

# Generating Multiple Hypotheses for 3D Human Pose Estimation with Mixture Density Network

Chen Li Gim Hee Lee

Department of Computer Science, National University of Singapore

{lic, gimhee.lee}@comp.nus.edu.sg

## Abstract

3D human pose estimation from a monocular image or 2D joints is an ill-posed problem because of depth ambiguity and occluded joints. We argue that 3D human pose estimation from a monocular input is an inverse problem where multiple feasible solutions can exist. In this paper, we propose a novel approach to generate multiple feasible hypotheses of the 3D pose from 2D joints. In contrast to existing deep learning approaches which minimize a mean square error based on an unimodal Gaussian distribution, our method is able to generate multiple feasible hypotheses of 3D pose based on a multimodal mixture density networks. Our experiments show that the 3D poses estimated by our approach from an input of 2D joints are consistent in 2D reprojections, which supports our argument that multiple solutions exist for the 2D-to-3D inverse problem. Furthermore, we show state-of-the-art performance on the Human3.6M dataset in both best hypothesis and multi-view settings, and we demonstrate the generalization capacity of our model by testing on the MPII and MPI-INF-3DHP datasets. Our code is available at the project website<sup>1</sup>.

## 1. Introduction

3D human pose estimation from a single RGB image is an extensively studied problem in computer vision because of many potential useful real world applications such as forensic science, sports analysis and surveillance *etc.* Significant progress in 3D human pose estimation has been made with deep learning in the recent years. One of the commonly used and effective deep learning based methods for 3D human pose estimation is the two-stage approach, where the 2D joints are first detected from the image input [18, 24] followed by the 3D joint estimations from the detected 2D joints [1, 29, 4, 15, 10, 6, 25, 17]. The advantage

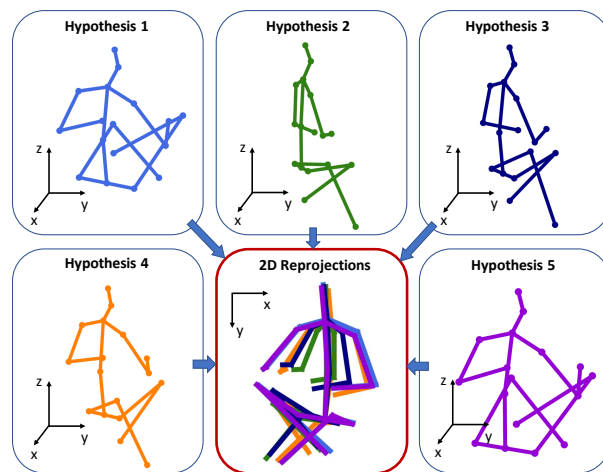


Figure 1: An example of multiple feasible 3D pose hypotheses generated from our network reprojecting into similar 2D joint locations. (Best view in color)

of the two-stage approach is that it decouples the harder problem of 3D depth estimation from the easier 2D pose estimation. In particular, variations in background scene, lighting, clothing shape, skin color *etc.* are removed before the 3D joint estimation stage. Furthermore, the model can be trained on different domains, *e.g.* indoor and outdoor, with 2D annotations that are readily available.

Despite the significant progress with deep learning, 3D human pose estimation remains as a very challenging task due to the ambiguity in recovering 3D information from a single RGB image. More specifically, recovering 3D information from a single RGB image or 2D joint locations is an inverse problem [3] where multiple solutions may exist for the depth of a 3D joint along the light ray that reprojects onto the same 2D joint location, as illustrated in Figure 1. The problem is further aggravated by the non-rigidity of the human pose and joint occlusions on the 2D image. Consequently, there could be many solutions of the 3D pose that satisfy the same 2D pose on an image, even after eliminating the infeasible 3D pose solutions by en-

<sup>1</sup><https://github.com/chaneyddtt/Generating-Multiple-Hypotheses-for-3D-Human-Pose-Estimation-with-Mixture-Density-Network>

forcing various geometric constraints, *e.g.* joint limits [1] and bone ratio [27] *etc.* In view of the inherent ambiguity of the 3D human pose estimation problem, we argue that it is more reasonable to design a model that generates multiple hypotheses of geometrically feasible 3D human pose that are consistent with the detected 2D joints from a single RGB image. In contrast, the widely adopted single estimate for the inverse problem with inherent ambiguity could lead to overfitting the model to the training data, and might not generalize well. This idea of generating multiple 3D pose hypotheses was first suggested very recently by Jahangiri and Yuille in [12].

To this end, we introduce the mixture density networks (MDN) [3, 26] to the 3D joint estimation module of the two-stage approach. Contrary to most existing works that generate a single 3D pose by minimizing the negative log-likelihood of an unimodal Gaussian, *i.e.* a mean squared error, we propose to estimate multiple hypotheses of the 3D pose by minimizing the negative log-likelihood of a multimodal mixture-of-Gaussians. The outputs of our mixture model is a set of mixing coefficients and parameters of the Gaussian kernels, *i.e.* means and variances. The set of 3D pose hypotheses are given by the means of the Gaussian kernels, and the mixing coefficient and variances represent the uncertainties of each 3D pose hypothesis. Specifically our network consists of a feature extractor to lift the 2D joints into a feature space, and a hypotheses generator to generate multiple hypotheses. The whole network is a simple network made up of several linear layers with different non-linear activation units.

We show that our network achieves state-of-the-art results on the Human3.6M dataset [11] in both best hypothesis and multi-view settings. We also report results of our network on the outdoor MPII [2] dataset and the MPI-INF-3DHP [16] dataset, where 3D pose labels are not used for training the network. Furthermore, we show the robustness of our network by applying it to scenarios where one or two limb joints are occluded/missing. Our main contributions are as follows:

- We explore the idea of generating multiple 3D pose hypotheses to alleviate the ambiguity problem that has not received much attention in the literature.
- To the best of our knowledge, we are the first to introduce the mixture density model into 3D human pose estimation, which is more powerful than the single Gaussian distribution.
- Our network achieves state-of-the-art results on Human3.6M dataset in both best hypothesis and multi-view settings, and in cases where one or two limb joints are occluded/missing.

## 2. Related Work

Existing human 3D pose estimation approaches fall into two categories according to their training techniques. The first category is to train deep convolutional neural networks (CNNs) end-to-end to estimate 3D human poses directly from the input images [20, 16, 28, 19, 27, 14, 23]. Zhou *et al.* [28] use a sparse representation for 3D poses and predict the 3D pose with an expectation-maximization (EM) algorithm. The 2D poses are regarded as a hidden variable in the EM algorithm to remove the need of synchronized 2D-3D data. Park *et al.* [19] improve conventional CNNs by concatenating 2D pose estimation as well as information on relative positions with respect to multiple joints. Pavlakos *et al.* [20] use volumetric representation to represent 3D poses and adopt the stacked hourglass network [18], which is originally designed for 2D pose estimation, to predict 3D volumetric heatmaps. Mehta *et al.* [16] use transfer learning to transfer the knowledge learned for 2D pose estimation to the task of 3D pose estimation. Similarly, Zhou *et al.* [27] propose a weakly-supervised transfer learning method that uses mixed 2D and 3D labels. The 2D pose estimation sub-network and 3D depth regression sub-network share the same features such that the 3D pose labels for indoor environments can be transferred to in-the-wild images. The direct approach benefits from the rich information contained in images, *e.g.* the front-back orientation of limbs. However, it will also be affected by a number of factors such as background, lighting, clothing *etc.* A network trained on one dataset can not be generalized well to the other datasets with different environment, for example from indoor and outdoor environment.

The second category [1, 29, 4, 15, 10, 6, 25, 17] decouples 3D pose estimation into the well-studied 2D joint detection [18, 24] and 3D pose estimation from the detected 2D joints. Akhter *et al.* [1] propose a multi-stage approach to estimate the 3D pose from 2D joints using an over-complete dictionary of poses. Bogo *et al.* [4] estimate 3D pose by first fitting a statistical body shape model to the 2D joints, and then minimizing the error between the reprojected 3D model and detected 2D joints. Chen [6] and Yasin [25] regard 3D pose estimation as a matching between the estimated 2D pose and the 3D pose from a large pose library. Martinez *et al.* [15] design a simple fully connected residual network to regress 3D pose from 2D joint detections. The decoupled approach can make use of both indoor and in-the-wild images to train the 2D pose estimators. More importantly, this approach is domain invariant since the input of the second stage is the 2D joints. However, estimating 3D pose from 2D joints is more challenging because 2D pose data contains less information than images, thus there are more ambiguities.

To solve the ill-posed problem of estimating 3D pose from 2D joints, Jahangiri and Yuille [12] first proposed

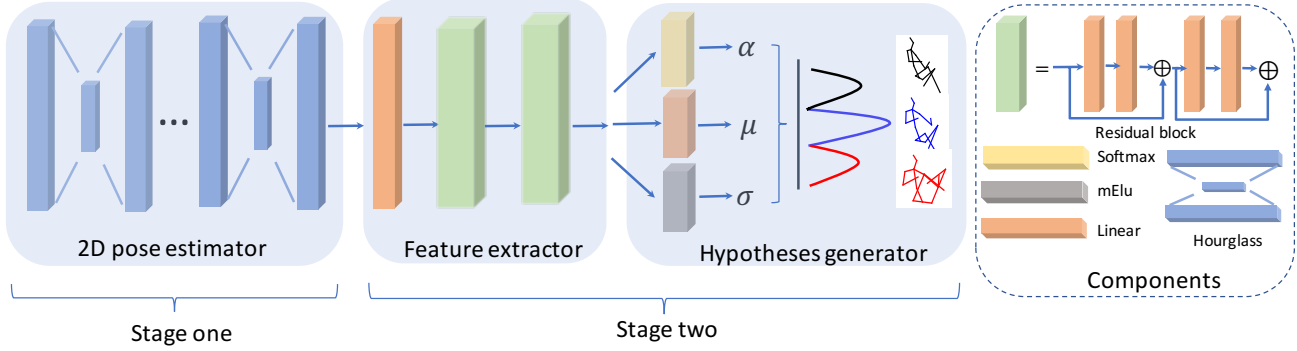


Figure 2: Our network consists of a feature extractor and a 3D pose hypotheses generator, it generates multiple pose hypotheses from the 2D joints detected by 2D pose estimator.

to generate multiple diverse pose hypotheses. They first learned a 3D Gaussian mixture model (GMM) model [22] from a uniformly sampled set of Human3.6M poses, and then use conditional sampling to get samples of the 3D poses with reprojected joints errors that are within a threshold. Inspired by their work, we solve the ambiguity problem by generating multiple hypotheses. Instead of using the traditional GMM approach, we introduce the MDN which was first proposed in [3]. The MDN can represent arbitrary conditional distributions by combining a conventional neural network with a mixture density model. Ye *et al.* [26] used a hierarchical MDN to solve the occlusion problem in hand pose estimation. Inspired by the work of Ye *et al.*, we use the MDN to solve the depth ambiguity and occlusion problem in 3D human pose estimation.

### 3. Our Mixture Density Network

Figure 2 shows the illustration of our deep network to generate multiple hypotheses for 3D human pose estimation. Our network follows the commonly used two-stage approach that first estimates the 2D joints from the input images followed by the 3D pose estimation from the estimated 2D joints. We adopt the state-of-the-art stacked hourglass [18] network as the 2D joint estimation module, and use our MDN which consists of a feature extractor and a hypotheses generator to generate the multiple 3D pose hypotheses. Given the 2D joint detections  $\mathbf{x} \in \mathbb{R}^{2N}$ , where  $N$  is the number of joints in one pose, our goal is to learn a function  $f : \mathbf{x} \mapsto \Theta$  which maps  $\mathbf{x}$  into a set of output parameters  $\Theta = \{\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\alpha}\}$  for our mixture model.  $\boldsymbol{\mu} = \{\mu_1, \dots, \mu_M \mid \mu_i \in \mathbb{R}^{3N}\}$ ,  $\boldsymbol{\sigma} = \{\sigma_1, \dots, \sigma_M \mid \sigma_i \in \mathbb{R}\}$  and  $\boldsymbol{\alpha} = \{\alpha_1, \dots, \alpha_M \mid 0 \leq \alpha_i \leq 1, \sum_i \alpha_i = 1\}$  are the means, variances and mixing coefficients of the mixture model.  $M$  is the number of Gaussian kernels. The mean of each Gaussian kernel  $\mu_i \in \boldsymbol{\mu}$  represents one 3D pose hypothesis, and the number of Gaussian kernels  $M$  decides the number of hypotheses generated by our model.

### 3.1. Model Representation

The probability density of the 3D pose  $\mathbf{y} \in \mathbb{R}^{3N}$  given the 2D joints  $\mathbf{x} \in \mathbb{R}^{2N}$  is represented as a linear combination of Gaussian kernel functions

$$p(\mathbf{y} \mid \mathbf{x}) = \sum_{i=1}^M \alpha_i(\mathbf{x}) \phi_i(\mathbf{y} \mid \mathbf{x}), \quad (1)$$

where  $M$  is the number of Gaussian kernels, *i.e.* the number of hypotheses.  $\alpha_i(\mathbf{x})$  is the mixing coefficients, which can be regarded as a prior probability of a 3D pose data  $\mathbf{y}$  being generated from the  $i^{\text{th}}$  Gaussian kernel given the input 2D joints  $\mathbf{x}$ . Here  $\alpha_i(\mathbf{x})$  must satisfy the constraint

$$0 \leq \alpha_i(\mathbf{x}) \leq 1, \quad \sum_{i=1}^M \alpha_i(\mathbf{x}) = 1. \quad (2)$$

$\phi_i(\mathbf{y} \mid \mathbf{x})$  is the conditional density of the 3D pose  $\mathbf{y}$  for the  $i^{\text{th}}$  kernel, which can be expressed as a Gaussian distribution

$$\phi_i(\mathbf{y} \mid \mathbf{x}) = \frac{1}{(2\pi)^{d/2} \sigma_i(\mathbf{x})^d} \exp - \frac{\|\mathbf{y} - \mu_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})^2}. \quad (3)$$

$\mu_i(\mathbf{x})$  and  $\sigma_i(\mathbf{x})$  denote the mean and variance of the  $i^{\text{th}}$  kernel, respectively.  $d$  is the dimension of the output 3D pose  $\mathbf{y}$ . All the parameters of the mixture model, including the mixing coefficients  $\alpha_i(\mathbf{x})$ , the mean  $\mu_i(\mathbf{x})$  and the variance  $\sigma_i(\mathbf{x})$  are functions of the input 2D pose  $\mathbf{x}$ .

Note that the mixture model degenerates to a single Gaussian distribution when the means and variances of all Gaussian kernels are similar, *i.e.*  $\mu_i(\mathbf{x}) \approx \mu(\mathbf{x})$ ,  $\sigma_i(\mathbf{x}) \approx \sigma(\mathbf{x})$  for  $i = 1, \dots, M$ . Hence,

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{x}) &= \sum_{i=1}^M \alpha_i(\mathbf{x}) \phi_i(\mathbf{y} \mid \mathbf{x}) \\ &\approx \sum_{i=1}^M \alpha_i(\mathbf{x}) \mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x})) \\ &= \mathcal{N}(\mu(\mathbf{x}), \sigma(\mathbf{x})). \end{aligned} \quad (4)$$

Specifically in our case, the 3D pose hypotheses generated by the MDN will collapse into approximately a single Gaussian when the given 2D pose is simple and less ambiguous, *e.g.* no occlusions and/or missing joints.

### 3.2. Network Architecture

From Eqn. (1), (2) and (3), we can see that all parameters  $\Theta(\mathbf{x}) = \{\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\sigma}(\mathbf{x}), \boldsymbol{\alpha}(\mathbf{x})\}$  of the Gaussian mixture distribution of  $\mathbf{y}$  are functional form of  $\mathbf{x}$ . Hence, we learn this function  $f: \mathbf{x} \mapsto \Theta$  using a deep network which can be expressed as

$$\Theta = f(\mathbf{x}; \mathbf{w}), \quad (5)$$

where  $\mathbf{w}$  is the set of learnable weights in the deep network. The probability density in Eqn. (1) can be rewritten to include the learnable weights  $\mathbf{w}$  of the deep network, *i.e.*,

$$p(\mathbf{y} | \mathbf{x}, \mathbf{w}) = \sum_{i=1}^M \alpha_i(\mathbf{x}, \mathbf{w}) \phi_i(\mathbf{y} | \mathbf{x}, \mathbf{w}), \quad (6)$$

where

$$\phi_i(\mathbf{y} | \mathbf{x}, \mathbf{w}) = \frac{1}{(2\pi)^{d/2} \sigma_i(\mathbf{x}, \mathbf{w})^d} \exp - \frac{\|\mathbf{y} - \boldsymbol{\mu}_i(\mathbf{x}, \mathbf{w})\|^2}{2\sigma_i(\mathbf{x}, \mathbf{w})^2}. \quad (7)$$

The parameters  $\Theta(\mathbf{x}, \mathbf{w}) = \{\boldsymbol{\mu}(\mathbf{x}, \mathbf{w}), \boldsymbol{\sigma}(\mathbf{x}, \mathbf{w}), \boldsymbol{\alpha}(\mathbf{x}, \mathbf{w})\}$  are now dependent on the learnable weights  $\mathbf{w}$  of the deep network  $f(\mathbf{x}; \mathbf{w})$ .

We modify the 3D pose estimation module in [15] to form our deep network  $f(\mathbf{x}; \mathbf{w})$ . More specifically, our approach is a simple multilayer neural network. Given an input of 2D joints  $\mathbf{x} \in \mathbb{R}^{2N}$ , we use one linear layer to map the input into an 1024 dimensional feature space, followed by two residual blocks which respectively consists of a linear layer, batch normalization, dropout, and Rectified Linear Units. And there are residual connections between the input and output of each residual block. Different from [15] which adds another linear layer to directly regress the 3D pose  $\mathbf{y} \in \mathbb{R}^{3N}$  from the feature space, our network estimates the parameters  $\Theta$  of the mixture model. In particular, we use different activation functions to satisfy the constraints of the three parameters  $\Theta(\mathbf{x}, \mathbf{w}) = \{\boldsymbol{\mu}(\mathbf{x}, \mathbf{w}), \boldsymbol{\sigma}(\mathbf{x}, \mathbf{w}), \boldsymbol{\alpha}(\mathbf{x}, \mathbf{w})\}$ . Specifically, we use a normal linear layer for parameter  $\boldsymbol{\mu}(\mathbf{x}, \mathbf{w})$ , a softmax function for the mixture coefficient  $\boldsymbol{\alpha}(\mathbf{x}, \mathbf{w})$  so that it lies in the range of  $[0, 1]$  and sums up to 1, and a modified ELU function [7] defined as:

$$h(t) = \begin{cases} t + 1, & \text{if } t \geq 0 \\ \gamma(\exp(t) - 1) + 1, & \text{otherwise} \end{cases} \quad (8)$$

for the variance  $\boldsymbol{\sigma}(\mathbf{x}, \mathbf{w})$  to keep it positive. Here,  $\gamma$  is a scale for negative factor.

### 3.3. Optimization

Given a training dataset with  $K$  pairs of ground truth labels for the corresponding 2D joints  $\mathbf{X}$  and 3D poses  $\mathbf{Y}$ , *i.e.*  $\{\mathbf{X}, \mathbf{Y}\} = \{\{\mathbf{x}_j, \mathbf{y}_j\} | j = 1, \dots, K\}$ , the objective is to find the maximum a posterior of the set of learnable weights  $\mathbf{w}$ . More formally, assuming that each training data is independent and identically distributed (i.i.d), the posterior distribution of  $\mathbf{w}$  is given by

$$\begin{aligned} p(\mathbf{w} | \mathbf{X}, \mathbf{Y}, \Psi) &\propto p(\mathbf{Y} | \mathbf{X}, \mathbf{w}) p(\mathbf{w} | \mathbf{X}, \Psi) \\ &= p(\mathbf{w} | \mathbf{X}, \Psi) \prod_{j=1}^K p(\mathbf{y}_j | \mathbf{x}_j, \mathbf{w}) \\ &= p(\mathbf{w} | \mathbf{X}, \Psi) \prod_{j=1}^K \sum_{i=1}^M \alpha_i(\mathbf{x}_j, \mathbf{w}) \phi_i(\mathbf{y}_j | \mathbf{x}_j, \mathbf{w}), \end{aligned} \quad (9)$$

where  $\Psi$  is the hyperparameter of the prior over the learnable weights  $\mathbf{w}$ . Hence, the optimal weight  $\mathbf{w}^*$  can be obtained from the minimization of the negative log-posterior

$$\begin{aligned} \mathbf{w}^* &= \underset{\mathbf{w}}{\operatorname{argmin}} - \ln p(\mathbf{w} | \mathbf{X}, \mathbf{Y}, \Psi) \\ &= \underset{\mathbf{w}}{\operatorname{argmin}} - \underbrace{\sum_{j=1}^K \ln p(\mathbf{y}_j | \mathbf{x}_j, \mathbf{w}) - \ln p(\mathbf{w} | \mathbf{X}, \Psi)}_{\mathcal{L}}, \end{aligned} \quad (10)$$

where  $\mathcal{L}$  is taken to be the loss function for training our deep network  $f(\mathbf{x}; \mathbf{w})$ . More specifically,

$$\begin{aligned} \mathcal{L} &= - \sum_{j=1}^K \ln p(\mathbf{y}_j | \mathbf{x}_j, \mathbf{w}) - \ln p(\mathbf{w} | \mathbf{X}, \Psi) \\ &= - \sum_{j=1}^K \ln \sum_{i=1}^M \alpha_i(\mathbf{x}_j, \mathbf{w}) \phi_i(\mathbf{y}_j | \mathbf{x}_j, \mathbf{w}) - \ln p(\mathbf{w} | \mathbf{X}, \Psi) \\ &= \mathcal{L}_{3D} + \mathcal{L}_{\text{prior}}. \end{aligned} \quad (11)$$

The prior loss  $\mathcal{L}_{\text{prior}}$  can be further evaluated into:

$$\begin{aligned} \mathcal{L}_{\text{prior}} &= - \ln p(\mathbf{w} | \mathbf{X}, \Psi) \\ &= - \ln p(\mathbf{w}, \mathbf{X} | \Psi) + \ln p(\mathbf{X} | \Psi) \\ &\propto - \ln p(\Theta(\mathbf{w}, \mathbf{X}) | \Psi) \\ &= - \ln p(\boldsymbol{\alpha}(\mathbf{w}, \mathbf{X}) | \Psi) - \ln p(\boldsymbol{\mu}(\mathbf{w}, \mathbf{X}), \boldsymbol{\sigma}(\mathbf{w}, \mathbf{X}) | \Psi), \end{aligned} \quad (12)$$

where the term  $\ln p(\mathbf{X} | \Psi)$  can be dropped in the loss function since it is independent of  $\mathbf{w}$ , and we write the random variables  $\{\mathbf{w}, \mathbf{X}\}$  in its functional form  $\Theta(\mathbf{w}, \mathbf{X})$  given by the deep network. We further assume a uniform prior over  $\boldsymbol{\mu}(\mathbf{w}, \mathbf{X})$  and  $\boldsymbol{\sigma}(\mathbf{w}, \mathbf{X})$ , and a Dirichlet conjugate prior over the mixing coefficients  $\boldsymbol{\alpha}(\mathbf{w}, \mathbf{X})$  that follows a Categorical

distribution, we get

$$\begin{aligned}\mathcal{L}_{\text{prior}} &= -\ln p(\boldsymbol{\alpha}(\mathbf{w}, \mathbf{X}) \mid \Lambda) \\ &= -\sum_{j=1}^K \ln p(\alpha_1(\mathbf{w}, \mathbf{x}_j), \dots, \alpha_M(\mathbf{w}, \mathbf{x}_j) \mid \Lambda),\end{aligned}\quad (13)$$

where

$$\begin{aligned}p(\alpha_1, \dots, \alpha_M \mid \Lambda) &= \text{Dir}_{[\alpha_1, \dots, \alpha_M]}[\lambda_1, \dots, \lambda_M] \\ &= \frac{\Gamma[\sum_{i=1}^M \lambda_i]}{\prod_{i=1}^M \Gamma[\lambda_i]} \prod_{i=1}^M \alpha_i(\mathbf{w}, \mathbf{x}_j)^{\lambda_i - 1}.\end{aligned}\quad (14)$$

$\Gamma[\cdot]$  is the Gamma function, and  $\Lambda = \{\lambda_1, \dots, \lambda_M\}$  are the hyperparameters of the Dirichlet distribution, where  $\lambda_i \in [0, 1]$  and  $\sum_i \lambda_i = 1$ . The total loss function to train our deep network is given by

$$\mathcal{L} = \mathcal{L}_{3\text{D}} + \mathcal{L}_{\text{prior}},$$

where

$$\begin{aligned}\mathcal{L}_{3\text{D}} &= -\sum_{j=1}^K \ln \sum_{i=1}^M \alpha_i(\mathbf{x}_j, \mathbf{w}) \phi_i(\mathbf{y}_j \mid \mathbf{x}_j, \mathbf{w}) \\ \mathcal{L}_{\text{prior}} &= -\sum_{j=1}^K \sum_{i=1}^M (\lambda_i - 1) \ln \alpha_i(\mathbf{w}, \mathbf{x}_j).\end{aligned}\quad (15)$$

Note that we drop  $\frac{\Gamma[\sum_{i=1}^M \lambda_i]}{\prod_{i=1}^M \Gamma[\lambda_i]}$  in  $\mathcal{L}_{\text{prior}}$  because it is independent of  $\mathbf{w}$ .

**Remarks:** The term  $\mathcal{L}_{\text{prior}}$  regularizes the mixing coefficients of our mixture model. Setting  $\lambda_i = 1$  for  $i = 1, \dots, M$  implies that we have no prior knowledge over the mixing coefficients. In our experiments, we set  $\lambda_1 = \dots = \lambda_M = C$ , where  $C > 1$  is a constant scalar value to prevent overfitting of a single Gaussian kernel in the MDN to the training data, *i.e.* a single mixing coefficient  $\alpha_i \approx 1$  and  $\alpha_{j \neq i} \approx 0$ .

## 4. Experiments

Our model is implemented in Tensorflow, and we use the ADAM [13] optimizer with an initial learning rate of 0.001 and exponential decay. The batch size is set to 64 and we initialize the weights of linear layers with the Kaiming initialization [9]. The number of Gaussian kernels is set to 5 and the hyperparameters  $\{\lambda_1, \dots, \lambda_M\}$  in Eqn. (14) are set to 2. We train our network for 200 epoches with a dropout rate of 0.5. We also apply max-norm constraint on the weight of each layer so that it is in range  $[0, 1]$ . Moreover, we clip the value of  $\alpha_i(x)$  to  $[1e-8, 1]$  and  $\sigma_i(x)$  to  $[1e-15, 1e15]$  to prevent the training loss from becoming NaN. We also use the *log-sum-exp* trick as previous work [5] to avoid the underflow problem.

### 4.1. Datasets and Protocols

We show numerical results for the Human3.6M dataset [11] and compare with other state-of-the-art approaches. We also apply our approach to other datasets including MPII [2] and MPI-INF-3DHP [16] datasets to test the generalization capacity of our network.

**Human3.6M dataset:** This is currently the largest available video pose dataset, which provides accurate 3D body joint locations recorded by a Vicon motion capture system. There are 15 activity scenarios in total such as ‘‘Walking’’, ‘‘Eating’’, ‘‘Sitting’’ and ‘‘Discussion’’, each action is performed by 7 professional actors. Accurate 2D joint locations, 3D pose annotations and camera parameters are provided. Following [15], we apply standard normalization to the 2D inputs and 3D outputs by subtracting the mean and dividing by the standard deviation of the training data. We also zero-center the 3D poses around the hip joint since we do not predict the global position of the 3D pose.

**MPII dataset:** This is a standard dataset for 2D human pose estimation, which contains 25K unconstrained images collected from YouTube videos. This is the most challenging in-the-wild dataset and we use it to test the generalization of our approach. We report qualitative result for this dataset because 3D pose information is not provided.

**MPI-INF-3DHP dataset:** It is a newly released 3D human pose dataset which is captured by a Mocap system in both indoor and outdoor scenes. We only use the test split of this dataset that includes 2935 frames from six subjects performing seven actions.

**2D detections:** We use the state-of-the-art stacked hourglass network [18] to get the 2D joint detections. The stacked hourglass network is pretrained on the MPII dataset and then fine-tuned on the Human3.6M dataset.

**Evaluation protocols:** For the Human3.6M dataset, we follow the standard protocol of using S1, S5, S6, S7 and S8 for training, and S9 and S11 for testing. The evaluation metric is the Mean Per Joint Position Error (MPJPE) in millimeter between the ground truth and the estimated 3D pose. Since our network generating multiple hypotheses for each 2D detection, we follow [12] to compute the MPJPE between the ground truth and the best 3D hypothesis generated by our network. The 3D Percentate of Correct Keypoints (3DPCK) [16] is adopted as the metric for the MPI-INF-3DHP dataset.

### 4.2. Results on Human3.6M dataset

We first report our results on the Human3.6 dataset and compare with other state-of-the-art approaches. From the results shown in Table 1, we can see that our method outperforms the others in most cases. Our approach achieves an

Table 1: Quantitative results of MPJPE in millimeter on Human3.6M under protocol # 1 and # 2. (Best result in bold)

Protocol #1	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
LinKDE <i>et al.</i> [11]	132.7	183.6	132.3	164.4	162.1	205.9	150.6	171.3	151.6	243.0	162.1	170.7	177.1	96.6	127.9	162.1
Du <i>et al.</i> [8]	85.1	112.7	104.9	122.1	139.1	135.9	105.9	166.2	117.5	226.9	120.0	117.7	137.4	99.3	106.5	126.5
Zhou <i>et al.</i> [28]	87.4	109.3	87.1	103.2	116.2	143.3	106.9	99.8	124.5	199.2	107.4	118.1	114.2	79.4	97.7	113.0
Pavlakos <i>et al.</i> [20]	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Jahangiri <i>et al.</i> [12]	63.1	55.9	58.1	64.5	68.7	61.3	55.6	86.1	117.6	71.0	71.2	66.3	57.1	62.5	61.0	68.0
Zhou <i>et al.</i> [27]	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6	75.2	111.6	64.1	66.0	51.4	63.2	55.3	64.9
Martinez <i>et al.</i> [15]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Lee <i>et al.</i> [14]	<b>43.8</b>	51.7	<b>48.8</b>	53.1	<b>52.2</b>	74.9	52.7	<b>44.6</b>	<b>56.9</b>	74.3	56.7	66.4	47.5	68.4	45.6	55.8
Ours	<b>43.8</b>	<b>48.6</b>	49.1	<b>49.8</b>	57.6	<b>61.5</b>	<b>45.9</b>	48.3	62.0	<b>73.4</b>	<b>54.8</b>	<b>50.6</b>	56.0	<b>43.4</b>	<b>45.5</b>	<b>52.7</b>
Protocol #2	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Yasin <i>et al.</i> [25]	88.4	72.5	108.5	110.2	97.1	142.5	81.6	107.2	119.0	170.8	108.2	86.9	92.1	165.7	102.0	110.1
Bogo <i>et al.</i> [4]	62.0	60.2	67.8	76.5	92.1	77.0	73.0	75.3	100.3	137.3	83.4	77.3	86.8	79.7	87.7	82.3
Moreno <i>et al.</i> [17]	66.1	61.7	84.5	73.7	65.2	67.2	60.9	67.3	103.5	74.6	92.6	69.6	71.5	78.0	73.2	74.0
Martinez <i>et al.</i> [15]	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Lee <i>et al.</i> [14]	37.4	<b>38.9</b>	45.6	43.8	48.5	54.6	39.9	39.2	53.0	68.5	51.5	<b>38.4</b>	<b>33.2</b>	55.8	37.8	45.7
Ours	<b>35.5</b>	39.8	<b>41.3</b>	<b>42.3</b>	<b>46.0</b>	<b>48.9</b>	<b>36.9</b>	<b>37.3</b>	<b>51.0</b>	<b>60.6</b>	<b>44.9</b>	40.2	44.1	<b>33.1</b>	<b>36.9</b>	<b>42.6</b>

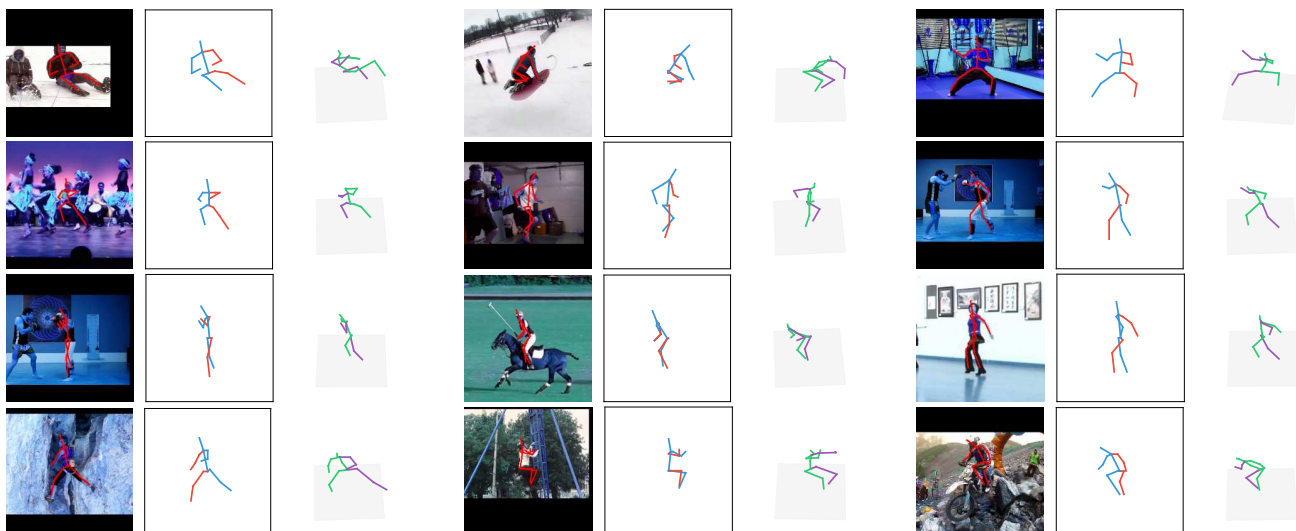


Figure 3: Qualitative results on the MPII test set. The first and second columns are the input images and output 2D joint detections of the stacked hourglass network, the last column is the 3D pose generated by our network.

Table 2: Results by using multi-view information

Methods	Lee[14]	Hossain[10]	Pavlakos [21]	Ours
Avg.	52.8	51.9	56.9	<b>49.6</b>

improvement of 5.5% compared to the previous best result 55.8 mm [14] and 16.2 % compared to the baseline architecture [15]. This indicates the efficiency of our approach by generating multiple hypotheses. Moreover, our network outperforms [12] which also generates multiple hypotheses by 22.5%. Following previous work, We show our result under Protocol #2 [4, 17] where the estimated pose has been further aligned with the ground truth via a rigid transformation. The MPJPE error in Table 1 shows that our approach consistently outperforms other approaches.

It is difficult to disambiguate the multiple 3D pose hy-

potheses generated by our model in a monocular view because most of them are feasible solutions to the inverse 2D-to-3D problem. Hence, we utilize the multi-view images from the set of calibrated cameras provided by the Human3.6M dataset to disambiguate and verify the correctness of the multiple 3D pose hypotheses generated by our network. Specifically, we transform the same pose under different cameras into the global world coordinates, and then we choose the pose which is most consistent with the poses from other camera coordinates. Finally, we get our estimated pose by averaging all poses from different camera coordinates. We list our result in Table 2 and compare with other state-of-the-art approaches based on multi-view [21] (spatial constraint) or video (temporal constraint) information [14, 10]. Note that it is however not a fair comparison

Table 3: Results with one (the first three rows) or two (the last three rows) missing joints

	Direct.	Discuss	Eating	Greet	Phone	Smoke	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Jahangiri <i>et al.</i> [12]	108.6	105.9	105.6	109.0	105.5	109.9	102.0	111.3	119.6	107.8	107.1	111.3	108.4	107.0	110.3	108.6
Martinez <i>et al.</i> [15]	57.4	61.6	64.3	65.6	73.3	85.5	61.0	62.1	84.0	101.1	68.2	66.7	70.8	55.6	59.6	69.1
Ours	<b>48.9</b>	<b>53.9</b>	<b>54.5</b>	<b>55.5</b>	<b>62.6</b>	<b>70.4</b>	<b>51.3</b>	<b>52.0</b>	<b>69.7</b>	<b>83.9</b>	<b>60.7</b>	<b>57.2</b>	<b>62.4</b>	<b>48.3</b>	<b>50.8</b>	<b>58.8</b>
Jahangiri <i>et al.</i> [12]	125.0	121.8	115.1	124.1	116.9	123.8	116.4	119.6	130.8	120.6	118.4	127.1	125.9	121.6	127.6	122.3
Martinez <i>et al.</i> [15]	62.9	66.9	69.9	71.4	80.2	93.8	66.3	65.9	90.6	109.7	74.2	72.1	75.5	61.7	65.7	75.1
Ours	<b>54.0</b>	<b>58.5</b>	<b>60.6</b>	<b>61.4</b>	<b>68.6</b>	<b>77.9</b>	<b>56.6</b>	<b>57.0</b>	<b>77.8</b>	<b>92.4</b>	<b>66.2</b>	<b>62.6</b>	<b>67.5</b>	<b>52.5</b>	<b>55.0</b>	<b>64.6</b>



Figure 4: 3D Pose hypotheses generated by our network. The first column is the input of our network, *i.e.* the 2D joints estimated by the stacked hourglass network. The second column is the ground truth 3D pose, and the third to seventh columns are the hypotheses generated by our network. The last column is the 2D reprojections of all five hypotheses. The corresponding 2D projection and 3D pose are drawn in the same color. (Best view in color)

Table 4: Quantitative results on MPI-INF-3DHP dataset

	Studio GS	Studio no GS	Outdoor	All PCK
Mehta <i>et al.</i> [16]	<b>84.1</b>	<b>68.9</b>	59.6	<b>72.5</b>
Ours	70.1	68.2	<b>66.6</b>	67.9

Table 5: Comparison between different number of kernels

Number of kernels	1	3	5	8
Avg. MPJPE	62.9	55.2	52.7	<b>52.6</b>

with other results listed in Table 1 because they did not use any multi-view or video information. The results show that our approach has the best performance among both spatial and temporal constraints based methods, indicating the advantage of our approach by generating multiple hypotheses.

In realistic scenarios, it is common that some joints are occluded and cannot be detected. In order to show that our model can handle with missing joints, we ran experiments with different number of missing joints selected randomly from the limb joints including l/r elbow, l/r wrist, l/r knee, l/r ankle. We show our results in Table 3 and compared with the baseline 2D-to-3D estimator [15] and the GMM

based methods [12] which also focus on generating multiple hypotheses. The baseline outperforms GMM based methods by a large margin, which indicates the advantage of using deep networks. Moreover, our method improves the baseline for all actions with average error decreased by 10.4mm for both cases, further showing the robustness of our method.

### 4.3. Transfer to MPII and MPI-INF-3DHP datasets

We test our method on the MPII and MPI-INF-3DHP datasets to validate the generalization capacity. Note that we train the feature extractor and hypotheses generator on the Human3.6 dataset which contains data from only the indoor environment. The validation set of MPI-INF-3DHP dataset includes images recorded under three different scenes: 1143 images in studio with green screen background (Studio GS), 1064 images in studio without green screen background (Studio no GS) and 728 images in outdoor environment (Outdoor). We use the 2D joints provided by the dataset as input and compute the 3DPCK. The results in Table 4 show that the 3DPCK of our approach is slightly lower than [16] even though we did not train on

Table 6: Comparison of our network with and without Dirichlet prior

	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Wo prior	44.4	49.6	50.0	51.0	<b>57.3</b>	63.0	46.0	49.2	64.1	78.7	55.4	51.4	56.8	<b>43.1</b>	<b>44.9</b>	53.7
W prior	<b>43.8</b>	<b>48.6</b>	<b>49.1</b>	<b>49.8</b>	57.6	<b>61.5</b>	<b>45.9</b>	<b>48.3</b>	<b>62.0</b>	<b>73.4</b>	<b>54.8</b>	<b>50.6</b>	<b>56.0</b>	43.4	45.5	<b>52.7</b>

Table 7: The similarity of the 2D reprojections of all five pose hypotheses

	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
PCKh@0.5	99.6	99.5	99.6	94.9	99.5	99.7	99.9	98.8	99.0	87.6	99.6	94.6	99.1	99.2	99.5	98.1

their dataset, indicating the generalization of our network. Moreover, our results for different scenes do not change too much compared to the results of [16]. This further suggests the domain-invariant capability of the two-stage approach that we adopted. We only give qualitative results for the MPII dataset in Figure 3 since the ground truth 3D pose data is not provided. We can see that our network can be generalized well to outdoor unseen scenes.

#### 4.4. Ablation Study

**Different number of kernels** Our hypotheses generator is based on MDN where each of the  $M$  Gaussian kernels in Eqn.(1) yields different result. We note that our network cannot fit the data completely if  $M$  is too small, while larger  $M$  requires more computation resource. We thus train three different models with  $M$  setting to 3, 5, 8, respectively. We show the average MPJPE on the Human3.6M dataset in Table 5 and compare them with the baseline method, which is based on single Gaussian distribution. The results suggest that our MDN has better performance than single Gaussian based method. Moreover, the performance does not improve much when  $M$  is larger than five. Consequently, we set  $M$  to five in our experiments in view of the trade-off between accuracy and computational complexity.

**Dirichlet prior** We add a Dirichlet conjugate prior to the distribution of the mixture coefficients  $\alpha(\mathbf{x})$  to prevent overfitting of a single Gaussian kernel to the training data. In order to explore the role of the Dirichlet prior, we compare the performance of our model with and without  $\mathcal{L}_{\text{prior}}$ . The results are shown in Table 6, it can be seen that the performance improves by adding the Dirichlet conjugate prior, especially for the difficult poses in actions “Sitting” and “SittingDown”. The reason is that most of the poses in the Human3.6 dataset are in a standing position, resulting in a worse performance on the “Sitting” and “SittingDown” actions. This further indicates that the Dirichlet conjugate prior can prevent overfitting effectively.

**What is generated by each kernel?** In order to explore the relation between different hypotheses, we reproject all five pose hypotheses into the image plane and compute the difference between projections and the 2D input joints. We

adopt the PCKh@0.5 score [18] which is the standard metric for 2D pose estimation to measure the difference. The high PCKh@0.5 score in Table 7 suggests that all the five hypotheses have almost the same 2D reprojections which are consistent with the 2D input. Note that we do not add any constraint as [12] did to force all hypotheses to be consistent in the 2D reprojections.

We give several visualization results in Figure 4 to further illustrate the relations between all pose hypotheses. As described by Eqn. (4), each Gaussian kernel can be seen to generate the same hypotheses for simple pose with less ambiguity, *e.g.* standing (first row). This means that single Gaussian distribution is sufficient for simple poses. In comparison, our network can be seen to generate different hypotheses for challenging poses like “GettingDown” or “SittingDown” (second and third rows) due to two reasons. Firstly, our network receives lesser information on this type of poses since most of the poses in the Human3.6 dataset are the “standing” poses. Secondly, there are more ambiguities and occlusions for the “GettingDown” or “SittingDown” poses. As a result, our network generates multiple pose hypotheses to mitigate the increase of the uncertainty. We also visualize the 2D reprojections of all hypotheses in the last column. We indicate the corresponding 2D reprojection and 3D pose with the same color. The overlaps between the 2D reprojections further validate that our network generates hypotheses that are consistent in the 2D image coordinates.

## 5. Conclusion

In this work, we introduce the use of a mixture density network to generate multiple feasible hypotheses for the inverse problem of 3D human pose estimation from 2D inputs. Experimental results show that our network achieves state-of-the-art results in both best hypothesis and multi-view settings. Furthermore, the 3D pose hypotheses generated by our network are consistent in the 2D reprojections suggests that the hypotheses model the ambiguity along the depth of the joints. Results on the MPII and MPI-INF-3DHP datasets further show the generalization capacity of our network.



## References

- [1] I. Akhter and M. J. Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1446–1455, 2015.
- [2] 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*, pages 3686–3693, 2014.
- [3] C. M. Bishop. Mixture density networks. Technical report, Citeseer, 1994.
- [4] 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*, pages 561–578, 2016.
- [5] A. Brando Guillaumes. Mixture density networks for distribution and uncertainty estimation. Master’s thesis, Universitat Politècnica de Catalunya, 2017.
- [6] W. Chen, H. Wang, Y. Li, H. Su, Z. Wang, C. Tu, D. Lischinski, D. Cohen-Or, and B. Chen. Synthesizing training images for boosting human 3d pose estimation. In *3D Vision, International Conference on*, pages 479–488, 2016.
- [7] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.
- [8] Y. Du, Y. Wong, Y. Liu, F. Han, Y. Gui, Z. Wang, M. Kankanhalli, and W. Geng. Marker-less 3d human motion capture with monocular image sequence and height-maps. In *European Conference on Computer Vision*, pages 20–36, 2016.
- [9] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.
- [10] M. R. I. Hossain and J. J. Little. Exploiting temporal information for 3d human pose estimation. In *European Conference on Computer Vision*, pages 69–86, 2018.
- [11] 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*, 36(7):1325–1339, 2014.
- [12] E. Jahangiri and A. L. Yuille. Generating multiple diverse hypotheses for human 3d pose consistent with 2d joint detections. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 805–814, 2017.
- [13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [14] K. Lee, I. Lee, and S. Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *European Conference on Computer Vision*, pages 119–135, 2018.
- [15] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *IEEE International Conference on Computer Vision*, pages 2640–2649, 2017.
- [16] 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 *3D Vision, International Conference on*, pages 506–516, 2017.
- [17] F. Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1561–1570, 2017.
- [18] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499, 2016.
- [19] 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*, pages 156–169, 2016.
- [20] 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*, pages 1263–1272, 2017.
- [21] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6988–6997, 2017.
- [22] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 421–428, 2004.
- [23] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In *European Conference on Computer Vision*, pages 529–545, 2018.
- [24] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.
- [25] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall. A dual-source approach for 3d pose estimation from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.
- [26] Q. Ye and T.-K. Kim. Occlusion-aware hand pose estimation using hierarchical mixture density network. *arXiv preprint arXiv:1711.10872*, 2017.
- [27] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *IEEE International Conference on Computer Vision*, pages 398–407, 2017.
- [28] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei. Deep kinematic pose regression. In *European Conference on Computer Vision*, pages 186–201, 2016.
- [29] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4966–4975, 2016.