstructing the 3D object model from a monocular image is ambiguous, but there are determinate correspondences between foreground pixels on 2D images and vertices on 3D surfaces. Such correspondences could be represented in the form of UV maps, where the foreground pixels contain the corresponding UV coordinate values. In this way, the pixels on the foreground could be projected back to vertices on the template mesh according to a predefined bijective mapping between the 3D surface space and the 2D UV space. For the human body model, the correspondence could have finer granularity by introducing the Index of the body parts [39], [62], which results in the IUV maps \( H = (H^i | H^u | H^v) \in \mathbb{R}^{(1+P) \times h_{iuv} \times w_{iuv} \times 3} \), where \( P \) denotes the number of body parts, \( h_{iuv} \) and \( w_{iuv} \) denote the height and width of IUV maps. The Index channels \( H^i \) indicates whether a pixel belongs to the background or a specific body part, while the UV channels \( H^u \) and \( H^v \) contain the corresponding \( U, V \) values of visible body parts respectively. The IUV maps \( H \) encode Index, \( U \), and \( V \) values individually for \( P \) body parts in a one-hot manner along \( (1+P) \) ways. The Index values for body parts count from 1 and Index 0 is reserved for the background. For each body part, the UV space is independent so that the representation could be more fine-grained. The IUV annotation of the human body is firstly introduced in DenseReg [62] and DensePose [39]. Figs. 2(a)(b)(c) show the Index, \( U \), and \( V \) values on the SMPL model as defined in DensePose [39].

Preparation of IUV Maps for 3D Human Pose Datasets. Currently, there is no 3D human pose dataset providing IUV annotations. In this work, for those datasets providing SMPL parameters with human images, we augment their annotations by rendering the corresponding ground-truth IUV maps based on the same IUV mapping protocol of DensePose [39]. Specifically, we first construct a template texture map from IUV values of each vertex on the SMPL model, and then employ a renderer to generate IUV maps. As illustrated in Fig. 2(d) for each face in the triangulated mesh, the texture values used for rendering is a triplet vector denoting the corresponding Index, \( U \), and \( V \) values. Then, given SMPL models, the corresponding IUV maps can be generated by existing rendering algorithms such as [63]. [64]. Specifically, the renderer takes the template texture map and 3D model as inputs and output a rendered image with the size of \( h_{iuv} \times w_{iuv} \times 3 \). Afterwards, the rendered image is reorganized as the shape of \( (1+P) \times h_{iuv} \times w_{iuv} \times 3 \) by converting values into one-hot representations.

4 Methodology

As illustrated in Fig. 3 our DaNet decomposes the prediction task into one global stream for the camera and shape predictions and multiple local streams for joint pose predictions. The overall pipeline involves two consecutive stages. In the first stage, the IUV maps are estimated from global and local perspectives in consideration of the different sizes
of the receptive fields required by the prediction of different parameters. In the second stage, the global and local IUV maps are used for different feature extraction and prediction tasks. The global features are extracted from global IUV maps and then directly used to predict camera and body shape. The rotation features are extracted from partial IUV maps and then directly used to predict camera and SMPL parameters. In the second stage, the global and local IUV maps are used for different feature extraction and prediction tasks. Specifically, a fully convolutional network is employed to produce \( K + 1 \) sets of IUV maps, including one set of global IUV maps and \( K \) sets of partial IUV maps for the corresponding \( K \) body joints. The global IUV maps are aligned with the original image through up-sampling, while the partial IUV maps are centered around the body joints. The feature maps outputted from the last layer of the FCN would be shared by the estimation tasks of both global and partial IUV maps. The estimation of the global IUV maps is quite straightforward since they could be obtained by simply feeding these feature maps into a convolutional layer. For the estimation of each set of partial IUV maps, a joint-centric RoI pooling would be first performed on these feature maps to extract appropriate sub-regions, which results in \( K \) sets of partial feature maps. Then, the \( K \) sets of partial IUV maps would be estimated independently from these partial feature maps. Now, we will give details about the RoI pooling process for partial IUV estimation.

### Joint-centric RoI Pooling
For pose parameters in the SMPL model, they represent the relative rotation of each body joint with respect to its parent in the kinematic tree. Hence, the perception of joint poses should individually focus on corresponding body parts. In other words, globally zooming, translating the human in the image should have no effect on the pose estimation of body joints. Moreover, the ideal scale factors for the perception of joint poses should vary from one joint to another since the proportions of body parts are different. To this end, we perform joint-centric RoI pooling on feature maps for partial IUV estimation. Particularly, for each body joint, a sub-region of the feature maps is extracted and spatially transformed to a fixed resolution for subsequent partial IUV map estimation and joint pose prediction. In our implementation, the RoI pooling is accomplished by a Spatial Transformer Network (STN) [65]. In comparison with the conventional STNs, the pooling process in our network is learned in an explicitly supervised manner.

As illustrated in Fig. 5(a), the joint-centric RoI pooling operations are guided by 2D joint positions so that each sub-region is centered around the target joint. Specifically, 2D joint heatmaps are estimated along with the global IUV maps in a multi-task learning manner, and 2D joint positions are retrieved from heatmaps using the soft-argmax [66] operation. Without loss of generality, let \( j_k \) denote the position of the \( k \)-th body joint. Then, the center and scale parameters used for spatial transformation are determined individually for each set of partial IUV maps. Specifically, for the \( k \)-th set of partial IUV maps, the center \( c_k \) is the position of the \( k \)-th joint, while the scale \( s_k \) is proportional to the size of the foreground region, i.e.,

\[
c_k = j_k, \quad s_k = \alpha_k \max(w_{bbox}, h_{bbox}) + \delta,
\]

where \( \alpha_k \) and \( \delta \) are two constants, \( w_{bbox} \) and \( h_{bbox} \) denote the width and height of the foreground bounding box respectively. In our implementation, the foreground is obtained from the part segmentation (i.e., Index channels of estimated IUV maps). Compared with our previous work [11] calculating \( s_k \) from 2D joints, the \( s_k \)s determined by foreground regions here are more robust to 2D joint localization.
2D Joint Position Transformation

Joint. The partial IUV maps before and after the simplification are depicted in Fig. 4(b) and Fig. 4(c) respectively. For each set of partial IUV maps, we retain specific channels by discarding those irrelevant body parts. For the k-th body joint, the extracted rotation features would be refined (see Sec. 4.3) and then used to predict the rotation matrix $R_k \in \mathbb{R}^{3 \times 3}$ via a fully connected layer. Here, we follow previous work [6, 7] to predict the rotation matrix representation of the pose parameters $\theta$ rather than the axis-angle representation defined in the SMPL model. Note that using other rotation representations such as the 6D continuous representation is also feasible. An $L_1$ loss is imposed on the predicted camera, shape, and pose parameter, and we denote it as $L_{\text{smpl}}$.

Following previous work [5], [6], [7], we also add additional constraint and regression objective for better performance. For the predicted rotation matrix, we impose an orthogonal constraint loss $L_{\text{orth}} = \sum_{k=0}^{K-1} \| R_k R_k^\top - I \|_2$ upon the predicted rotation matrices $[R_k]_{k=0}^{K-1}$ to guarantee their orthogonality. Moreover, given the predicted SMPL parameters, the performance could be further improved by adding supervision explicitly on the resulting model $M(\theta, \beta)$. Specifically, three $L_1$ based loss functions are used to measure the difference between the ground-truth positions and the predicted ones. The corresponding losses are denoted as $L_{\text{cert}}$ for vertices on 3D mesh, $L_{3D\text{kp}}$ for sparse 3D human keypoints, and $L_{\text{reproj}}$ for the reprojected 2D human keypoints, respectively. For the sparse 3D human keypoints, the predicted positions are obtained via a pretrained linear regressor by mapping the mesh vertices to the 3D keypoints defined in human pose datasets. Overall, the objective in this prediction stage is the weighted sum of multiple losses:

$$L_{\text{target}} = \lambda_{\text{smpl}} L_{\text{smpl}} + \lambda_{\text{orth}} L_{\text{orth}} + \lambda_{\text{point}} (L_{\text{cert}} + L_{3D\text{kp}} + L_{\text{reproj}}),$$

where $\lambda_{\text{smpl}}$, $\lambda_{\text{orth}}$, and $\lambda_{\text{point}}$ are balance weights.

Part-based Dropout. Our approach learn the shape and pose from the IUV intermediate representation, which contains dense correspondences of the body parts. Following previous work on data augmentation [61] and model regularization [58], [60], we introduce a Part-based Dropout (PartDrop) strategy to drop out semantic information from intermediate representations during training. PartDrop has a dropping rate $\gamma$ as the probability of dropping values in the estimated IUV maps. In contrast to other dropping out strategies such as Dropout [59] and DropBlock [60], the proposed PartDrop strategy drops features in contiguous regions at the granularity level of body parts. Specifically, for each training sample, the index subset $I_{\text{drop}}$ of the body
4.3 Rotation Feature Refinement

In our approach, the rotation features extracted in local streams are aggregated to exploit spatial relationships among body joints. As illustrated in Fig. 7(a), the position-aided rotation feature refinement involves three consecutive steps, namely rotation feature collection, position feature refinement, and refined feature conversion. Specifically, the rotation features are first collected into the position feature space where the feature refinement is performed. After that, the rotation feature refinement is accomplished by converting the refined position features back to the rotation feature space. All these three steps are performed by customized graph convolution layers. In particular, we consider the following graph-based convolution layer \( \mathcal{G}(\cdot) \) that employs one popular formulation of the Graph Convolution Networks as proposed in Kipf et al. [49]:

\[
Z_{out} = \mathcal{G}(A, Z_{in}) = \sigma(\tilde{A}Z_{in}W),
\]

where \( Z_{in} \) and \( Z_{out} \) are input and output features respectively, \( \sigma(\cdot) \) is the activation function, \( W \) is the parameters of convolution kernels, \( \tilde{A} \) denotes the row-normalized matrix of the graph adjacency matrix \( A \), i.e., \( \tilde{A} = D^{-1/2}AD^{-1/2} \) if \( A \) is a symmetric matrix, and otherwise \( \tilde{A} = D^{-1}A \), where \( D \) is the diagonal node degree matrix of \( A \) with \( D_{ii} = \sum_j A_{ij} \). For simplicity, we also refer to the graph with adjacency matrix of \( A \) as graph \( A \).

**Step 1: Rotation Feature Collection.** Note that the rotation of each body joint could be viewed as sequential data along the kinematic chain. This is inspired by the fact that the human could act in a recurrent manner according to the kinematic tree shown in Fig 7(b). The position of a specific body joint can be calculated from the collection of the relative rotations and bone lengths of those joints belonging to the same kinematic chain. At the feature level, we propose to learn the mapping from rotation feature space to position feature space. To that end, one graph convolution layer is customized to gather information from body joints along the kinematic chain and learn such mapping. Formally, let \( X \in \mathbb{R}^{K \times C} \) denote the rotation features extracted from \( K \) sets of partial IUV maps with \( C \) being the feature dimension. The position features \( Y \in \mathbb{R}^{K \times C} \) of \( K \) joints is obtained by feeding \( X \) to the graph convolution, i.e.,

\[
Y = \mathcal{G}(A^{rp}, X),
\]

where the graph with adjacency matrix \( A^{rp} \) is customized as a collection graph for mapping rotation features into the position feature space, in which \( A^{rp}_{ij} = 1 \) if the \( j \)-th joint is one of the ancestors of the \( i \)-th joint along the kinematic chain, and otherwise \( A^{rp}_{ij} = 0 \). The adjacency matrix \( A^{rp} \) of the collection graph is depicted in Fig. 7(c).

**Step 2: Position Feature Refinement.** Since there are strong spatial correlations among neighboring body joints, utilizing such structured constraints could effectively improve the features learned at each joint. Towards this goal, a graph-based convolution network is employed to exploit spatial relationships between joints. Specifically, the position features \( Y \) are fed into \( L \) graph convolution layers with the following layer-wise formulation:

\[
Y^{(l)} = \mathcal{G}(A^{rf}, Y^{(l-1)}),
\]

where \( Y \) denotes the position features obtained from the \( l \)-th layer with \( Y^{(0)} = Y \), and the graph with adjacency matrix \( A^{rf} = I + \tilde{A}^{rf} \) serves as a refinement graph for feature refinement, in which \( \tilde{A}^{rf}_{ij} = 1 \) if the \( i \)-th and \( j \)-th joints are neighboring, and otherwise \( \tilde{A}^{rf}_{ij} = 0 \). After graph convolutions, the refined position features \( \hat{Y} \) are obtained by adding \( \hat{Y} \) with the original position features \( Y \) in a residual manner, i.e., \( \hat{Y} = Y + \hat{Y} \). Fig. 7(d) shows an example of the adjacency matrix \( A^{rf} \) which considers both one-hop and two-hop neighbors. Note that \( A^{rf} \) could have various forms according to the neighbor definition of body joints.

Inspired by previous work [52, 69], we also add a learnable edge weighting mask on the graph convolution of this step since messages from different joints should have different contributions to the feature refinement of the target joint. In this way, we have the adjacency matrix \( A^{rf} \) improved as

\[
A^{rf} = I + M \circ \tilde{A}^{rf},
\]
where \(\circ\) denotes the element-wise product, \(M \in [0, 1]^{K \times K}\) is the learnable edge weighting matrix serving as an attention mask of the graph to balance the contributions of neighboring features to the target feature.

**Step 3: Refined Feature Conversion.** The last step of refinement is to convert the refined features back to the original rotation feature space. Since the rotation and position of body joints are two mutual representation of 3D human pose, after the refinement of position features, the rotation features can be refined accordingly. Specifically, for the \(k\)-th body joint, its rotation feature can be refined by aggregating messages from the refined position features of three consecutive body joints, i.e., the joint itself and its parent and child joints. Similar to the first step, the mapping from position features to rotation features is also learned via a graph-based convolution layer, where the difference lies in the adjacency matrix of the graph. Formally, the refined position features \(\hat{Y}\) are fed into the graph to obtain features in the rotation space, resulting in the refined rotation features \(\hat{X}\) for the final prediction of joint pose parameters, i.e.,

\[
\hat{X} = \mathcal{G}(A^{p2r}, \hat{Y}),
\]

where the graph with adjacency matrix \(A^{p2r} = I + \tilde{A}^{p2r}\) is customized as a conversion graph for mapping position features to rotation features, in which \(\tilde{A}^{p2r}\) is the adjacency matrix of the conversion graph depicted in Fig. 7(e).

**Supervision in Refinement.** The rotation and position feature spaces are built under corresponding supervisions during training. As illustrated in Fig. 7(a), the rotation features \(X\) and \(\hat{X}\) are used to predict joint rotations, while the position features \(Y\) and \(\hat{Y}\) are used to predict joint positions. \(L_1\) based rotation and position supervisions are imposed on these predictions correspondingly, which compose the objective \(L_{refine}\) involved in the refinement procedure. Note that these intermediate predictions are unnecessary during testing.

## 5 Experiments

### 5.1 Implementation Details

The FCN for IUV estimation in our framework adopts the architecture of HRNet-W48 [70], which is one of the most recent state-of-the-art networks for dense estimation tasks. The FCN receives the 224 \times 224 input and produces 56 \times 56 feature maps for estimating global and local IUV maps with the same resolution. The IUV estimation network is initialized with the model pretrained on the COCO keypoint detection dataset [71], which is helpful for robust joint-centric RoI pooling and partial IUV estimation. Two ImageNet-pretrained ResNet-18 [67] are employed as the backbone networks for global and rotation feature extraction respectively. During training, data augmentation techniques, including color jittering and flipping, are applied randomly to input images. Random rotation is used when in-the-wild datasets are involved for training. The IUV estimation task is first trained for 5k iterations before involving the parameter prediction task. The \(\alpha_k\)s in Eq. (2) are first learned using ground-truth IUV maps as inputs and then frozen as constants for other experiments, while \(\delta\) is empirically set to 0.1. The hyper-parameters \(\lambda\)s are decided based on the scales of values in objectives. The dropping rate \(\gamma\) for PartDrop is adopted as 0.3 in our experiments. For more robust pose recovery from the estimated partial IUV, we perform random jittering on the estimated 2D joint position and the scale of partial IUV maps during training. Following previous work [5], [43], the predicted poses are initialized from the mean pose parameters. For faster runtime, the local streams are implemented to run in a parallel manner. Specifically, the partial IUV maps of all body joints are concatenated batch-wise and then fed into the backbone feature extractor. Moreover, individual rotation feature extraction is implemented based on group convolution. By default, we adopt the Adam [2] optimizer with an initial learning rate of \(1 \times 10^{-4}\) to train our model, and reduce the learning rate to \(1 \times 10^{-5}\) after 30k iterations. The learning process converges after around 60k iterations and takes about 25 hours on a single TITAN Xp GPU. During testing, due to the fundamental depth-scale ambiguity, we follow previous work [5], [4] to center the person within the image and perform scaling such that the inputs have the same setting as training. Our experiments are implemented in PyTorch [73]. More implementation details could be found in the publicly available code.

### 5.2 Datasets and Evaluation Metrics

**Human3.6M.** Human3.6M [74] is a large-scale dataset which consists of 3.6 millions of video frames captured in the controlled environment, and currently the most commonly used benchmark dataset for 3D human pose estimation.
Kanazawa et al. [3] generated the ground truth SMPL parameters by applying MoSH [27] to the sparse 3D MoCap marker data. Following the common protocols [5, 6, 33], we use five subjects (S1, S5, S6, S7, S8) for training and two subjects (S9, S11) for evaluation. We also down-sample the original videos from 50fps to 10fps to remove redundant frames, resulting in 312,188 frames for training and 26,859 frames for testing.

**UP-3D.** UP-3D [3] is a collection dataset of existing 2D human pose datasets (i.e., LSP [76], LSP-extended [77], MPII HumanPose [78], and FashionPose [79]), containing 5,703 images for training, 1,423 images for validation, and 1,389 images for testing. The SMPL parameter annotations of these real-world images are augmented in a semi-automatic way by using an extended version of SMPLify [3].

**COCO.** The COCO dataset [71] contains a large scale of images and person instances labeled with 17 keypoints. Based on the COCO dataset, DensePose-COCO [39] further provides the dense correspondences from 2D images to the 3D surface of the human body model for 50K humans. Different from our rendered IUV maps, the correspondence annotations in DensePose-COCO only consist of approximately 100-150 points per person, which are a sparse subset of the foreground pixels of human images. In our experiments, we discard those persons without 2D keypoint annotations, resulting in 39,210 samples for training. Since there are no ground-truth shape and pose parameters for COCO, we evaluate our method quantitatively on the keypoint localization task using its validation set, which includes 50,197 samples.

**3DPW.** The 3DPW dataset [80] is a recent in-the-wild dataset providing accurate shape and pose ground truth annotations. This dataset captured IMU-equipped actors in challenging outdoor scenes with various activities. Following previous work [15, 43], we do not use its data for training but perform evaluations on its defined test set only. There are 35,515 samples extracted from videos for testing.

**Fitted SMPL labels from SPIN.** Kolotouros et al. [43] proposed SPIN to incorporate a fitting procedure within the training of a SMPL regressor. The regressor provided better initialization for the fitting of human models to 2D keypoints, and the resulting SMPL parameters could be more accurate than those fitted in a static manner. For evaluation on 3DPW [80], our model would be supervised with the final fitted SMPL labels from SPIN [43] for in-the-wild datasets including LSP [76], LSP-Extended [77], MPII [78], COCO [71], and MPI-INF-3DHP [81].

**Evaluation Metrics.** Following previous work [6, 15, 34], for evaluating the reconstruction performance, we adopt the mean Per-vertex Error (PVE) as the primary metric, which is defined as the average point-to-point Euclidean distance between the predicted model vertices and the ground truth model vertices. Besides the PVE metric, we further adopt PVE-S and PVE-P as secondary metrics for separately evaluate the shape and pose prediction results. The PVE-S computes the per-vertex error with the pose parameters of ground truth and predicted models set as zeros (i.e., models under the rest pose [1]), while the PVE-P computes the analogous per-vertex error with the shape parameters set as zeros. For the Human3.6M dataset, the widely used Mean Per Joint Position Error (MPJPE) and the MPJPE after rigid alignment of the prediction with ground truth using Procrustes Analysis (MPJPE-PA) are also adopted to quantitatively evaluate the 3D human pose estimation performance. The above three metrics will be reported in millimeters (mm) by default.

For the keypoint localization task on COCO, the commonly-used metric is the Average Precision (AP) defined by its organizer. The keypoint localization AP is calculated based on the Object Keypoint Similarity (OKS), which plays the same role as the IoU in object detection. We report results using the mean AP, and the variants of AP including AP50 (AP at OKS = 0.50), AP75 (AP at OKS = 0.75), APM for medium objects, and APL for large objects.

### 5.3 Comparison with State-of-the-art Methods

#### 5.3.1 Comparison on the Indoor Dataset.

**Evaluation on Human3.6M.** We evaluate the 3D human mesh recovery as well as pose estimation performance for quantitative comparison on Human3.6M, where our model is trained on its training set. Table 1 reports the comparison results with previous methods that output more than sparse 3D joint positions. For regression-based methods in Table 1 different architectures have been designed to predict the shape and pose parameters. Among them, HMR [5] adopts a single CNN and an iterative regression module to produce all parameters. Pavlakos et al. [6] decompose the shape and pose prediction tasks, while their pose parameters are predicted from 2D joints positions. NBF [7] adopts segmentation as the intermediate representation and learns all parameters from it. CMR [40] directly regresses 3D meshes with a graph-based convolutional network. All these architectures estimate pose parameters through a single stream with an exception that HoloPose [14] regresses poses using a part-based model. As can be seen from Table 1, our network significantly outperforms the above-mentioned architectures. It’s worth noting that the methods reported in Table 1 are not strictly comparable since they may use

<table>
<thead>
<tr>
<th>Method</th>
<th>PVE</th>
<th>MPJPE</th>
<th>MPJPE-PA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou et al. [54]</td>
<td>-</td>
<td>107.3</td>
<td>-</td>
</tr>
<tr>
<td>Tung et al. [4]</td>
<td>-</td>
<td>98.4</td>
<td></td>
</tr>
<tr>
<td>SMPLify</td>
<td>202.0</td>
<td>-</td>
<td>82.3</td>
</tr>
<tr>
<td>SMPLify++</td>
<td>-</td>
<td>80.7</td>
<td></td>
</tr>
<tr>
<td>Pavlakos et al. [6]</td>
<td>155.5</td>
<td>-</td>
<td>75.9</td>
</tr>
<tr>
<td>HMR [5]</td>
<td>-</td>
<td>88.0</td>
<td>56.8</td>
</tr>
<tr>
<td>NBF [7]</td>
<td>-</td>
<td>-</td>
<td>59.9</td>
</tr>
<tr>
<td>Xiang et al. [38]</td>
<td>-</td>
<td>65.6</td>
<td></td>
</tr>
<tr>
<td>Arnab et al. [17]</td>
<td>-</td>
<td>77.8</td>
<td>54.3</td>
</tr>
<tr>
<td>CMR [40]</td>
<td>-</td>
<td>-</td>
<td>50.1</td>
</tr>
<tr>
<td>HoloPose [14]</td>
<td>-</td>
<td>60.3</td>
<td>46.5</td>
</tr>
<tr>
<td>TexturePose [19]</td>
<td>-</td>
<td>49.7</td>
<td></td>
</tr>
<tr>
<td>DensePose [43]</td>
<td>-</td>
<td>76.8</td>
<td>48.0</td>
</tr>
<tr>
<td>DaNet-LSTM [11]</td>
<td>75.1</td>
<td>61.5</td>
<td>48.6</td>
</tr>
<tr>
<td>Ours</td>
<td>66.5</td>
<td>54.6</td>
<td>42.9</td>
</tr>
</tbody>
</table>

2. https://cocodataset.org/#keypoints-eval
Table 2
Quantitative comparison of MPJPE-PA across different actions on the Human3.6M dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Direct</th>
<th>Discuss</th>
<th>Eating</th>
<th>Greet</th>
<th>Phone</th>
<th>Photo</th>
<th>Pose</th>
<th>Purch.</th>
<th>Sitting</th>
<th>SittingD.</th>
<th>Smoke</th>
<th>Wait</th>
<th>WalkD.</th>
<th>Walk</th>
<th>WalkT.</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pavlakos et al. [33]</td>
<td>47.5</td>
<td>50.5</td>
<td>48.3</td>
<td>49.3</td>
<td>50.7</td>
<td>55.2</td>
<td>46.1</td>
<td>48.0</td>
<td>61.1</td>
<td>78.1</td>
<td>51.1</td>
<td>48.3</td>
<td>52.9</td>
<td>41.5</td>
<td>46.4</td>
<td>51.9</td>
</tr>
<tr>
<td>Martinez et al. [28]</td>
<td>39.5</td>
<td>43.2</td>
<td>46.4</td>
<td>47.0</td>
<td>51.0</td>
<td>56.0</td>
<td>41.4</td>
<td>40.6</td>
<td>56.5</td>
<td>69.4</td>
<td>49.2</td>
<td>45.0</td>
<td>49.5</td>
<td>38.0</td>
<td>43.1</td>
<td>47.7</td>
</tr>
<tr>
<td>SMPLify [2]</td>
<td>62.0</td>
<td>60.2</td>
<td>67.8</td>
<td>76.5</td>
<td>92.1</td>
<td>77.0</td>
<td>73.0</td>
<td>100.3</td>
<td>83.4</td>
<td>77.3</td>
<td>86.8</td>
<td>81.7</td>
<td>82.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HMR [5]</td>
<td>53.2</td>
<td>56.8</td>
<td>50.4</td>
<td>62.4</td>
<td>54.0</td>
<td>72.9</td>
<td>49.4</td>
<td>51.4</td>
<td>73.7</td>
<td>54.4</td>
<td>50.0</td>
<td>62.6</td>
<td>47.1</td>
<td>55.0</td>
<td>57.2</td>
<td></td>
</tr>
<tr>
<td>CMR [40]</td>
<td>41.8</td>
<td>44.8</td>
<td>42.6</td>
<td>46.6</td>
<td>45.9</td>
<td>57.2</td>
<td>40.8</td>
<td>40.6</td>
<td>52.2</td>
<td>66.0</td>
<td>46.6</td>
<td>42.8</td>
<td>51.7</td>
<td>44.6</td>
<td>48.2</td>
<td></td>
</tr>
<tr>
<td>SPIN [43]</td>
<td>37.6</td>
<td>42.4</td>
<td>38.8</td>
<td>42.6</td>
<td>40.4</td>
<td>45.9</td>
<td>36.1</td>
<td>36.7</td>
<td>48.7</td>
<td>58.6</td>
<td>41.2</td>
<td>37.9</td>
<td>46.6</td>
<td>38.4</td>
<td>41.1</td>
<td></td>
</tr>
<tr>
<td>DaNet-LSTM [11]</td>
<td>43.3</td>
<td>48.8</td>
<td>50.6</td>
<td>48.3</td>
<td>47.3</td>
<td>55.5</td>
<td>41.6</td>
<td>42.7</td>
<td>53.8</td>
<td>61.5</td>
<td>47.4</td>
<td>43.2</td>
<td>53.3</td>
<td>40.8</td>
<td>47.9</td>
<td>48.6</td>
</tr>
<tr>
<td>Ours</td>
<td>37.9</td>
<td>44.3</td>
<td>41.2</td>
<td>43.3</td>
<td>42.1</td>
<td>48.7</td>
<td>36.2</td>
<td>38.9</td>
<td>47.4</td>
<td>53.7</td>
<td>41.1</td>
<td>39.9</td>
<td>46.0</td>
<td>34.6</td>
<td>41.3</td>
<td>42.9</td>
</tr>
<tr>
<td>Ours-6D</td>
<td>35.7</td>
<td>40.4</td>
<td>39.0</td>
<td>40.3</td>
<td>40.5</td>
<td>47.4</td>
<td>35.1</td>
<td>34.9</td>
<td>45.2</td>
<td>51.7</td>
<td>39.6</td>
<td>37.8</td>
<td>43.4</td>
<td>34.4</td>
<td>39.8</td>
<td>40.5</td>
</tr>
</tbody>
</table>

For more comprehensive comparison, Table 2 reports pose estimation performance across different actions on Human3.6M. Compared with SPIN and other methods, our method can be more robust to challenging actions such as Sitting and Sitting Down. We believe these benefits come from our decomposition design which enables our network to capture more detailed information for joint poses and produce more accurate reconstruction results. We can also see from the last row of Table 2 that, by simply replacing rotation matrices with the 6D representations [68] for pose parameters as SPIN do, our method can achieve results on par with or even better than SPIN.

5.3.2 Comparison on In-the-wild Datasets
Reconstructing 3D human model on real-world images is much more challenging due to factors such as extreme poses...
and heavy occlusions. In our network, the aggregated refinement module and PartDrop training strategy are proposed to enhance its robustness and generalization. We conduct evaluation experiments on UP-3D, COCO, and 3DPW to demonstrate the efficacy of our method.

**Evaluation on UP-3D.** For comparison on the UP-3D dataset, we report quantitative results in the PVE of the reconstructed meshes in Table 3. In comparison with previous methods, our method outperforms them across all subsets of UP-3D by a large margin. Our closest competitor BodyNet [34] has the PVE value of 102.5 on LSP, while ours is 88.5. Moreover, BodyNet [34] uses both 2D and 3D estimation as the intermediate representation, which is much more time-consuming than ours. Reconstruction results on UP-3D are visualized in Fig. 8. Compared with other methods, our DaNet could produce more satisfactory results under challenging scenarios.

**Evaluation on COCO.** For evaluation on COCO, we train our model on the mixture of training data from DensePose-COCO and Human3.6M datasets, and perform both qualitative and quantitative comparison on the COCO validation set. We first show qualitative reconstruction results in Fig. 9, and make comparisons with HMR [5], Rong et al. [15], and SPIN [43]. As we can see, our method has better generalization in real-world scenarios with more accurate and well-aligned reconstruction performances. Our method can produce reasonable results even in cases of extreme poses, occlusions, and incomplete human bodies, while competitors fail or produce visually displeasing results.

To perform quantitative evaluations on COCO, we project keypoints from the estimated SMPL models on the image plane, and compute the Average Percision (AP) based...
on the keypoint similarity with the ground truth annotations. We report keypoint localization APs of different approaches in Table 4, where we also include 2D human pose estimation approaches [70], [82], [83] for comparison. It can be seen that, in terms of keypoint localization results, approaches for 3D human mesh recovery lag far behind those for 2D human pose estimation. Among approaches for human mesh recovery, our model achieves significantly higher APs than previous ones. Compared with the recent state-of-the-art method SPIN [43], our model improves the mean AP and AP50 by 16.5% and 29.5%, respectively. We attribute such remarkable improvements to our decompose-and-aggregate design. To validate this, we upgrade the backbone of SPIN to HRNet-W64-C [70], a more powerful classification network, and denote it as SPIN-HRNet. As shown in Table 4, though SPIN-HRNet has a stronger backbone with more parameters than our whole network, it brings much less gains over SPIN (3.9% improvement in mean AP from 17.3% to 21.2%). In contrast, our network decomposes the perception tasks and aggregates them efficiently, making our SMPL regressor more effective to handle challenging cases in real-world scenes.

Table 4 also presents the comparison of our approach against our previous model DaNet-LSTM [11]. DaNet-LSTM has the same network architecture with ours except that its aggregation procedure is performed sequentially along kinetic chains via LSTM. Based on DaNet-LSTM, we introduce the graph-based aggregation module and PartDrop strategy in this work. The graph-based aggregation performs feature refinement in parallel for all body parts, while PartDrop regularizes the network and encourages learning features from complementary body parts. These newly introduced designs can help to improve the robustness and generalization of our model. As shown in Table 4, both new components contribute to higher performance in this challenging dataset. By replacing the LSTM-based aggregation module with the graph-based one, our DaNet-GCN obtains a 6.8% improvement in AP50. By adopting the PartDrop strategy for training, we further have a 3.1% improvement in AP50. Taking these two updates together, our approach improves the AP50 by 9.9% over DaNet-LSTM from 58.7% to 68.6%. We can also see from Table 4 that other dropping out strategies such as Dropout and DropBlock do not work well as PartDrop and even degrade the performance. One intuitive explanation for this is that our PartDrop can better imitate the corrupted IUV maps in challenging cases. As we can observe from Fig. 10 that the body parts are missing irregularly from the estimated IUV maps due to occlusions. PartDrop helps to produce more natural and well-aligned results in comparison with its alternatives.

**Evaluation on 3DPW.** In Table 5, we report the results of our approach and other state-of-the-art approaches on the 3DPW test set. Here, we use the same datasets and training strategy as SPIN [43] and do not use any data from 3DPW for training. Besides, the valid SMPL parameters fitted in SPIN are adopted as ground-truth labels for those in-the-wild training datasets. As shown in Table 5, our approach reduces the MPJPE-PA by 4.4 mm over SPIN to 54.8 mm, achieving the best performance among frame-based and even temporal approaches. Table 5 also includes SPIN-HRNet for comparison, where we can see that there is only a 0.7 mm reduction in MPJPE-PA over SPIN. Fig. 11 depicts the qualitative results of our approach. We can observe that our model has better generalization performances on 3DPW in comparison with SPIN.

### 5.3.3 Running Time

During inference, our method takes about 93ms on a Titan Xp GPU, where the IUV estimation accounts for 60ms while the parameter prediction accounts for the rest 33ms. The running time and platform of different models are included in Table 6 for comparison. Numbers are obtained from respective literature or evaluated using their official implementation. Overall, our method has a moderate computation cost among regression-based reconstruction methods.

**Table 6**

Comparison of running time (ms) with state-of-the-art models.

<table>
<thead>
<tr>
<th>Method</th>
<th>Run Time</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMR [5]</td>
<td>40</td>
<td>GTX 1080 Ti</td>
</tr>
<tr>
<td>Pavlakos et al. [6]</td>
<td>50</td>
<td>Titan X</td>
</tr>
<tr>
<td>NBF [7]</td>
<td>110</td>
<td>Titan Xp</td>
</tr>
<tr>
<td>BodyNet [34]</td>
<td>280</td>
<td>Modern GPU</td>
</tr>
<tr>
<td>CMR [40]</td>
<td>33</td>
<td>RTX 2080 Ti</td>
</tr>
<tr>
<td>DenseRaC [41]</td>
<td>75</td>
<td>Tesla V100</td>
</tr>
<tr>
<td>Ours</td>
<td>93</td>
<td>Titan Xp</td>
</tr>
</tbody>
</table>
Effect of IUV Estimation Quality. We further conduct experiments to investigate the impact of the quality of dense estimation on the final shape and pose prediction performance. To this end, different architectures or initializations of the IUV estimators are adopted in ablation experiments to produce IUV maps with different qualities. Specifically, the IUV estimator adopts the pose estimation networks built upon ResNet-50 and ResNet-101 as alternative architectures, and these models are pretrained on ImageNet or COCO. Following the protocol of DensePose, we measure the quality of dense correspondence estimations via the pointwise evaluation, where the area under the curve at the threshold of 10 cm (i.e., AUC\textsubscript{10}) is adopted as the metric. Fig. 12(a) reports the reconstruction results of ablation approaches versus their qualities of IUV estimations. As we can see, networks with better IUV estimations consistently achieve better reconstruction performance. To investigate the performance upper bound of adopting IUV maps as intermediate representations, we also report the results of the approach using ground truth IUV maps as input with the removal of the IUV estimator. As shown in the rightmost result of Fig. 12(a), the approach learning from the ground truth IUV maps achieves much better performance than using the estimated one outputted from networks, which means that there is still a large margin for improvement by adopting IUV maps as intermediate representations.

In contrast to the concurrent work, our approach augments the annotation of Human3.6M with the rendered IUV maps so that our IUV estimator can be trained on Human3.6M with dense supervision, which enables our network to have a higher quality of IUV estimation. To verify this, the IUV estimator is firstly trained on DensePose-COCO or Human3.6M, and then frozen to generate IUV maps for the training of the reconstruction task on Human3.6M. As can be seen from Fig. 12(b), approaches with the IUV estimators trained on Human3.6M consistently achieve better performances on both IUV estimation and model reconstruction tasks.

5.4.2 Decomposed Perception
The decomposed perception provides fine-grained information for detailed pose estimation. To validate the effectiveness of such a design, we report the performance of the approaches using one-stream and multiple streams in Table 8, where the D-Net denotes the variant of our DaNet without using the aggregated refinement module and PartDrop strategy. Results in PVE-S and PVE-P are also reported in Table 8 for separately studying the efficacy of the decomposed design on the shape and pose predictions. It can be seen that the reconstruction performance metric PVE is actually dominated by the PVE-P metric. Comparison of the first and second rows in Table 8 shows that using multiple streams has barely effects on the shape prediction but brings a significant improvement in the pose prediction (i.e., the PVE-P value drops more than 14%). We also report results to validate the use of different ratios α\textsubscript{k} and the simplification of partial IUV maps. In the 3rd and 4th rows of Table 8, D-Net-ES adopts equal scales with all α\textsubscript{k}s set to 0.5, while D-Net-AP adopts partial IUV maps with all body parts. As can be seen, such modifications degrade the performance, which is due to two facts that (i) the proportions of body parts are different and (ii) the rotational states of different body joints are relatively independent and
Eating
20
Greeting
Sitting
10cm
50
30
40
Posing
50
Waiting
15cm
30
40
40
50
20
Smoking
Directions
Photo
20
(b) Shape
Discussion
10
5cm
(b)
(c)
(b)
(c) Pose
7x25
0cm
18x42
Per-vertex Error (mm)
18x81
PVE-P (mm)
18x88
PVE (mm)
25x56
62
40
75
64
80
66
100
85
68
90
70
140
95
72
51
56 60 62 60 59 63 65 63 62
74.3 36.3 64.0 61.8 48.5
76.8 36.8 65.8 63.4 49.5
76.1 36.6 65.5 63.1 49.8
76.8 36.8 65.8 63.4 49.5
To visualize the reconstruction performance on different body areas, Fig. 14 depicts the average per-vertex error with respect to the surface areas of the human model. As shown in Fig. 14(a) for the baseline network, the per-vertex errors of limb parts (hands, feet) are much higher than that of the torso. By comparing Figs. 14(a) and 14(b), we can conclude that our decomposed perception design alleviates the above issue and achieves much better reconstruction performance on limb parts. Reconstruction performances across different actions on Human3.6M are also reported in Fig. 13 for comprehensive evaluations. We can see that the decomposed perception design reduces reconstruction errors consistently for all actions.

### 5.4.3 Part-based Dropout

The proposed Part-based Dropout (PartDrop) strategy drops IUV values in contiguous regions at the granularity level of body parts. Such a dropping out strategy can effectively regularize the neural network by removing semantic information from foreground areas of intermediate representations. In this subsection, we conduct experiments to validate its effectiveness and evaluate the impact of the dropping rate on the reconstruction performance.

To validate the superiority of our PartDrop strategy, we adopt DropBlock [60] and Dropout [58] as alternative strategies to drop values from intermediate representations during training. For DropBlock, following the setting of [60], the size of the block to be dropped is set to 7 in our experiments. For fair comparison, only the foreground pixels are involved in counting the dropping rate. Fig. 15 reports the performance of the full model reconstruction as well as its shape and pose components under different strategies across different dropping rates. It can be seen that the performance gains brought by dropping out strategies mainly come from the pose prediction tasks since the evaluation metric PVE is dominated by its pose component PVE-P. Among three strategies, Dropout is the worst and its performance deteriorates quickly when increasing the rate of dropping out. DropBlock works better than Dropout and brings marginal gains when the dropping rate is less than 20%. Though we can see from the PVE-S curves in Fig. 15(b) that DropBlock has comparable results with PartDrop on shape prediction when the dropping rate is larger than 40%, its pose prediction results degrade significantly as shown in Fig. 15(c). We hypothesize that the removal of a large area of block makes DropBlock similar to PartDrop for the global perception but does harm to the local perception for pose prediction. Compared with these two alternatives, the proposed PartDrop is more robust to the dropping rate and achieves the best results at a dropping rate around 30%. The above comparison of unit-wise, block-wise, and part-wise dropping strategies suggest that removing features in a structured manner is crucial to our reconstruction task, where PartDrop performs best among them. The efficacy of
To validate their functions, we discard position supervisions and convert position features back to the rotation feature space.

The only difference between the direct and position-implicit refinement strategies is that, in the latter one, there are two mapping operations performed before and after the refinement. We report the results of the approaches using direct, position-implicit, and position-aided strategies in Table 9 for comparison. It can be seen that the position-implicit strategy achieves inferior results than the position-aided strategy but better results than the direct strategy, which means that the implicit position space still works better than the rotation space for feature refinement. Example results of the approach using the direct or position-aided refinement strategy are also depicted in Fig. 16 for comparison. We can see that the position-aided refinement helps to handle challenging cases and produce more realistic and well-aligned results, while the direct refinement brings marginal to no improvement.

The reason behind the inferior performance of the direct refinement is that the correlation between rotation features is weak, and the messages of neighboring rotation features are generally irrelevant to refine the target rotation feature. Our refinement module builds an auxiliary position feature space for feature refinement, making it much more efficient than that in the original rotation feature space. To verify this, we extract the features before refinement from the rotation, implicit position, and position spaces of the three strategies mentioned above, and compute the correlations between features of different body joints. Fig. 17 shows the comparison of correlation matrices of these three types of features. As observed from Fig. 17(a), the correlation matrix of rotation features approximates to an identity matrix, meaning that the correlations between the rotation features of different joints are rather weak even for two adjacent joints. By contrast, for implicit position features in Fig. 17(b) and position features in Fig. 17(c), the correlations between features of adjacent joints are much higher, making it more feasible to refine features with the messages from neighboring joints.

5.4.4 Aggregated Refinement

Our aggregated refinement module is proposed to impose spacial structure constraints upon rotation-based pose features. As observed from Fig. 14(d) and Fig. 13, the aggregation in DaNet effectively reduces the reconstruction errors across all surface areas and human actions considerably.

A straightforward strategy to refine the feature would be conducting refinement between the rotation features directly. In such a direct refinement strategy, the first and third steps of our refinement procedure are removed and the rotation features are directly refined by the graph convolution layers of the second step. The features outputted from the last refinement layer are also added with the original rotation features in a residual manner and then used to predict joint rotations. For fair comparison, the refinement layer number of the direct strategy is equal to the number of the layers involved in the three steps of the position-aided strategy.

Rotation Feature Space vs. Position Feature Space. The proposed position-aided refinement strategy performs refinement in the position feature space instead of the rotation feature space. The graphs $A^{r2p}$ and $A^{r2r}$ of the first and last refinement steps are customized to connect the rotation and position feature spaces. The graph $A^{r2p}$ collects rotation features to the position feature space, while the graph $A^{r2r}$ converts position features back to the rotation feature space. To validate their functions, we discard position supervisions from the objective $L_{\text{refine}}$ during refinement. We refer to this strategy as the position-implicit refinement strategy since the position feature space is built in an implicit manner. The only difference between the direct and position-implicit refinement strategies is that, in the latter one, there are two mapping operations performed before and after the refinement. We report the results of the approaches using direct, position-implicit, and position-aided strategies in Table 9 for comparison. It can be seen that the position-implicit strategy achieves inferior results than the position-aided strategy but better results than the direct strategy, which means that the implicit position space still works better than the rotation space for feature refinement. Example results of the approach using the direct or position-aided refinement strategy are also depicted in Fig. 16 for comparison. We can see that the position-aided refinement helps to handle challenging cases and produce more realistic and well-aligned results, while the direct refinement brings marginal to no improvement.

Benefit from Learnable Graph Edge and PartDrop. The learnable edge weighting matrix $M$ of the refinement graph contributes to better balancing the importance of neighboring messages, while the PartDrop strategy helps to encourage the network to leverage more information from neighboring joints. To verify their effectiveness during feature refinement, Table 10 reports the results of the ablation approaches incrementally adopting the learnable edge in the refinement graph and the PartDrop strategy,
Ablation study of using learnable graph edge and PartDrop strategies on the Human3.6M dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>PVE</th>
<th>MPJPE</th>
<th>MPJPE-PA</th>
</tr>
</thead>
<tbody>
<tr>
<td>D-Net + PartDrop</td>
<td>74.3</td>
<td>61.8</td>
<td>48.5</td>
</tr>
<tr>
<td>D-Net+Direct</td>
<td>72.1</td>
<td>59.4</td>
<td>46.9</td>
</tr>
<tr>
<td>+ LearntEdge</td>
<td>72.7</td>
<td>59.6</td>
<td>47.0</td>
</tr>
<tr>
<td>+ PartDrop</td>
<td>70.3</td>
<td>58.1</td>
<td>45.5</td>
</tr>
<tr>
<td>D-Net+Pos.-aided</td>
<td>70.8</td>
<td>57.1</td>
<td>45.9</td>
</tr>
<tr>
<td>+ LearntEdge</td>
<td>68.9</td>
<td>55.8</td>
<td>44.9</td>
</tr>
<tr>
<td>+ PartDrop</td>
<td>66.5</td>
<td>54.6</td>
<td>42.9</td>
</tr>
</tbody>
</table>

Fig. 18. Visualization of learned edge weighting matrices under different training settings. (a),(b) Direct refinement without and with PartDrop. (c),(d) Position-aided refinement without and with PartDrop.

where D-Net+Direct and D-Net+Pos.-aided adopt the refinement module with the direct and position-aided strategy, respectively. It can be seen that, for the direct refinement, the performance gains mainly come from the PartDrop strategy. In contrast, for the position-aided refinement, the performance gains are attributed to both the learnable edge and the PartDrop strategy. Fig. 18 depicts the learned edge weighting matrices of different ablation approaches. As observed, the learned edge weighting matrices of the direct refinement are relatively flat with lower values. When using the PartDrop strategy, the learnable values of most edges in the refinement graph rise for the position-aided refinement, while such a phenomenon is not observed for the direct refinement. We conjecture that the PartDrop strategy brings gains from two perspectives. First, PartDrop regularizes the backbone feature extractor to focus on more complementary regions in intermediate representations for better feature exploitation. Second, PartDrop encourages the refinement module to borrow more information from neighbors in the position feature space for better feature refinement.

### 6 Conclusion

In this work, a Decompose-and-aggregate Network is proposed to learn 3D human shape and pose from dense correspondences of body parts with the decomposed perception, aggregated refinement, and part-based dropout strategies. All these new designs contribute to better part-based learning and effectively improve the reconstruction performance by providing well-suited part perception, leveraging spatial relationships for part pose refinement, and encouraging the exploitation of complementary body parts. Extensive experiments have been conducted to validate the efficacy of key components in our method. In comparison with previous ones, our network can produce more accurate results, while being robust to extreme poses, heavy occlusions, and incomplete human bodies, etc. In future work, we may explore integrating dense refinement \cite{14} to further improve the shape and pose recovery results.

###Acknowledgments

The authors would like to thank the associate editor and reviewers for their helpful comments to improve this manuscript. This work was supported in part by the National Natural Science Foundation of China (Grant No. U1836217, 61806197) and the National Key Research and Development Program of China (Grant No. 2017YFC0821602). This work was done when H. Zhang visited the University of Sydney with the support of the Joint Ph.D. Training Program of the University of Chinese Academy of Sciences.

###References

\begin{itemize}
\end{itemize}


