Chasing the Tail in Monocular 3D Human Reconstruction with Prototype Memory

Yu Rong, Ziwei Liu and Chen Change Loy

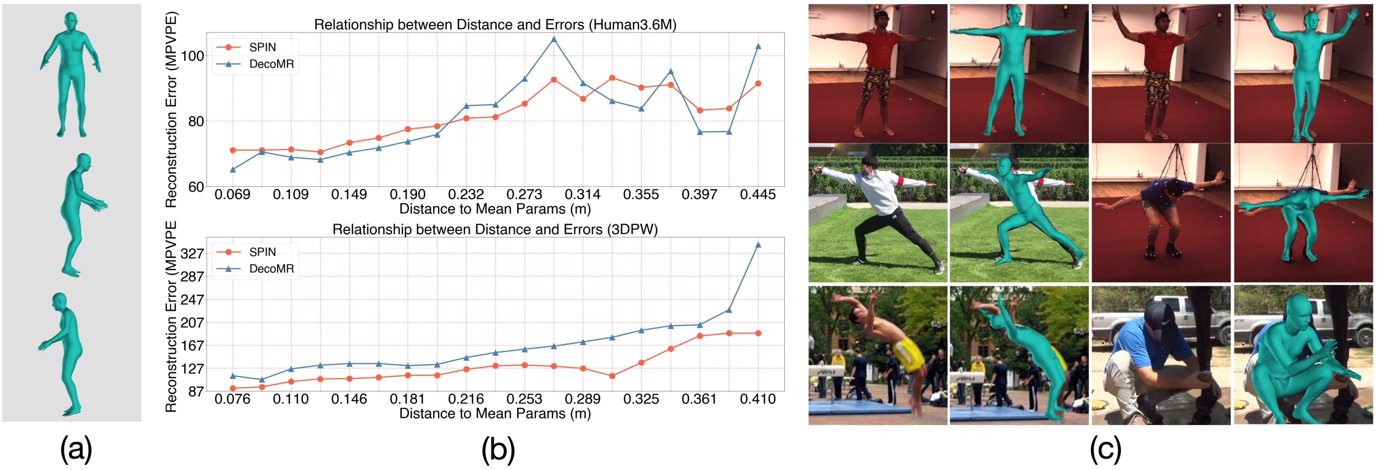


Fig. 1: **Overview**. In (a), we show one typical singular human prototype used by previous works [1]. In (b), we plot the relationship between the samples' distances to the mean parameters and models' reconstruction error (mean per-vertex position error) on the two widely used evaluation sets, *i.e.* the evaluation sets of Human3.6M [2] and 3DPW [3]. We select two state-of-the-art 3D human reconstruction models, namely SPIN [1] and DecoMR [4]. In (c), we show several typical examples with gradually increasing distances to the human prototype depicted in (a). As the figure shows, the model (SPIN [1]) generates less precise results on samples with larger distances.

***Abstract*—Deep neural networks have achieved remarkable progress in single-image 3D human reconstruction. However, existing methods still fall short in predicting rare poses. The reason is that most of the current models perform regression based on a single human prototype, which is similar to common poses while far from the rare poses. In this work, we 1) identify and analyze this learning obstacle and 2) propose a prototype memory-augmented network, PM-Net, that effectively improves performances of predicting rare poses. The core of our framework is a memory module that learns and stores a set of 3D human prototypes capturing local distributions for either common poses or rare poses. With this formulation, the regression starts from a better initialization, which is relatively easier to converge. Extensive experiments on several widely employed datasets demonstrate the proposed framework's effectiveness compared to other state-of-the-art methods. Notably, our approach significantly improves the models' performances on rare poses while generating comparable results on other samples.**

***Index Terms*—Motion Capture, 3D Pose Estimation, Clustering**

I. Introduction

Recovering 3D human models from single-view monocular images facilitates numerous applications in augmented reality and creative entertainment. Most existing methods [1], [4], [5], [6], [7] employ a parametric 3D human shape model known as Skinned Multi-Person Linear Model (SMPL) [8] to represent 3D humans and use deep neural networks (DNN) to estimate its parameters. They also employ 3D human prototypes, such as mesh template [4], [7] or mean parameters [1], [5], as the foundation to predict 3D human models. Using these methods, models produce satisfactory results for images with common poses. However, performances of these models decline drastically when applied to images with rare poses or uncommon views, as shown in the third row of Fig. 1 (c).

Performance degradation in predicting rare poses is primarily caused by the usage of a single 3D human prototype for all the samples. The human prototype calculated from these datasets, as visualized in Fig. 1 (a), is more similar to common poses with usual viewpoints, such as standing. Models trained with this single prototype inherently bias towards the common poses while performing less well on rare poses. To illustrate this phenomenon, we show the visualization of the single human prototype used by SPIN [1] in Fig 1 (a) and calculate the Euclidean distance of each sample to the human prototype. We use the distances between the corresponding SMPL vertices. As depicted in Fig. 1 (b), when a single prototype is adopted, two state-of-the-art models, namely SPIN [1] and DecoMR [4], perform well on samples close to the human prototype. However, their reconstruction errors rise rapidly on samples that have a larger distance to the mean parameters.

Take SPIN [1] and Human3.6M [2] for instance, the reconstruction errors of samples with distances to mean parameters of around 4.3 is 20% larger than those with distances to mean parameters of around 2.0. The performance drop is more severe on challenging in-the-wild datasets such as 3DPW [3]. We further show several typical examples in Fig. 1 (c). It is observed that current state-of-the-art models' performances start to corrupt when the samples' poses gradually differ from the human prototype. For better clarification, in the following of the paper, we call the samples that are close to the human prototype as head classes and the samples far from the human prototype as tail classes. Head classes are typically composed of common poses, while rare poses majorly lie in tail classes.

To ameliorate the models' performance collapse on tail classes, we propose a prototypical memory network, PM-Net. The core of PM-Net is a memory module that stores multiple 3D human prototypes, each of which covers a non-overlapping subset of the data. Instead of using one single prototype for all data samples, we assign each data to the closest prototype in the memory. In this way, the distance of data to its assigned prototype is significantly reduced. The regression process starts from a much better initialization. Furthermore, the regression process is transformed to be performed on compact local distributions instead of on the global yet sparse distribution. During inference, a classifier is employed to assign input data to the corresponding prototype, which serves as the basis for the follow-up SMPL parameter regression.

Previous works such as LCR-Net [9] also adopt multiple human prototypes by applying standard K-Means on 3D pose data. However, it is not feasible to directly apply the same method in 3D human reconstruction. Firstly, although the distances between 3D poses can be easily measured by Euclidean distance between coordinates, it is not reasonable to use the same metric on SMPL parameters. Instead, Euclidean distances between vertices are more reasonable. Furthermore, the weights of each vertex should be elaborately selected in calculating the distance. For example, the number of vertices that belong to the head is nearly the same as the number of vertices of limbs, while the latter has much more influence on the overall poses and thus should be placed larger weights in distance calculation. Another issue is how to obtain the centers of each cluster after clustering. A direct average of vertices is obviously infeasible since the resulting vertices are highly likely to not lie in the valid human body topology. Therefore, we choose to conduct on the samples' SMPL parameters and apply separate averaging strategies on pose and shape parameters to better suit their properties.

In summary, we make the following contributions: We identify the relationship between monocular 3D human reconstruction models' performances and samples' distances to the applied singular human prototype. To improve the models' performances on tail classes, we design a prototype memory module to fully leverage the information contained across all samples with both common and rare poses. Thanks to the unique formulation of local prototypes, our method improves the models' performances on both tail classes and the overall samples. Our model achieves state-of-the-art performances on Human3.6M [2], MPI-INF-3DHP [10], 3DPW [3], and UP-3D [11]. In particular, the model's average reconstruction errors on tail classes are reduced by 12 mm on challenging in-the-wild dataset, *i.e.* UP-3D [11]. The proposed method is easy to implement and can be adapted to various frameworks or other visual tasks that perform regression from mean parameters.

II. Related Work

3D Body Pose Estimation. In recent years, researchers adopt deep neural networks to resolve the challenging task of 3D body pose estimation. Pioneers [12], [13], [14] use convolutional neural networks to directly regress 3D body joint coordinates. More recent works either use 3D heatmaps [15], [16], [17], [18], [19] to pursure better pixel location or leverage 2D poses to serve as the auxiliary [20], [21], [22], [23], [24]. Recently, there is a surge of 3D pose estimation works trying to produce 3D poses lifted from input 2D poses [25], [26], [27], [28], [29], [30], [31], [32], [33], [31]. 3D pose estimation is closely related to 3D human pose and shape recovery. Due to the lack of direct supervision, most of the 3D human reconstruction works [5], [34], [35], [36], [37] leverage 3D joint positions in model training. There are also works that directly leverage 3D poses during inference. Choi *et al* [38] estimate 2D and 3D poses as intermediate representation and then obtain the final 3D meshes. HybrIK [39] uses estimated 3D joint locations to directly calculate the global orientations and part of the joint rotations for 3D human reconstruction.

Single-Image 3D Human Reconstruction. Most works of 3D human pose and shape recovery use a parametric model SMPL [8] to represent 3D humans. Although there are works that cope with sequential input [40], [41], [42], [43], [44], [45], multi-view images [46], [47], [48], [49], [50], or point clouds [51], [52], we majorly discuss works that estimate 3D humans from monocular single images. Recent methods for single-image 3D human reconstruction share a similar pipeline. In particular, DNN or optimization is employed to obtain the parameters or vertex coordinates of the SMPL model. Most learning-based methods [5], [53], [54], [34], [35], [55], [56], [36], [37], [57], [58], [59], [60], [61] use CNN-based models to predict the parameters of SMPL directly. Kanazawa *et al* [5] use adversarial losses [62] to judge the predicted 3D human poses. Pavlakos *et al* [53] propose to predict the 2D poses heatmaps and silhouette as the intermediate representation to facilitate the model prediction. A series of works [6], [54], [34], [63], [64], [65] incorporate DensePose [66] into the framework. Georgakis *et al* [55] adopt a hierarchical strategy to predict the SMPL poses part by part, instead of regressing parameters altogether. PARE [67] adopts part attentions to replace the previously used global features. Other researchers exploit the self-contact [57] or cloth-semantics [68] to increase the prediction accuracy. Other learning-based methods adopt different frameworks. Kolotouros *et al* [7] and Choi *et al* [38] use graph convolution network [69] to predict SMPL vertices directly while Lin *et al* [70] uses transformer architecture [71]. Moon *et al* [35] uses 1D-hetmaps to encode SMPL vertices. Zeng *et al* [4] and Zhang *et al* [72] regress UV maps instead of the original SMPL parameters or vertices.

Optimization-based methods optimize the SMPL parameters instead of using DNNs. Bogo *et al* [73] obtain SMPL parameters by minimizing the distances between ground-truth 2D

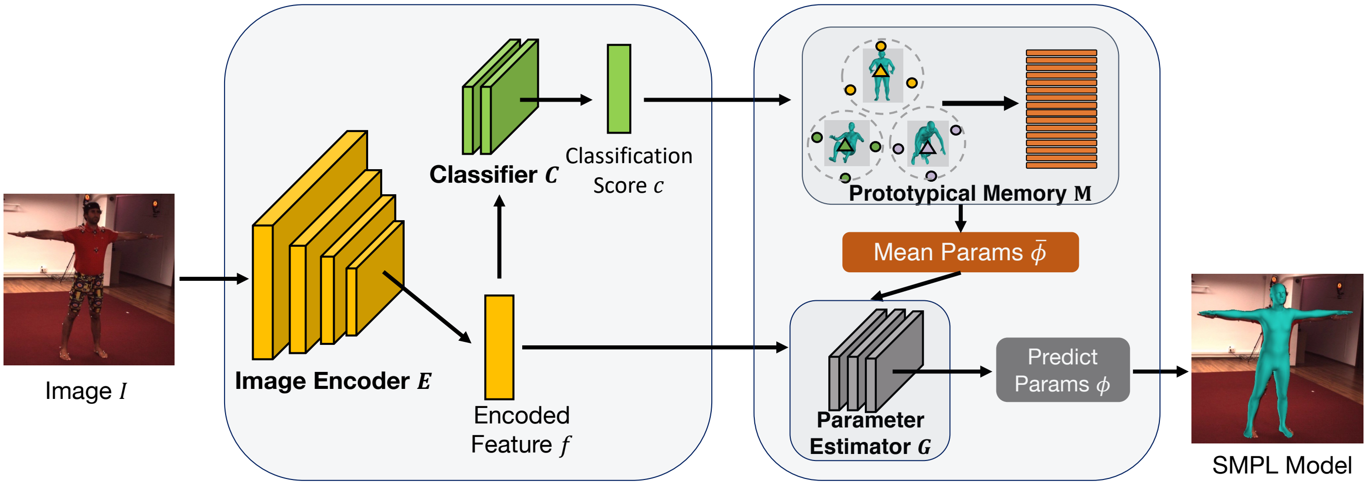


Fig. 2: **Overall framework of PM-Net**. The framework is composed of an image encoder E, a classifier C, a parameter estimator G, and a prototypical memory M. The most important component that distinguishes our work from other methods is the Prototypical Memory M that stores multiple sets of mean parameters. Using the classification score c generated by the classifier, the matching mean parameters $\bar{\phi}$ are then selected from the memory M. Parameter Estimator G then uses the matching mean parameters as the initialization and regresses the SMPL parameters ϕ after several iterations. In the end, SMPL model generates the corresponding vertices and joints of the predicted parameters.

keypoints and projected 2D keypoints from predicted SMPL models. Kolotouros *et al* [1] design a framework to unify the learning-based and optimization-based methods. They first deploy a CNN model to predict the SMPL parameters of the given images. The predicted SMPL parameters then serve as the initialization of the optimization fitting [73]. After that, the optimized SMPL parameters are used to supervise the CNN model. Those steps iterate until convergence. Our model shares a similar framework as SPIN [1] while we use the fitted SMPL parameters to supervise the model instead of performing "in-the-loop" optimization in the training process. Another major difference is that we apply prototypical memory and assign each sample the closest prototype learned from data.

Prototypes used in 3D Human Reconstruction. HMR [5]-based methods [1], [54], [34], [37], [60], [68], [57], [60], [36] use mean parameters calculated from the datasets to represent the 3D human prototype. Güler *et al* [66] use a set of Euler angles to form a convex hull as the basis for SMPL pose prediction. Kolotouros *et al* [7] and Lin *et al* [70] use mesh template as the input to directly regress 3D vertices of the SMPL model. DecoMR [4] also adopts the reference mesh as the base for predicting location maps. Although these works achieve promising results on common poses, their performances drop drastically when applied to images with rare poses. It is because such methods only adopt one single 3D human prototype close to common poses while far from rare poses. Therefore, these methods are biased towards predicting common poses. To overcome this drawback, we propose to use multiple 3D human prototypes. Each data sample is assigned to the closest learned prototype. To our knowledge, LCR-Net [9] is the only work that applies multiple prototypes in the 3D human estimation area. The prototypes used by LCR-Net are obtained from obtaining Naive K-Means to 3D pose data. Nevertheless, obtaining effective prototypes for 3D human reconstruction is not that trivial. Directly applying Naive K-Means on SMPL parameters, *i.e.*, pose parameters and shape parameters, will lead to suboptimal performance. To circumvent this hurdle, we elaborately design a specified clustering algorithm for SMPL model, which is more effective in 3D human reconstruction.

TABLE I: List of mathematical symbols.

Meaning	Math Symbol
SMPL Pose Parameters	θ
SMPL Shape Parameters	β
SMPL Parameters	$\phi = (\theta, \beta)$
SMPL Vertices	V
SMPL Joint Regressor	J
3D Joints	J^{3D}
2D Joints	J^{2D}
Keypoints Visibility	μ
Cluster Center	π
Weights for each Vertex	W
Number of Clusters	K
Input Image	I
Image Encoder	E
Classifier	C
Parameter Estimator	G
Prototypical Memory	M
Encoded Feature	f
Classification Score	c
Mean Parameters	$\bar{\phi}$
Predict Parameters	phi

III. Methodology

In this section, we will first introduce the 3D human model and the inference process of PM-Net. Next, we present the design of prototypical memory. Finally, we discuss the training strategy used in this work. The mathematical symbols used in this paper are listed in Tab. I.

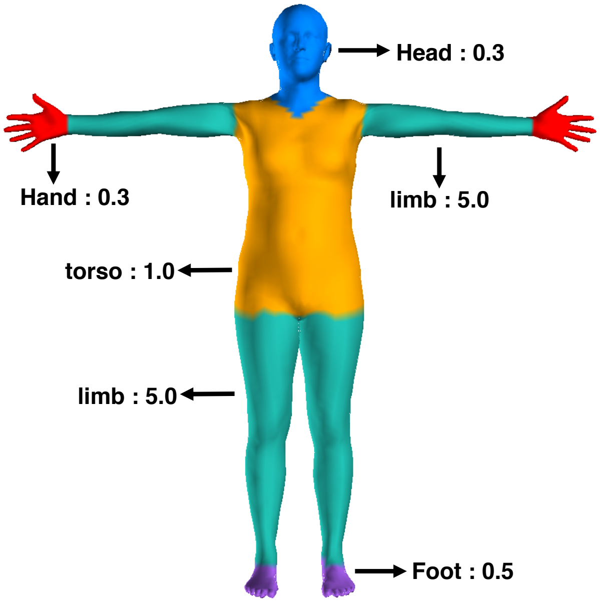


Fig. 3: **Visualization of vertices belonging to different body parts.** In performing part-aware weighting, we divide the whole body into different body parts and assign different weights for each part. The body parts include head, hand, foot, limbs (arms and legs), and torso.

A. *3D Human Model*

We use SMPL [8], a triangle-mesh-based model parameterized by the shape and pose parameters to represent a 3D human model. The shape parameters $\beta \in \mathbb{R}^{10}$ influence the overall body shape. The pose parameters θ model both the relative 3D rotations for a predefined kinematics skeleton with 23 joints and the global rotation for the whole body. The original SMPL pose parameters are in axis-angle representation. Follow the practice of SPIN [1], we use the continuous rotation representations proposed by Zhou *et al* [74] to represent 3D rotations. Therefore, the dimension of pose parameters becomes $\theta \in \mathbb{R}^{6\times24}$. Given the shape and pose parameters, SMPL model calculates the coordinates of vertices $V \in \mathbb{R}^{3\times6890}$. Given SMPL vertices V, 3D joints $J^{3D} \in \mathbb{R}^{3\times24}$ are obtained by using predefined joint regressor $J \in \mathbb{R}^{24\times6890}$ by $J^{3D} = V \cdot J^T$.

B. *Framework*

The overall framework is shown in Fig. 2. A single image I serves as the only input. The image encoder E takes in input I and outputs encoded feature f. The encoded feature is then fed into the classifier C, which then produces the classification score c. The score c is then used to selected the matching mean parameters $\bar{\phi} = (\bar{\theta}, \bar{\beta})$ from the prototypical memory M. Taking the encoded feature f and selected mean parameters $\bar{\phi}$ as the input, the parameter estimator G regresses the SMPL parameters ϕ. In the last step, 3D meshes are generated by the SMPL model using the estimated parameters.

The image encoder E is a ResNet-50 [75]. The classifier C is a one-layer MLP with dimension K, which is the number of prototypes in the prototypical memory M. The parameter estimator G is a three-layer MLP. The dimensions of each layer are 1024, 1024, and 157, respectively. The 157 dimensions are composed of shape parameters (10), pose parameters (6×24) and camera parameters (3). Following the practice of previous works [1], [5], the final parameter ϕ is obtained through iterative error feedback (IEF). To be specific, the estimator G takes the concatenation of encoded feature $f \in \mathbb{R}^{1\times1024}$ and current estimated parameter $\phi_t \in \mathbb{R}^{1\times157}$ (ϕ_t is initialized as mean parameters $\bar{\phi}$ in the first loop $\phi_0 = \bar{\phi}$) as input and outputs the parameter residual $\Delta\phi_t$. The estimated parameter is then updated by adding the current parameter and the residual $\phi_{t+1} = \phi_t + \Delta\phi_t$. The whole loop iterates for 3 times.

Contrary to the previous practices [1], [5] that use the same mean parameters for all samples, our model selects the best matching mean parameters for each sample. The framework maintains a prototypical memory $M \in \mathbb{R}^{K\times157}$ composed of multiple sets of mean SMPL parameters (camera parameters in each prototype are identical). Given each sample, the matching set of parameters is selected. In data preprocessing, each data sample is assigned a one-hot class label $\hat{c} \in \mathbb{R}^{1\times K}$, indicating to which set of mean parameters the sample is closest to. During inference, classification score $c \in \mathbb{R}^{1\times K}$ produced by the classifier C for each sample is used to select mean parameters $\bar{\phi}$ from the prototype memory M as $\bar{\phi} = cM$.

C. *Building Prototypical Memory*

To construct the prototypical memory M, we first apply clustering on the training data. The obtained cluster centers are used to compose the memory. Instead of applying Naive K-Means on SMPL parameters $\phi = (\theta, \beta)$, we elaborately design a clustering algorithm that exploits the characteristics of 3D human reconstruction. The algorithm is based on K-Means with the following modifications:

1) *Distance calculating* – Instead of calculating the distances between parameters ϕ, we use the SMPL model to obtain the vertices V of the corresponding parameters ϕ. Then we calculate the Euclidean distance based on the vertex coordinates. We argue that this distance format is more suitable since it directly and effectively reflects the pose and shape variations across different samples. Furthermore, in calculating the distances based on vertices, we find it is suboptimal to assign the same weights to all the vertices. For example, the number of vertices belonging to the head is similar to the number of vertices of limbs, while the limbs have more influence on the overall pose estimation. Therefore, we assign larger weights to body parts that have more influence on pose estimation while smaller weights to the other parts. We call this strategy as *part-aware weighting*. In our experiment, the weights for vertices belonging to different body parts are empirically set as 5.0 for limbs (arms and legs), 0.3 for head and hand, 0.5 for foot, and 1.0 for torso. Visualization of each body part is shown in Fig. 3.

2) *Cluster center updating* – To update the centers of each cluster after each iteration, a straightforward idea is to directly average pose and shape parameters of the assigned samples. Averaging shape parameters is valid since the process of blending shapes is linear. Nevertheless, directly averaging pose parameters, which are in the form of 3D rotations, is not

Algorithm 1 Part-Aware 3D Human K-Means

```
Require: Threshold for average sample-to-center distance $\hat{\gamma}$;
Require: Threshold for total number of iterations $\hat{\lambda}$.
Require: N samples: $\Phi = \{\phi_1, ...\phi_N\}$; $\phi_i \in \mathbb{R}^{154}$
Require: Initial clusters centers: $\Pi = \{\pi_1, ..., \pi_K\}$; $\pi_j \in \mathbb{R}^{154}$,
Require: Weights for each vertex $W \in \mathbb{R}^{3\times6890}$.
procedure CLUSTERING($\hat{\gamma}$, $\hat{\lambda}$, $\Phi$, $\Pi$, $W$)
    Initialize sets of assigned samples $\Phi_i = \{\}$, $i = 1, \dots, K$
    Initialize set of sample-to-center distances. $\Gamma = \{\}$.
    Initialize average sample-to-center-distance $\bar{\gamma} = \infty$;
    Initialize the number of iterations $\lambda = 0$.
    while $\bar{\gamma} < \hat{\gamma}$ or $\lambda < \hat{\lambda}$ do

        for $i \leftarrow 1, N$ do                          ▷ Assign samples.
            for $j \leftarrow 1, K$ do
                $V_i = smpl(\phi_i)$; $V_j = smpl(\pi_j)$;
                $\gamma_{ij} = \|(V_i - V_j) \circ W\|_2^2$
            end for
            $a_i = \text{argmin}_j(\gamma_{ij})$.
            $\gamma_i = \text{argmin}_{\gamma_{ij}}(\gamma_{ij})$.
            Add sample $\phi_i$ to $a_i$-th cluster.
            Add distance $\gamma_i$ to $\Gamma$.
        end for

        for $j \leftarrow 1, K$ do                 ▷ Update cluster centers.
            $\Theta_j = \{\}$, $B_j = \{\}$
            for $\phi_{jk}$ in $\Phi_j$ do
                $(\theta_{jk}, \beta_{jk}) = \phi_{jk}$.
                Add $\theta_{jk}$ to $\Theta_j$.
                Add $\beta_{jk}$ to $B_j$.
            end for
            $\theta_j = average\_\theta(\Theta_j)$.
            $\beta_j = average\_\beta(B_j)$.
            $\pi_j = (\theta_j, \beta_j)$.
        end for

        $\bar{\gamma} = average(\Gamma)$.          ▷ Update average distances.
        $\lambda = \lambda + 1$.
    end while
end procedure
```

reasonable, as pointed out by previous works [77]. To obtain the valid averaged rotations, we first transfer the 3D rotations to the format of quaternions. Then we apply the quaternion averaging algorithm proposed by Markley *et al* [77].

To be more specific, suppose we have n quaternions q_i to be averaged, the averaged quaternions $\bar{q}$ can be obtained by solve the following equations:

$$\begin{aligned} M &= \sum_{n}^{i=1} q_i q_i^T \\ \bar{q} &= \underset{q \in \mathbb{S}^3}{\text{argmax}}\, q^T M q \end{aligned} \quad (1)$$

$\mathbb{S}^3$ is the 3 dimensional unit sphere. The solution of the maximization problem is the eigenvector of M corresponding to the maximum eigenvalue. We refer the readers to the original paper [77] for more details.

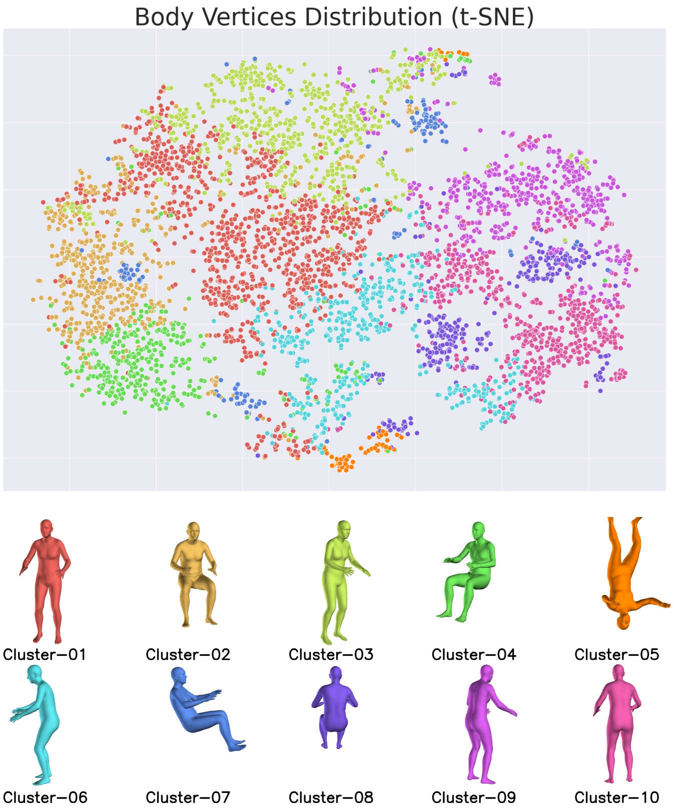


Fig. 4: **Visualization of the clusters.** We use t-SNE [76] to visualize the distribution of clusters obtained from P3DH K-Means. The samples are randomly selected from the training data. The number of clusters is set to be 10 for better visualization.

Equipped with the aforementioned characteristics, we design a new clustering algorithm for 3D human reconstruction called Part-Aware 3D Human K-Means (P3DH K-Means). The whole process of P3DH K-Means is composed of initialization, sample assignment, and center updating. The number of clusters is fixed to be K. Suppose we use N samples in performing the clustering. Each sample is represented as the concatenation of its pose parameters $\theta_i \in \mathbb{R}^{6\times24}$ and shape parameters $\beta_i \in \mathbb{R}^{10}$. The centers of each cluster π_j are represented in the same way. In the initialization step, we randomly select K samples from the training data and use them as the initial cluster centers. In the cluster assignment step, for each sample, the distances of its SMPL vertices between the SMPL vertices of each cluster center are calculated. The sample is then assigned to the closest cluster based on the distances. In the center updating step, shape parameters are calculated by directly averaging the shape parameters of the assigned samples. The pose parameters of the assigned samples are first converted to quaternions format. Then the quaternion averaging algorithm in Eq. (1) proposed in [77] is used to obtain the averaged quaternions. After that, the averaged quaternions are converted back to the continuous rotation representation [74]. The processes of sample assignment and center updating iterate until the algorithm converges. The whole process of P3DH K-Means is presented in Algorithm 1.

In Fig. 4, we visualize the distribution of clusters via t-SNE [76] and the corresponding cluster centers (prototypes). To be specific, we randomly select around 6000 samples from the training data. The number of clusters is set to be 10 for better visualization. It is observed that samples are clearly separated by the clusters obtained from the proposed P3DH K-Means algorithm. Besides, the obtained cluster centers cover both common and rare poses with different view types.

D. Training Scheme

We employ 3D and 2D losses to train our model. The 3D loss is composed of SMPL pose parameter loss L_θ, SMPL shape parameter loss L_β and 3D joint loss L_{J3D}. These losses are defined as follows:

$$\begin{aligned} L_\theta &= \|\theta - \hat{\theta}\|_2^2, \\ L_\beta &= \|\beta - \hat{\beta}\|_2^2, \\ L_{J3D} &= \|J^{3D} - \hat{J}^{3D}\|_2^2, \end{aligned} \tag{2}$$

where $\hat{\theta}$, $\hat{\beta}$ and $\hat{J}^{3D}$ are ground truth SMPL pose parameters, shape parameters and 3D Joints.

To estimate 2D keypoints, we adopt a weak perspective camera model that is composed of scale s and camera translation $[t_x, t_y]$. The camera parameters $[s, t_x, t_y,]$ are also predicted by the parameter estimator G. Given the camera parameters, 2D keypoints J^{2D} are projected from J^{3D}. Then predicted J^{2D} is used to calculate the 2D keypoint loss L_{J2D} with the ground-truth 2D keypoints $\hat{J}^{2D}$. The whole process is formulated as below,

$$\begin{aligned} J_x^{2D} &= s \times J_x^{3D} + t_x, \\ J_y^{2D} &= s \times J_y^{3D} + t_y, \\ L_{J2D} &= \|\mu \cdot (J^{2D} - \hat{J}^{2D})\|_2^2, \end{aligned} \tag{3}$$

where μ is the visibility indicator.

Since our model requires assigning different prototypes to each sample, we apply a cross-entropy classification loss L_C, defined as:

$$L_C = -\hat{c} \cdot \log(c). \tag{4}$$

Overall Loss Function. The overall loss L is defined as:

$$L = \lambda_1 L_{J3D} + \lambda_2 L_{J2D} + \lambda_3 L_\theta + \lambda_4 L_\beta + \lambda_5 L_C. \tag{5}$$

In experiments, we set $\lambda_1 = \lambda_2 = 5.0$, $\lambda_3 = 1.0$, $\lambda_4 = 1\text{e}{-3}$ and $\lambda_5 = 1.0$. These hyper-parameters are obtained from grid search.

Training schedule. We first use the singular human prototype of SPIN [1] (depicted in Fig. 1 (a)) to train image encoder E and parameter estimator G from scratch. The losses defined in Eq. (2) and Eq. (3) are adopted in this stage. Then we finetune the image encoder E and parameter estimator G together with the classifier C using the prototypical memory M and losses defined in Eq. (5). In both stages, the batch size is set to be 256, and Adam optimizer [78] with a learning rate 1e−4 is employed to train the models.

TABLE II: **Ablation studies on the whole evaluation sets.** We show evaluation results of PM-Net leveraging the variants of prototypical memory. The models are evaluated on the whole evaluation sets of each dataset. The evaluation metric is MPVPE.

Dataset → Methods ↓	Human3.6M [2]	3DPW [3]	UP-3D [11]
SPIN [1]	78.7	117.8	119.3
Naive K-Means	79.1	116.6	115.1
Random Select	76.9	115.9	114.1
3DH K-Means	75.4	115.8	113.5
P3DH K-Means	**73.6**	**114.6**	**112.6**

IV. Experiments

In this section, we first introduce the experiment setup. Then we discuss the design of key features of the prototypical memory. After that, we compare our method with previous state-of-the-art methods. In the end, we provide further analysis on the design of PM-Net and typical failure cases.

A. Experimental Setting

In our experiments, we employ several 3D human datasets: Human3.6M [2], MPI-INF-3DHP [10], 3DPW [3], and UP-3D [11]. To increase the generalization ability of the models, we also adopt several in-the-wild 2D datasets, including LSP [79], MPII [80], and COCO [81]. Human3.6M, MPI-INF-3DHP, MPII, LSP, and COCO are used for training. Our models are evaluated on Human3.6M, MPI-INF-3DHP, 3DPW, and UP-3D. In the following of this subsection, we first discuss the evaluation metrics used in our work. Then we give brief introductions to each dataset.

Evaluation Metrics. We mainly use mean per-vertex position error (MPVPE) as the evaluation metric. We believe MPVPE is more suitable than joint-based metrics such as mean per-joint position error (MPJPE), since the latter only considers joint positions while neglecting the joint rotations and body shapes. For completeness, we also adopt MPJPE and PA-MPJPE (MPJPE after applied Procrustes Analysis [82]). These two metrics are used to compare with the previous works and evaluate on datasets without ground-truth SMPL parameters, such as MPI-INF-3DHP [10]. The units of aforementioned metrics are all millimeter (mm).

Human3.6M. Human3.6M [2] is an indoor dataset with 3D joint annotations. We use Mosh [83] to collect SMPL parameters from 3D Mocap markers. Our models are trained on subject S1, S5, S6, S7, and S8 and evaluated on S9 and S11. Following the practice of previous works [1], [5], we only evaluate on samples of the frontal camera (camera 3). This evaluation strategy is often called "Protocol-2" in the literature.

MPI-INF-3DHP. The dataset [10] is captured with a multi-view camera system under a controlled environment. We use the subject S1 to S8 for training and evaluate our models on the evaluation set. The original dataset only provides 3D joints. In training, we also use the pseudo ground truth SMPL parameters provided by SPIN [1] through multi-view fitting.

3DPW. Images of 3DPW [3] are captured from in-the-wild scenarios. The ground truth SMPL pose parameters are obtained

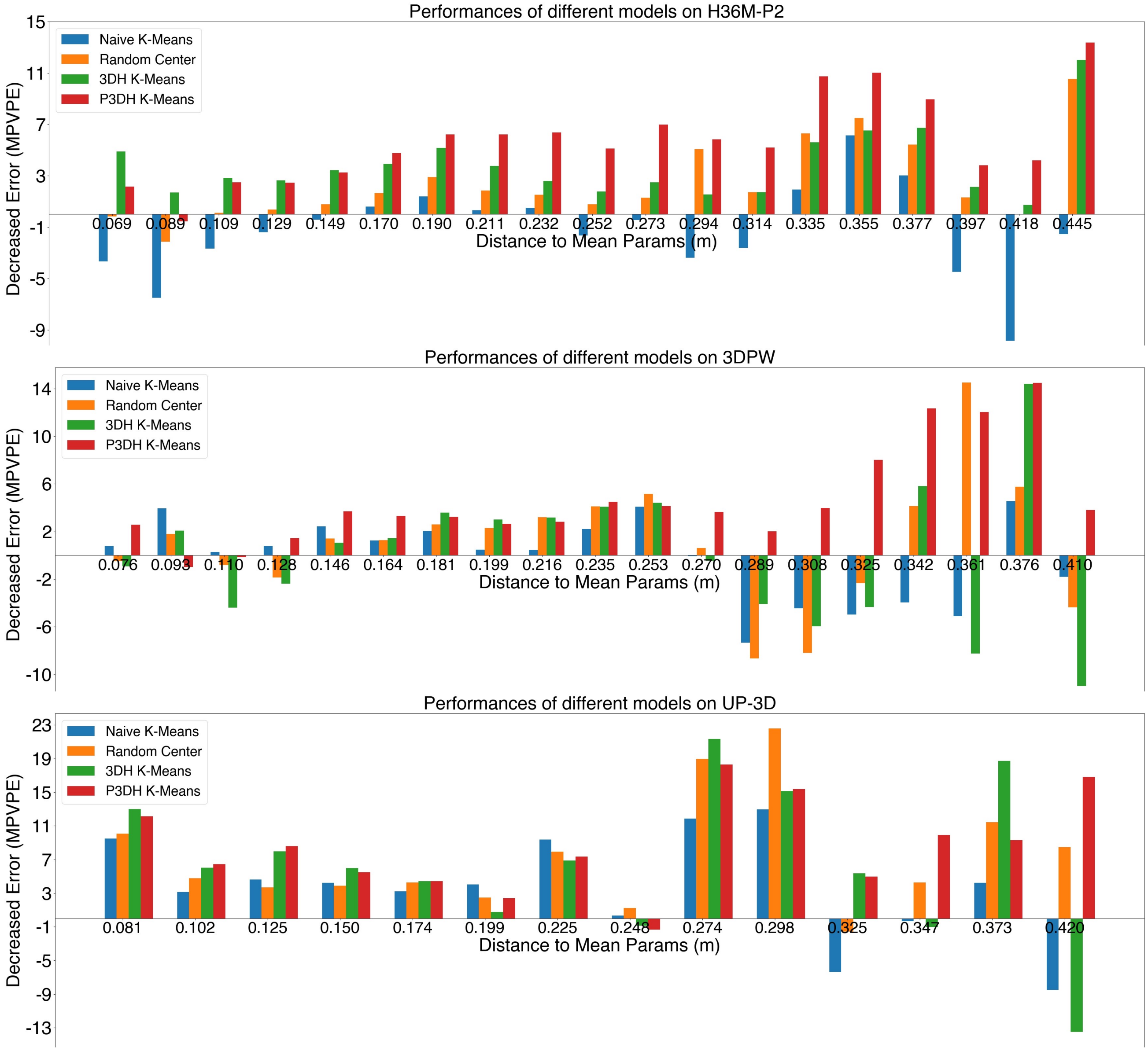


Fig. 5: **Ablation studies on subsets**. We show the performances of the proposed PM-Net using four different clustering methods, namely Naive, K-Means, Random Center, 3DH K-Means, P3DH K-Means. Models are evaluated on Human3.6M [2], 3DPW [3], and UP-3D [11]. Each evaluation set is divided into non-overlapping subsets according to the samples' distances to the singular human prototype visualized in Fig. 1 (a). The x-axis of each sub-figure is the samples' distances to the singular prototype. The y-axis is the models' decreased prediction errors compared to the baseline model, SPIN [1].

from IMUs, while the shape parameters are obtained through 3D scanning. To enable fair comparisons with previous methods, we only use 3DPW's test set for evaluation and do not train our models on it.

UP-3D. UP-3D [11] is composed of selected images from four 2D pose datasets. An extended SMPLify [73] is first used to predict the SMPL parameters on those images. Human annotators are then asked to select the good samples. We only use UP-3D's test set for evaluation and do not train our model on it.

2D Pose Datasets. We leverage several in-the-wild datasets, including LSP [79], MPII [80] and COCO [81]. We use the SMPL fitting results provided by SPIN [1] and originally provided 2D landmark annotations to train our model.

B. Design of Prototypical Memory

We use SPIN [1], which only adopts a single prototype, to serve as the baseline. Apart from reporting results on the overall evaluation set, we also split the evaluation set of Human3.6M [2], 3DPW [3] and UP-3D [11] into unoverlapping subsets, according to samples' distances to the singular prototype depicted in Fig. 1 (a) and report models' reconstruction errors (MPVPE) on each subset.

TABLE III: **Comparison with SOTA on whole evaluation sets.** We compare the proposed PM-Net with previous state-of-the-art methods on several widely used dataset including Human3.6M [2], 3DPW [3], UP-3D [11], and MPI-INF-3DHP [10]. All metrics use millimeters (mm) as the unit and are the lower the better. We use MPVPE, MPJPE, and PA-MPJPE as the evaluation metrics. We mark the methods that use UP-3D in training with $*$.

Dataset & Metrics →	Human3.6M [2]			3DPW [3]			UP-3D [11]	MPI-INF-3DHP [10]	
Methods ↓	MPVPE	MPJPE	PA-MPJPE	MPVPE	MPJPE	PA-MPJPE	MPVPE	MPJPE	PA-MPJPE
Pavlakos *et al* [53]*	-	-	75.9	-	-	-	117.7	-	-
CMR* [7]	87.1	71.9	50.1	144.2	127.6	74.6	96.2	-	-
DecoMR* [4]	78.6	59.6	**39.1**	141.4	126.6	73.7	**93.4**	-	-
HMR [5]	-	-	56.8	-	-	81.3	-	124.2	89.8
DenseRac [54]	-	-	48.0	-	-	-	-	114.2	83.5
SPIN [1]	78.8	62.5	41.1	117.8	98.3	60.2	119.3	105.2	67.5
HKMR [55]	-	60.0	-	-	-	-	-	-	-
I2L-MeshNet [35]	-	**55.7**	41.7	-	95.4	58.6	-	-	-
Pose2Mesh [38]	-	67.9	49.9	-	**91.4**	60.1	-	-	-
Ours	**73.6**	59.6	40.6	**114.6**	94.6	**58.4**	**112.6**	**97.0**	**62.8**

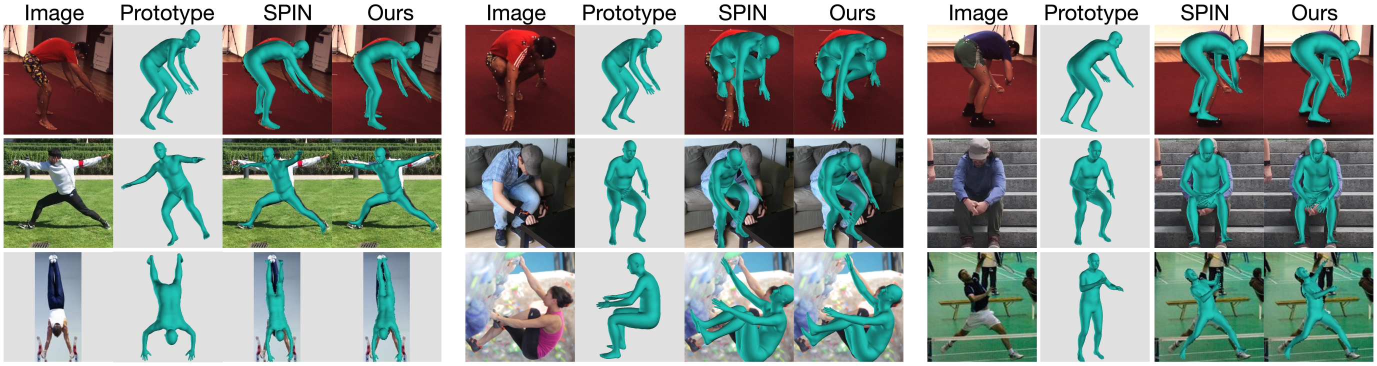


Fig. 6: **Qualitative comparison.** We qualitatively compare the PM-Net with previous state-of-the-art method, SPIN [1].

TABLE IV: **Comparison with SOTA on tail classes.** We compare PM-Net with several state-of-the-art methods on tail classes on Human3.6M [2], 3DPW [3] and, UP-3D [11]. "T-x%" means to only evaluate the samples with the largest x% distances to the singular prototype (*i.e.* tail classes). The evaluation metric is MPVPE. We mark the methods that use UP-3D in training with $*$.

Dataset	Human3.6M [2]		3DPW [3]		UP-3D [11]	
Methods	T-5%	T-10%	T-5%	T-10%	T-10%	T-20%
CMR* [7]	92.5	97.3	209.9	194.3	170.4	149.2
DecoMR* [4]	84.9	88.2	184.3	175.1	**162.0**	**142.5**
SPIN [1]	89.0	88.8	130.0	130.6	184.2	167.9
Ours	**80.0**	**81.1**	**124.9**	**126.4**	172.1	158.6

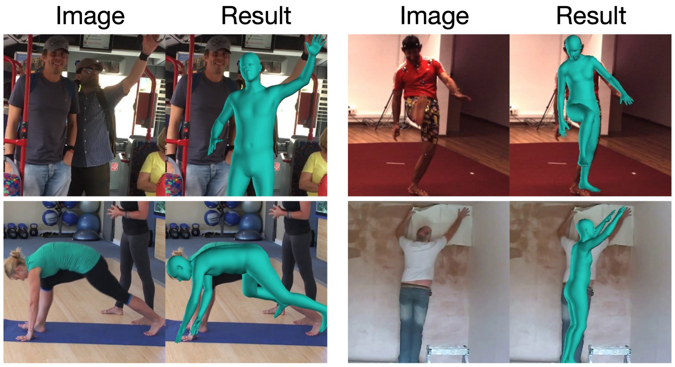


Fig. 7: **Failure cases.** Typical failure cases include occlusion, depth ambiguities, multiple person interaction and challenging backgrounds.

In our experiments, we evaluate four types of prototypical memory, including Naive K-Means, Random Center, 3DH K-Means, and P3DH K-Means. "Naive K-Means" means directly applying the original K-Means algorithm on the SMPL parameters (pose parameters and shape parameters). "Random Center" uses initial centers, which are randomly selected from the samples, as the final cluster centers. "3DH K-Means" refers to the P3DH K-Means without applying part-aware weighting. For all cluster methods, the numbers of prototypes are set to be 50 for Human3.6M [2] and UP-3D [11]. For 3DPW [3], the number of prototypes is set to be 10 for all cluster methods. Part-aware weighting is not applicable for "Naive K-Means", "3DH K-Means", and the process of obtaining cluster centers of "Random Center". Part-aware weighting with limb weight 5 is adopted in the sample assignment process of "Random Center". Experimental results show that these settings can lead to the best performances for all four cluster methods. Please refer to Sec. IV-D for detailed discussions on the influences of number of prototypes and limb weights. These four different models share the same pipeline depicted in Fig. 2. The only differences are how the prototypical memories are built and how samples are assigned to the corresponding prototypes.

The evaluation results are shown in Fig. 5. The x-axis of each sub-figure is the samples' distances to the singular prototype depicted in Fig. 1 (a). The y-axis is the models' error reduction (in terms of MPVPE) compared to the baseline model (SPIN [1]). We also list evaluation results on the whole evaluation sets in Tab. II. It is revealed that P3DH K-Means achieves the best performance on nearly all the subsets of all the three datasets, especially on tail classes that are far from the singular human prototype. 3DH K-Means performs better than Random Center, proving that performing clustering to get better sample assignment is beneficial. Furthermore, it is worth noting that directly performing Naive K-Means on SMPL parameters is not effective. It is not even comparable to the Random Center. The experimental results demonstrate the significance of performing the 3D human-specific clustering.

C. Main Results

Comparison with state-of-the-art methods. In this subsection, we compare our model with previous state-of-the-art methods. For methods with code released, we use their original code and models to obtain the evaluation results that are not reported in the original papers. For other methods, we report the original results listed in their papers. We report evaluation results on both the whole evaluation sets and tail classes only. The results are listed in Tab. III for the whole dataset evaluation and Tab. IV for tail class evaluation, separately. In Tab. IV, "T-x%" means to only evaluate samples with the largest x% distances to the singular prototype (*i.e.* tail classes). The results in Tab. III show that our model achieves comparable performances with previous state-of-the-art methods on the whole evaluation sets. Furthermore, results in Tab. IV show that our model achieves better results on tail classes. These experimental results show that the proposed PM-Net effectively improves the models' performances on challenging tail classes while achieving comparable or better performances on the samples with common poses. We further show qualitative results in Fig. 6 to compare with the baseline model (SPIN [1]). The results show that our model can generate more precise results on challenging samples with rare poses such as squatting on the ground or uncommon views such as head upside down.
Qualitative Results. We show several qualitative results in Fig. 8. It is demonstrated that the proposed PM-Net can effectively select the matched prototypes for the input images and use them as the foundation to generate precise predictions.
Failure Cases. We show several typical failure cases in Fig 7. As the images demonstrate, the model tends to fail when there are severe occlusions, ordinal depth ambiguities, multiple person interactions, and challenging backgrounds.

D. Further Analysis

In this subsection, we evaluate two hyperparameters of PM-Net that influence the models' performances. 1) The number of prototypes. 2) The weight of limbs in performing part-aware weighting. The experiments are conducted on the evaluation set of Human3.6M [2], 3DPW [3] UP-3D [11], and MPI-INF-3DHP [10]. We evaluate the models' performances on the whole evaluation sets and the challenging subsets with rare poses. We evaluate MPI-INF-3DHP using MPJPE and adopt MPVPE for other datasets. The models reported in this subsection all leverage P3DH K-Means to build the prototypical memory. The evaluation results are listed in Tab. V.
Influence of the Number of Prototypes. We first study the influence of the number of prototypes (denoted as K). In our experiments, K ranges from 5 to 1000. The results are listed in the top half of Tab. V. For Human3.6M [2] and UP-3D [11], the highest performances are achieved when K equals to 50. For 3DPW [3] and MPI-INF-3DHP [10], 10 prototypes achieve the best results. When K is too small, the capacity of prototype memory is not sufficient to provide the best matching prototype for each data sample. When K is too large, the data samples assigned to each prototype become insufficient to train the conditional regressor.
Influence of the Limb Weights. We next study the influence of the limb weights in performing part-aware clustering. The number of prototypes are set as the optimal value for each dataset, *i.e.* 50 for Human3.6M and UP-3D while 10 for 3DPW and MPI-INF-3DHP. We also do experiments to examine the influence of weights of other body parts. It turns out the models' performances only differ slightly. Therefore, we only focus on the influence of limb weights in this subsection. The weights of other body parts are set to be values shown in Fig. 3. In our experiments, the limb weights range from 1.0 to 10.0. The results are listed in the bottom half of Tab. V. It is revealed that weight 5.0 achieves the best performances for all datasets. When the weight of limbs is smaller than the optimal value, the vertices of limbs are not assigned enough importance. When it is larger than the optimal value, paying too much attention to the limbs affects the learning of other parts.

V. Conclusion

In this paper, we propose the Prototypical Memory Network (PM-Net) to mitigate current single-image 3D human reconstruction models' suboptimal performances on challenging samples with rare poses or viewpoints. The key of the proposed method is a prototypical memory that learns and stores a set of 3D human prototypes that can capture compact local distributions to represent both common and rare poses. It lifts the burden from the parameter regressor in coping with diverse poses and converts the parameter regression process to start from a local compact distribution, which facilitates the model's convergence. While we mainly focus on the task of 3D human reconstruction in this paper, the notion of Prototypical Memory can be extended to other visual tasks that need to handle data with multiple modes.

Acknowledgment

This study is supported under the RIE2020 Industry Alignment Fund Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s). It is also partially supported by Singapore MOE AcRF Tier 1 (2021-T1-001-096).

TABLE V: **Influence of the number of clusters and the weights of limb.** Experimental results of models using the different numbers of prototypes and limb weights. The models are evaluated on the whole evaluation sets ("Full") and the subsets of tail classes (Tail-$x\%$) of Human3.6M [2], 3DPW [3], UP-3D [11] and MPI-INF-3DHP [10]. "Tail-$x\%$" means to only evaluate the samples with the largest $x\%$ distances to the singular prototype. We use MPJPE to evaluate MPI-INF-3DHP [10] and MPVPE for other datasets.

Dataset & Metrics →	Human3.6M [2] (MPVPE)			3DPW [3] (MPVPE)			UP-3D [11] (MPVPE)			MPI-INF-3DHP [10] (MPVPE)
Methods ↓	Full	Tail-5%	Tail-10%	Full	Tail-5%	Tail-10%	Full	Tail-10%	Tail-20%	Full
Influence of Number of Clusters										
5	75.9	85.7	86.9	115.4	130.4	132.1	112.6	178.7	165.0	97.6
10	74.9	81.1	81.7	**114.6**	**124.9**	**126.4**	112.6	174.9	159.8	**97.0**
50	**73.6**	**80.0**	**81.1**	115.2	133.3	130.9	**112.6**	**172.1**	**158.6**	98.2
100	76.4	80.9	83.1	115.1	131.2	130.4	114.4	174.0	160.0	98.4
500	74.1	86.5	86.5	114.5	136.9	133.1	114.4	185.8	165.2	98.6
1000	74.8	86.0	86.3	116.4	134.4	134.8	115.2	179.6	163.1	98.7
Influence of Weights of Limb										
1.0	75.5	84.1	83.4	**114.4**	134.2	132.0	113.0	177.0	160.7	97.9
3.0	76.6	85.7	86.9	114.8	134.9	132.1	113.7	177.0	160.1	97.9
5.0	**73.6**	**80.0**	**81.1**	114.6	**124.9**	**126.4**	**112.6**	**172.1**	**158.6**	**97.0**
7.0	75.5	88.1	88.4	115.8	137.7	135.4	113.5	180.7	162.8	98.8
10.0	73.6	80.7	82.0	115.7	130.3	128.6	112.8	178.3	161.2	97.3

References

[1] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *IEEE International Conference on Computer Vision*, 2019.

[2] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[3] T. von Marcard, R. Henschel, M. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3D human pose in the wild using IMUs and a moving camera," in *European Conference on Computer Vision*, 2018.

[4] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang, "3D human mesh regression with dense correspondence," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[5] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[6] R. A. Guler and I. Kokkinos, "Holopose: Holistic 3D human reconstruction in-the-wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[7] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[8] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Transactions on Graphics*, 2015.

[9] G. Rogez, P. Weinzaepfel, and C. Schmid, "Lcr-net++: Multi-person 2D and 3D pose detection in natural images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[10] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3D human pose estimation in the wild using improved CNN supervision," in *International Conference on 3D Vision*, 2017.

[11] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the people: Closing the loop between 3D and 2D human representations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] S. Park, J. Hwang, and N. Kwak, "3d human pose estimation using convolutional neural networks with 2d pose information," in *European Conference on Computer Vision*, 2016.

[13] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[14] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in *IEEE International Conference on Computer Vision*, 2017.

[15] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *ACM Transactions on Graphics*, 2017.

[16] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in *European Conference on Computer Vision*, 2018.

[17] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnect: Real-time multi-person 3d human pose estimation with a single rgb camera," *ACM Transactions on Graphics*, 2019.

[18] Y. Cheng, B. Wang, B. Yang, and R. T. Tan, "Monocular 3d multi-person pose estimation by integrating top-down and bottom-up networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[19] N. D. Reddy, L. Guigues, L. Pishchulin, J. Eledath, and S. G. Narasimhan, "Tessetrack: End-to-end learnable multi-person articulated 3d pose tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[20] C.-H. Chen and D. Ramanan, "3d human pose estimation= 2d pose estimation+ matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[21] G. Pavlakos, X. Zhou, and K. Daniilidis, "Ordinal depth supervision for 3d human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[22] M. Kocabas, S. Karagoz, and E. Akbas, "Self-supervised learning of 3d human pose using multi-view geometry," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[23] I. Habibie, W. Xu, D. Mehta, G. Pons-Moll, and C. Theobalt, "In the wild human pose estimation using explicit 2d features and intermediate 3d representations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 905–10 914.

[24] J. Lin and G. H. Lee, "Multi-view multi-person 3d pose estimation with plane sweep stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[25] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3d pose estimation from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[26] R. Mitra, N. B. Gundavarapu, A. Sharma, and A. Jain, "Multiview-consistent semi-supervised learning for 3d human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[27] J. Wang, S. Yan, Y. Xiong, and D. Lin, "Motion guided 3d pose estimation from videos," in *European Conference on Computer Vision*, 2020.

[28] B. Wandt, M. Rudolph, P. Zell, H. Rhodin, and B. Rosenhahn, "Canonpose: Self-supervised monocular 3d human pose estimation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[29] F. Yu, M. Salzmann, P. Fua, and H. Rhodin, "Pcls: Geometry-aware neural reconstruction of 3d pose with perspective crop layers," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[30] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3d human pose estimation with spatial and temporal transformers," in *IEEE International Conference on Computer Vision*, 2021.

[31] H. Cho, Y. Cho, J. Yu, and J. Kim, "Camera distortion-aware 3d human pose estimation in video with optimization-based meta-learning," in *IEEE International Conference on Computer Vision*, 2021.

[32] Z. Zou and W. Tang, "Modulated graph convolutional network for 3d human pose estimation," in *IEEE International Conference on Computer Vision*, 2021.

[33] T. Wehrbein, M. Rudolph, B. Rosenhahn, and B. Wandt, "Probabilistic

monocular 3d human pose estimation with normalizing flows," in *IEEE International Conference on Computer Vision*, 2021.

[34] Y. Rong, Z. Liu, C. Li, K. Cao, and C. C. Loy, "Delving deep into hybrid annotations for 3D human recovery in the wild," in *IEEE International Conference on Computer Vision*, 2019.

[35] G. Moon and K. M. Lee, "I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image," in *European Conference on Computer Vision*, 2020.

[36] Y. Rong, T. Shiratori, and H. Joo, "Frankmocap: A monocular 3d whole-body pose estimation system via regression and integration," in *IEEE International Conference on Computer Vision Workshop*, 2021.

[37] H. Joo, N. Neverova, and A. Vedaldi, "Exemplar fine-tuning for 3d human pose fitting towards in-the-wild 3d human pose estimation," in *International Conference on 3D Vision*, 2021.

[38] H. Choi, G. Moon, and K. M. Lee, "Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose," in *European Conference on Computer Vision*, 2020.

[39] J. Li, C. Xu, Z. Chen, S. Bian, L. Yang, and C. Lu, "Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[40] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3d human dynamics from video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[41] A. Arnab, C. Doersch, and A. Zisserman, "Exploiting temporal context for 3d human pose estimation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[42] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[43] J. Dong, Q. Shuai, Y. Zhang, X. Liu, X. Zhou, and H. Bao, "Motion capture from internet videos," in *European Conference on Computer Vision*, 2020.

[44] H. Choi, G. Moon, J. Y. Chang, and K. M. Lee, "Beyond static features for temporally consistent 3d human pose and shape from a video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[45] G.-H. Lee and S.-W. Lee, "Uncertainty-aware human mesh recovery from video by learning part-based 3d dynamics," in *IEEE International Conference on Computer Vision*, 2021.

[46] J. Liang and M. C. Lin, "Shape-aware human pose and shape reconstruction using multi-view images," in *IEEE International Conference on Computer Vision*, 2019, pp. 4352–4362.

[47] G. Pavlakos, N. Kolotouros, and K. Daniilidis, "Texturepose: Supervising human mesh estimation with texture consistency," in *IEEE International Conference on Computer Vision*, 2019.

[48] A. Sengupta, I. Budvytis, and R. Cipolla, "Probabilistic 3d human shape and pose estimation from multiple unconstrained images in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[49] Q. Fang, Q. Shuai, J. Dong, H. Bao, and X. Zhou, "Reconstructing 3d human pose by watching humans in the mirror," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[50] Z. Dong, J. Song, X. Chen, C. Guo, and O. Hilliges, "Shape-aware multi-person pose estimation from multi-view images," in *IEEE International Conference on Computer Vision*, 2021.

[51] K. Wang, J. Xie, G. Zhang, L. Liu, and J. Yang, "Sequential 3d human pose and shape estimation from point clouds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[52] G. Liu, Y. Rong, and S. Lu, "Votehmr: Occlusion-aware voting network for robust 3d human mesh recovery from partial point clouds," in *ACM Multimedia*, 2021.

[53] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3D human pose and shape from a single color image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[54] Y. Xu, S.-C. Zhu, and T. Tung, "Denserac: Joint 3D pose and shape estimation by dense render-and-compare," in *IEEE International Conference on Computer Vision*, 2019.

[55] G. Georgakis, R. Li, S. Karanam, T. Chen, J. Kosecka, and Z. Wu, "Hierarchical kinematic human mesh recovery," in *European Conference on Computer Vision*, 2020.

[56] A. Zanfir, E. G. Bazavan, H. Xu, B. Freeman, R. Sukthankar, and C. Sminchisescu, "Weakly supervised 3D human pose and shape reconstruction with normalizing flows," in *European Conference on Computer Vision*, 2020.

[57] L. Muller, A. A. Osman, S. Tang, C.-H. P. Huang, and M. J. Black, "On self-contact and human pose," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[58] H. Choi, G. Moon, J. Park, and K. M. Lee, "3dcrowdnet: 2d human pose-guided3d crowd human pose and shape estimation in the wild," in *IEEE International Conference on Computer Vision*, 2021.

[59] W. Jiang, N. Kolotouros, G. Pavlakos, X. Zhou, and K. Daniilidis, "Coherent reconstruction of multiple humans from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[60] M. Kocabas, C.-H. P. Huang, J. Tesch, L. Muller, O. Hilliges, and M. J. Black, "Spec: Seeing people in the wild with an estimated camera," in *IEEE International Conference on Computer Vision*, 2021.

[61] Y. Rong, J. Wang, Z. Liu, and C. C. Loy, "Monocular 3d reconstruction of interacting hands via collision-aware factorized refinements," in *International Conference on 3D Vision*, 2021.

[62] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.

[63] H. Zhang, J. Cao, G. Lu, W. Ouyang, and Z. Sun, "Danet: Decompose-and-aggregate network for 3d human shape and pose estimation," in *ACM Multimedia*, 2019.

[64] ——, "Learning 3d human shape and pose from dense body parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[65] H. Zhang, Y. Tian, X. Zhou, W. Ouyang, Y. Liu, L. Wang, and Z. Sun, "Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop," in *IEEE International Conference on Computer Vision*, 2021.

[66] R. Alp Güler, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[67] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black, "Pare: Part attention regressor for 3d human body estimation," in *IEEE International Conference on Computer Vision*, 2021.

[68] S. K. Dwivedi, N. Athanasiou, M. Kocabas, and M. J. Black, "Learning to regress bodies from images using differentiable semantic rendering," in *IEEE International Conference on Computer Vision*, 2021.

[69] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representation*, 2017.

[70] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[71] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[72] T. Zhang, B. Huang, and Y. Wang, "Object-occluded human shape and pose estimation from a single color image," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[73] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *European Conference on Computer Vision*, 2016.

[74] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[75] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[76] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[77] F. L. Markley, Y. Cheng, J. L. Crassidis, and Y. Oshman, "Averaging quaternions," *Journal of Guidance, Control, and Dynamics*, 2007.

[78] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representation*, 2015.

[79] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[80] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[81] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European Conference on Computer Vision*, 2014.

[82] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[83] M. Loper, N. Mahmood, and M. J. Black, "Mosh: Motion and shape capture from sparse markers," *ACM Transactions on Graphics*, 2014.

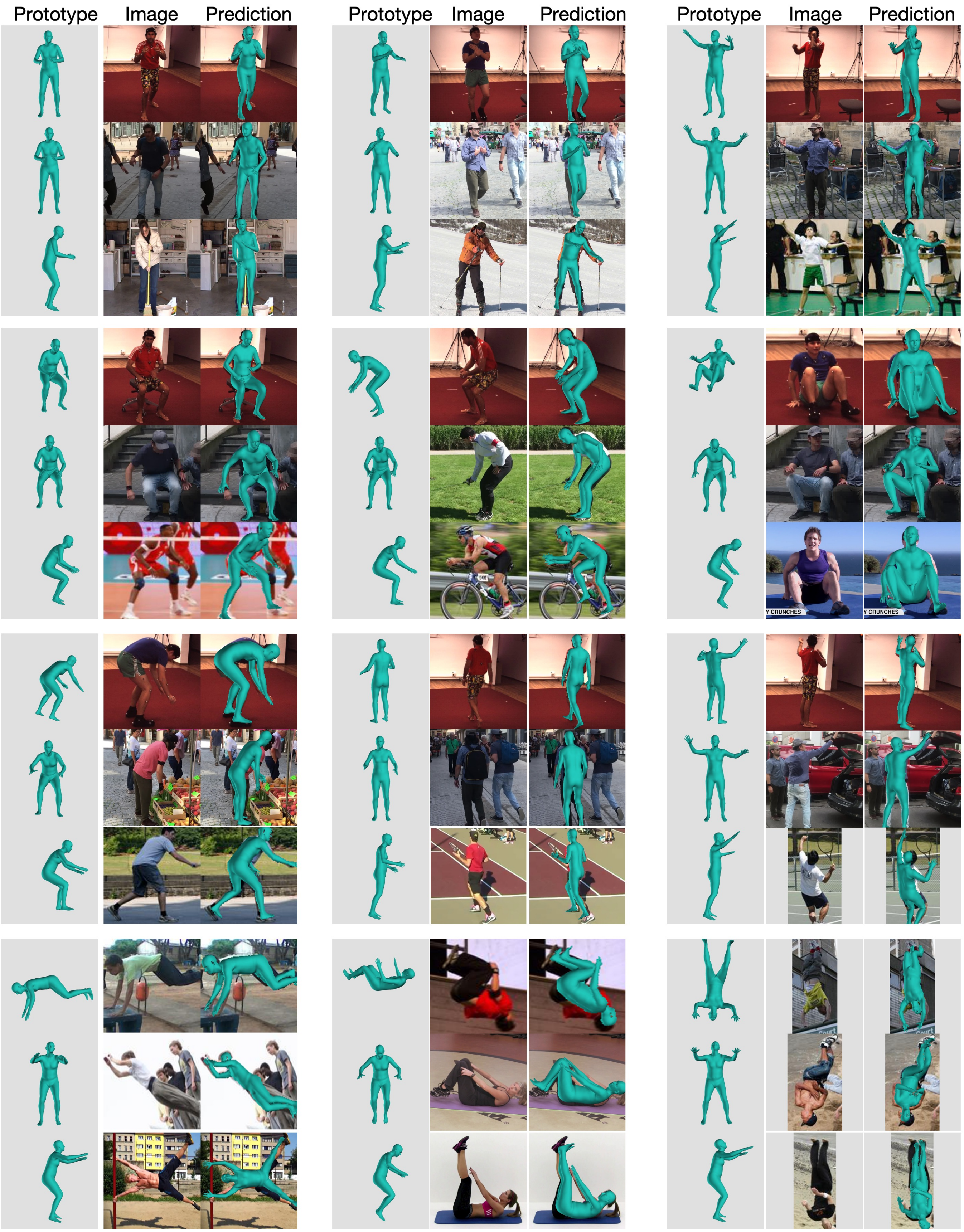
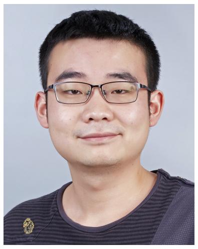


Fig. 8: **Qualitative Results.** Qualitative results of PM-Net and P3DH K-Means. The images are selected from Human3.6M [2], 3DPW [3] and UP-3D [11]. For each selected sample, we show the selected prototype, input images and PM-Net's prediction. Each prototype is rendered from original view, frontal view and side view.



Yu Rong received the BEng degree in computer science from Tsinghua University in 2016. He received the PhD degree in Multimedia Laboratory (MMLab) at the Chinese University of Hong Kong (CUHK) in 2021. He has papers accepted to CVPR/ICCV/3DV. His research interests include computer vision and deep learning, especially for topics in 3D human motion capture, including body and hand.

Ziwei Liu is currently a Nanyang Assistant Professor at Nanyang Technological University (NTU). Previously, he was a senior research fellow at the Chinese University of Hong Kong. Before that, Ziwei was a postdoctoral researcher at University of California, Berkeley, working with Prof. Stella Yu. Ziwei received his PhD from the Chinese University of Hong Kong, under the supervision of Prof. Xiaoou Tang and Prof. Xiaogang Wang. He has published extensively on top-tier conferences and journals in relevant fields, including CVPR, ICCV, ECCV, NeurIPS, ICLR, SIGGRAPH, T-PAMI, TOG, and Nature - Machine Intelligence. He is the recipient of Microsoft Young Fellowship, Hong Kong PhD Fellowship, ICCV Young Researcher Award, and HKSTP best paper award.

Chen Change Loy (Senior Member, IEEE) received the PhD degree in computer science from the Queen Mary University of London, in 2010. He is an associate professor with the School of Computer Science and Engineering, Nanyang Technological University. Prior to joining NTU, he served as a research assistant professor with the Department of Information Engineering, The Chinese University of Hong Kong, from 2013 to 2018. His research interests include computer vision and deep learning. He serves as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and the International Journal of Computer Vision. He also serves/served as an Area Chair of ICCV 2021, CVPR 2021, CVPR 2019, ECCV 2018, AAAI 2021 and BMVC 2018-2020.