

# PoseAug: A Differentiable Pose Augmentation Framework for 3D Human Pose Estimation

Kehong Gong<sup>\*†</sup> Jianfeng Zhang<sup>\*</sup> Jiashi Feng  
National University of Singapore

gongkehong@u.nus.edu zhangjianfeng@u.nus.edu elefjia@nus.edu.sg

## Abstract

Existing 3D human pose estimators suffer poor generalization performance to new datasets, largely due to the limited diversity of 2D-3D pose pairs in the training data. To address this problem, we present PoseAug, a new auto-augmentation framework that learns to augment the available training poses towards a greater diversity and thus improve generalization of the trained 2D-to-3D pose estimator. Specifically, PoseAug introduces a novel pose augmentor that learns to adjust various geometry factors (e.g., posture, body size, view point and position) of a pose through differentiable operations. With such differentiable capacity, the augmentor can be jointly optimized with the 3D pose estimator and take the estimation error as feedback to generate more diverse and harder poses in an online manner. Moreover, PoseAug introduces a novel part-aware Kinematic Chain Space for evaluating local joint-angle plausibility and develops a discriminative module accordingly to ensure the plausibility of the augmented poses. These elaborate designs enable PoseAug to generate more diverse yet plausible poses than existing offline augmentation methods, and thus yield better generalization of the pose estimator. PoseAug is generic and easy to be applied to various 3D pose estimators. Extensive experiments demonstrate that PoseAug brings clear improvements on both intra-scenario and cross-scenario datasets. Notably, it achieves 88.6% 3D PCK on MPI-INF-3DHP under cross-dataset evaluation setup, improving upon the previous best data augmentation based method [22] by 9.1%. Code can be found at: <https://github.com/jfzhang95/PoseAug>.

## 1. Introduction

3D human pose estimation aims to estimate 3D body joints in images or videos. It is a fundamental task with broad applications in action recognition [48, 40], human-

<sup>\*</sup>Equal contribution. <sup>†</sup>Work done during an internship at Huawei international Pte Ltd.

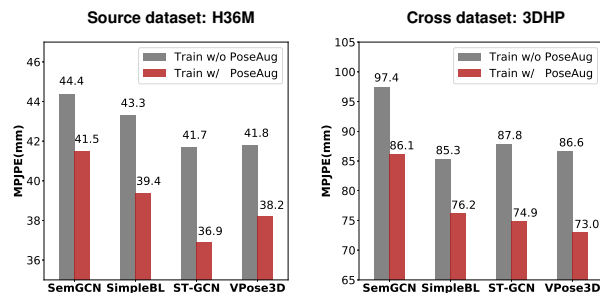


Figure 1: Estimation error (in MPJPE) on H36M (intra-dataset evaluation) and 3DHP (cross-dataset evaluation) of four well established models [53, 26, 34, 3] trained with and without PoseAug. PoseAug significantly improves their performance for both the intra- and cross-dataset settings.

robot interaction [11], human tracking [29], etc. This task is typically solved using learning-based methods [26, 53, 3, 32] with ground truth annotations that are collected in the laboratorial environments [16]. Despite their success in indoor scenarios, these methods are hardly generalizable to cross-scenario datasets (e.g., an in-the-wild dataset). We argue that their poor generalization is mainly due to the limited diversity of training data, such as limited variations in human posture, body size, camera view point and position.

Recent works explore data augmentation to improve the training data diversity and enhance the generalization of their trained models. They either generate data through image composition [38, 29, 28] and synthesis [5, 43], or directly generate 2D-3D pose pairs from the available training data by applying pre-defined transformations [22]. However, all of these works regard data augmentation and model training as two separate phases, and conduct data augmentation in an offline manner without interaction with model training. Consequently, they tend to generate ineffective augmented data that are too easy for model training, leading to marginal boost to the model generalization. Moreover, these methods heavily rely on pre-defined rules such as joint angle limitations [1] and kinematics constraints [38]

for data augmentation, which limit the diversity of the generated data and make the resulting model hardly generalize to more challenging in-the-wild scenes.

To improve the diversity of augmented data, we propose *PoseAug*, a novel auto-augmentation framework for 3D human pose estimation. Instead of conducting data augmentation and network training separately, PoseAug jointly optimizes the augmentation process with network training end-to-end in an online manner. Our main insight is that the feedback from the training process can be used as effective guidance signals to adapt and improve the data augmentation. Specifically, PoseAug exploits a differentiable augmentation module (the ‘augmentor’) implemented by a neural network to directly augment 2D-3D pose pairs in the training data. Considering the potential domain shift with respect to geometry in pose pairs (*e.g.*, postures, view points) [37, 22, 51], the augmentor learns to perform three types of augmentation operations to respectively control 1) the skeleton joint angle, 2) the body size, and 3) the view point and human position. In this way, the augmentor is able to produce augmented poses with more diverse geometric features and thus relieves the diversity limitation issue. With its differentiable capacity, the augmentor can be optimized together with the pose estimator end-to-end via an error feedback strategy. Concretely, by taking increasing training loss of the estimator as the learning target, the augmentor can learn to enrich the input pose pairs via enlarging data variations and difficulties; in turn, through combating such increasing difficulties, the pose estimator can become increasingly more powerful during the training process.

To ensure the plausibility of the augmented poses, we use a pose discriminator module to guide the augmentation, to avoid generating implausible joint angles [1], unreasonable positions or view points that may hamper model training. In particular, the module consists of a 3D pose discriminator for enhancing the joint angle plausibility and a 2D pose discriminator for guiding the body size, view point and position plausibility. The 3D pose discriminator adopts the Kinematic Chain Space (KCS) [45] representation and extends it into a *part-aware* KCS for local-wise supervision. More concretely, it splits skeleton joints into several parts and focuses on joint angles in each part separately instead of the whole body pose, which yields greater flexibility of the augmented poses. By jointly training the pose augmentor, estimator and discriminator in an end-to-end manner (Fig. 2), PoseAug can largely improve the training data diversity, and thus boost model performance on both source and more challenging cross-scenario datasets.

Our PoseAug framework is flexible regarding the choice of the 3D human pose estimator. This is demonstrated by the clear improvements made with PoseAug on four representative 3D pose estimation models [53, 26, 34, 3] over both source (H36M) [16] and cross-scenario (3DHP) [29]

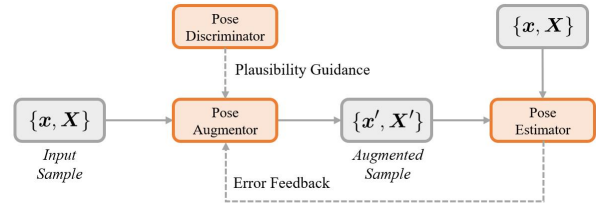


Figure 2: **Overview of our PoseAug framework.** The augmentor, estimator and discriminator are jointly trained end-to-end with an error-feedback training strategy. As such, the augmentor learns to augment data with guidance from the estimator and discriminator.

datasets (Fig. 1). Remarkably, it brings more than 13.1% average improvement w.r.t. MPJPE for all models on 3DHP. Moreover, it achieves 88.6% 3D PCK on 3DHP under cross-dataset evaluation setup, improving upon the previous best data augmentation based method [22] by 9.1%.

Our contributions are three-fold. 1) To the best of our knowledge, we are the first to investigate differentiable data augmentation on 3D human pose estimation. 2) We propose a differentiable pose augmentor, together with the error feedback design, which generates diverse and realistic 2D-3D pose pairs for training the 3D pose estimator, and largely enhances the model’s generalization ability. 3) We propose a new *part-aware* 3D discriminator, which enlarges the feasible region of augmented poses via local-wise supervision, ensuring both data plausibility and diversity.

## 2. Related Work

**3D human pose estimation** Recent progress of 3D human pose estimation is largely driven by the deployment of various deep neural network models [42, 26, 12, 53, 31, 3, 39, 54]. However, they all highly rely on well-annotated data for fully-supervised model training and hardly generalize to the new scenarios that present unseen patterns in the training dataset, such as new camera views and subject poses. Thus some recent works explore to leverage external information to improve their generalization ability. For example, some methods [55, 49, 9, 45, 14, 46, 6, 34, 20] utilize 2D pose data collected in the wild for model training, *e.g.*, through exploring kinematics priors for regularization or post-processing [55, 9, 34], and adversarial training [49, 45]. More recently, geometry-based self-supervised learning [37, 10, 4, 19, 35, 23, 36] has been used to train models with unlabeled data. Though effective, applying these methods is largely constrained by the availability of suitable external datasets. Instead of focusing on complex network architectures and learning schemes, we explore a learnable pose augmentation framework to enrich the 3D pose data at hand directly. Specifically, the proposed frame-

work can generate 2D-3D pose pairs with both diversity and plausibility for training pose estimation models. In addition, our framework is generic and can adapt to those methods to further improve their performance.

**Data augmentation on 3D human poses** Data augmentation is widely used to alleviate the bottleneck of training data diversity and improve model generalization ability. Some works augment data by stitching image patches [38, 29, 52], and some generate new data with graphics engines [5, 43]. More recently, Li *et al.*, [22] directly augment 2D-3D pose pairs through randomly applying partial skeleton recombination and joint angle perturbation on source datasets. To ensure data plausibility, several constraints are imposed, including joint angle limitation [1] and fixed augmentation range on view point and human position. Despite the good results on source data, these pre-defined rules limit the data diversity expansion and harm the model applicability to more challenging in-the-wild scenarios. Unlike all these methods, we make the first attempt to explore learnable data augmentation on 3D human pose estimation, which is shown effective for improving model generalization ability.

### 3. Method

#### 3.1. Problem Definition

Let  $\mathbf{x} \in \mathbb{R}^{2 \times J}$  denote 2D spatial coordinates of  $J$  keypoints of the human in the image, and  $\mathbf{X} \in \mathbb{R}^{3 \times J}$  denote the corresponding 3D joint position in the camera coordinate system. We aim to obtain a 3D pose estimator  $\mathcal{P} : \mathbf{x} \mapsto \mathbf{X}$  to recover the 3D pose information from the input 2D pose. Conventionally, the estimator  $\mathcal{P}$ , with parameters  $\theta$ , is trained on a well-annotated source dataset (*e.g.*, well-controlled indoor environment [16]) by solving the following optimization problem:

$$\min_{\theta} \mathcal{L}_{\mathcal{P}}(\mathcal{P}_{\theta}, \mathcal{X}) = \mathcal{L}_{\mathcal{P}}(\mathcal{P}_{\theta}(\mathbf{x}), \mathbf{X}), \quad (1)$$

where  $\mathcal{X} = \{\mathbf{x}, \mathbf{X}\}$  denotes paired 2D-3D poses from the source training dataset, and the loss function  $\mathcal{L}_{\mathcal{P}}$  is typically defined as mean square errors (MSE) between predicted and ground truth 3D poses. However, it is often observed that the pose estimator  $\mathcal{P}$  trained on such an indoor dataset can hardly generalize to a new dataset (*e.g.*, in-the-wild scenario) which features more diverse poses, body sizes, view points or human positions [14, 51, 47].

To improve generalization ability of the model, we propose to design a pose augmentor  $\mathcal{A} : \mathcal{X} \mapsto \mathcal{X}'$ , to augment the training pose pair  $\mathcal{X}$  into a more diverse one  $\mathcal{X}' = \{\mathbf{x}', \mathbf{X}'\}$  for training the model  $\mathcal{P}$ :

$$\min_{\theta} \mathcal{L}_{\mathcal{P}}(\mathcal{P}_{\theta}, \mathcal{A}(\mathcal{X})). \quad (2)$$

There are several strategies to construct the augmentor in an offline manner, *e.g.*, random [5, 29, 43] or evolution-based

augmentations [22]. Differently, we propose to implement the augmentor  $\mathcal{A}$  via a neural network with parameters  $\theta_A$  and train it jointly with the estimator in an online manner, such that the pose estimator loss can be fully exploited as a surrogate for the augmentation diversity and effectively guide the augmentor learning. In particular, the augmentor is trained to generate harder augmented samples that could increase the training loss of the current pose estimator:

$$\min_{\theta} \max_{\theta_A} \mathcal{L}_{\mathcal{P}}(\mathcal{P}_{\theta}, \mathcal{A}_{\theta_A}(\mathcal{X})). \quad (3)$$

#### 3.2. PoseAug Formulation

Our proposed framework aims to generate diverse training data, with proper difficulties for the pose estimator, to improve model generalization performance. Two challenges thus need to be tackled: how to make the augmented data diverse and beneficial for model training; and how to make them natural and realistic. To address them, we propose two novel ideas in training the augmentor.

**Error feedback learning for online pose augmentation** Instead of performing random pose augmentation in an offline manner [38, 5, 22], the proposed pose augmentor  $\mathcal{A}$  deploys a differentiable design which enables online joint-training with the pose estimator  $\mathcal{P}$ . Using the training error from the pose estimator  $\mathcal{P}$  as feedback (see Eqn. (3)), the pose augmentor  $\mathcal{A}$  learns to generate poses that are most suitable for the current pose estimator—the augmented poses present proper difficulties and diversity due to online augmentation, thus maximally benefiting generalization of the trained 3D pose estimation model.

**Discriminative learning for plausible pose augmentation** Purely pursuing error-maximized augmentations may result in implausible training poses that violate the biomechanical structure of human body and may hurt model performance. Previous augmentation methods [38, 5, 22] mostly rely on pre-defined rules for ensuring plausibility (*e.g.*, joint angle constraint [1]), which however would severely limit the diversity of generated poses. For example, some harder yet plausible poses may fail to pass their rule-based plausibility check [22] and will not be adopted for model training. To address this issue, we deploy a pose discriminator module over the local relation of body joints [45] to assist training the augmentor, thus ensuring the plausibility of augmented poses without sacrificing the diversity.

#### 3.3. Architecture

Fig. 2 summarizes our PoseAug architecture design. It includes 1) a pose augmentor that augments the input pose pair  $\{\mathbf{x}, \mathbf{X}\}$  to an augmented one  $\{\mathbf{x}', \mathbf{X}'\}$  for pose estimator  $\mathcal{P}$  training; 2) a pose discriminator module with two discriminators in 3D and 2D spaces, to ensure the plausibility of the augmented data; and 3) a 3D pose estimator, that provides pose estimation error feedback.

**Augmentor** Given a 3D pose  $\mathbf{X} \in \mathbb{R}^{3 \times J}$ , the augmentor first obtains its bone vector  $\mathbf{B} \in \mathbb{R}^{3 \times (J-1)}$  via a hierarchical transformation<sup>1</sup>  $\mathbf{B} = \mathcal{H}(\mathbf{X})$  [45, 22], which can be further decomposed into a bone direction vector  $\hat{\mathbf{B}}$  (representing the joint angle) and a bone length vector  $\|\mathbf{B}\|$  (representing the body size).

Then the augmentor applies multi-layer perceptron (MLP) for feature extraction from the input 3D pose  $\mathbf{X}$ . Additionally, a noise vector based on Gaussian distribution is concatenated with  $\mathbf{X}$  in the feature extraction process to incur sufficient randomness for enhancing the feature diversity. The extracted features are then used for regressing three operation parameters ( $\gamma_{ba}$ ,  $\gamma_{bl}$  and  $(\mathbf{R}, \mathbf{t})$ ) to change the joint angles, body size, as well as view point and position as illustrated in Fig. 3. Among these parameters,

- 1)  $\gamma_{ba} \in \mathbb{R}^{3 \times (J-1)}$  is the bone angle residual vector that is used for adjusting the Bone Angle (BA) as follows:

$$\hat{\mathbf{B}}' = \hat{\mathbf{B}} + \gamma_{ba}, \quad (\text{BA operation}). \quad (4)$$

Specifically, BA operation will rotate the input bone direction vector  $\hat{\mathbf{B}}$  by  $\gamma_{ba}$ , generating a new bone direction vector  $\hat{\mathbf{B}}'$ .

- 2)  $\gamma_{bl} \in \mathbb{R}^{1 \times (J-1)}$  represents the bone length ratio vector that is used for adjusting the Bone Length (BL):

$$\|\mathbf{B}'\| = \|\mathbf{B}\| \times (1 + \gamma_{bl}), \quad (\text{BL operation}). \quad (5)$$

BL operation modifies the input bone length vector  $\|\mathbf{B}\|$  by  $\gamma_{bl}$  to adjust the body size. Notably, to ensure bio-mechanical symmetry, the left and right body parts share the same parameters.

- 3)  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{t} \in \mathbb{R}^{3 \times 1}$  denote the rotation and translation parameters respectively for Rigid Transformation (RT) operation to control pose view point and position:

$$\mathbf{X}' = \mathbf{R}[\mathcal{H}^{-1}(\mathbf{B}')] + \mathbf{t}, \quad (\text{RT operation}), \quad (6)$$

where  $\mathbf{B}' = \|\mathbf{B}'\| \times \hat{\mathbf{B}}'$  is the augmented bone vector from the above BA and BL operations.  $\mathcal{H}^{-1}$  is the inverse hierarchical conversion to transform  $\mathbf{B}'$  back to a 3D pose [45, 22].

By applying these operations, the augmentor can generate the augmented 3D pose  $\mathbf{X}'$  with more challenging pose, body size, view point and position from the original 3D pose  $\mathbf{X}$  (Fig. 3). The augmented pose is then re-projected to 2D with  $\mathbf{x}' = \Pi(\mathbf{X}')$ , where  $\Pi: \mathbb{R}^3 \rightarrow \mathbb{R}^2$  denotes perspective projection [15] via the camera parameters from the original data. The augmented 2D-3D pair  $\{\mathbf{x}', \mathbf{X}'\}$  is then used for further training the pose estimator.

<sup>1</sup>The hierarchical transformation converts the  $J$  joints of  $\mathbf{X}$  into  $J - 1$  column vectors of  $\mathbf{B}$ , each of which represents a line segment connecting two adjacent joints.

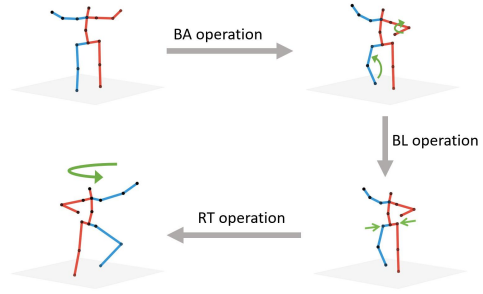


Figure 3: **Augmentation operations with PoseAug.** A source 3D pose is augmented by modifying its posture (via BA operation), body size (via BL operation) and view point and position (via RT operation).

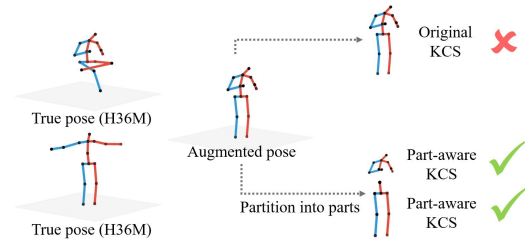


Figure 4: **Illustrations of the difference between original and part-aware KCS based discriminator.** Given a novel and valid augmented pose, the original KCS based discriminator would wrongly classify it as fake as it does not appear in source data (H36M), while the part-aware KCS based discriminator would recognize it as real and approve it, since it inspects local joint relations. It can be seen the part-aware KCS based discriminator can help the augmentor generate more diverse and plausible pose augmentation.

**Discriminator** Due to lacking priors in the augmentation procedure, the augmented poses may present implausible joint angles that violate the bio-mechanical structure [1], or unreasonable positions and view points. Though such poses are indeed harder cases for the estimator, training on them would not benefit the model generalization ability.

To ensure the plausibility of the augmented poses, we introduce a pose discriminator module to guide the augmentation. Specifically, the module consists of a 3D pose discriminator  $D_{3d}$  for evaluating the joint angle plausibility and a 2D discriminator  $D_{2d}$  for evaluating the body size, viewpoint and position plausibility.

The key to the 3D pose discriminator design is to ensure the pose plausibility without sacrificing the diversity. Inspired by the Kinematic Chain Space (KCS) [45], we design a *part-aware* KCS as input to the discriminator. Instead of taking the whole body pose into consideration as in the original KCS, our part-aware KCS only focuses on local joint angle and thus enlarges the feasible region of the augmented

pose, ensuring both plausibility and diversity (Fig. 4).

Specifically, to compute the part-aware KCS of an input pose, either  $\mathbf{X}$  or its augmentation  $\mathbf{X}'$ , we convert the pose to its bone direction vector  $\hat{\mathbf{B}}$  as above and separate it into 5 parts (torso and left/right arm/leg) [1], denoted as  $\hat{\mathbf{B}}_i, i = 1, \dots, 5$ , respectively. We then calculate the following local joint angle matrix  $KCS_{local}^i$  for the  $i$ -th part:

$$KCS_{local}^i = \hat{\mathbf{B}}_i^\top \hat{\mathbf{B}}_i, \quad (7)$$

which encapsulates the inter joint angle information within the  $i$ -th part. Based on the above local KCS representation, a 3D pose discriminator  $D_{3d}$  is constructed which takes the  $KCS_{local}^i$  as input and is trained for distinguishing the original and augmented 3D poses.

Besides the 3D discriminator, we also introduce a 2D discriminator to guide the augmentor to generate real body size, view points and positions. As the 2D poses contain information such as view point (rotation), position (translation), and body size (bone length), the 2D discriminator can learn such information through adversarial training and guide the pose augmentor in generating realistic rotation  $\mathbf{R}$ , translation  $\mathbf{t}$ , and bone length ratio  $\gamma_{bl}$ .

**Estimator** The pose estimator  $\mathcal{P}$  estimates 3D poses from 2D poses. We use the original and augmented 2D-3D pose pair  $\{\mathbf{x}, \mathbf{X}\}$  and  $\{\mathbf{x}', \mathbf{X}'\}$  to train the pose estimator. The pose estimator contains a feature extractor to capture internal features from 2D poses, and a regression layer to estimate the corresponding 3D poses. Moreover, any existing effective estimator can be implemented in our PoseAug framework. In Sec. 4.3, we conduct experiments to check robustness of PoseAug with different estimators, and the results show PoseAug can bring noticeable improvements on both source and cross-scenario datasets for all models.

### 3.4. Training Loss

**Pose estimation loss** We adopt the mean squared errors (MSE) of the ground truth (GT)  $\mathbf{X}$  and predicted poses  $\tilde{\mathbf{X}}$  as the pose estimation loss, which is formulated as

$$\mathcal{L}_{\mathcal{P}} = \|\mathbf{X} - \tilde{\mathbf{X}}\|_2^2. \quad (8)$$

We train the pose estimator using  $\mathcal{L}_{\mathcal{P}}$  with both original and augmented pose pairs jointly, which can significantly boost performance for the challenging in-the-wild scenes.

**Pose augmentation loss** To facilitate model training, augmented data should be harder than the original one, *i.e.*,  $\mathcal{L}_{\mathcal{P}}(\mathbf{X}') > \mathcal{L}_{\mathcal{P}}(\mathbf{X})$ , but not too hard to hurt the training process. A simple way to design the loss function is to let the difference between the pose estimation loss on augmented and original data within a proper range. Inspired by [25, 21], we implement a controllable feedback loss as

$$\mathcal{L}_{fb} = |1.0 - \exp[\mathcal{L}_{\mathcal{P}}(\mathbf{X}') - \beta \mathcal{L}_{\mathcal{P}}(\mathbf{X})]|, \quad (9)$$

where  $\beta > 1$  controls the difficulty level for the generated poses, making the value of  $\mathcal{L}_{\mathcal{P}}(\mathbf{X}')$  stay within a certain range w.r.t.  $\mathcal{L}_{\mathcal{P}}(\mathbf{X})$ . During training, as the pose estimator becomes increasingly more powerful, we accordingly increase  $\beta$  value to generate more challenging augmentation data for training it.

Additionally, to prevent extremely hard cases from causing training collapse, we introduce a rectified L2 loss for regularizing the augmentation parameters  $\gamma_{ba}$  and  $\gamma_{bl}$ :

$$\mathcal{L}_{reg}(\gamma) = \begin{cases} 0, & \text{if } \bar{\gamma} < \text{threshold}, \\ \|\gamma\|^2, & \text{otherwise,} \end{cases} \quad (10)$$

where  $\gamma$  denotes  $\gamma_{ba}$  and  $\gamma_{bl}$ , and  $\bar{\gamma}$  denotes the mean value over all of its elements. Combining Eqn. (9) and Eqn. (10), the overall augmentation loss  $\mathcal{L}_{\mathcal{A}}$  is formulated as

$$\mathcal{L}_{\mathcal{A}} = \mathcal{L}_{fb} + \mathcal{L}_{reg}. \quad (11)$$

**Pose discrimination loss** For the discrimination loss  $\mathcal{L}_{\mathcal{D}}$ , we adopt the LS-GAN loss [25] for both 3D and 2D spaces:

$$\mathcal{L}_{\mathcal{D}} = \mathbb{E}[(D_{3d}(\mathbf{X}) - 1)^2] + \mathbb{E}[D_{3d}(\mathbf{X}')^2] \\ + \mathbb{E}[(D_{2d}(\mathbf{x}) - 1)^2] + \mathbb{E}[D_{2d}(\mathbf{x}')^2], \quad (12)$$

where  $\{\mathbf{x}, \mathbf{X}\}$  and  $\{\mathbf{x}', \mathbf{X}'\}$  denote the original (real) and the augmented (fake) pose pairs, respectively.

**End-to-end training strategy** With the differentiable design, the pose augmentor, discriminator and estimator can be jointly trained end-to-end. We update them alternatively by minimizing losses Eqn. (11), Eqn. (12) and Eqn. (8). In addition, we first pre-train the pose estimator  $\mathcal{P}$  before training the whole framework end-to-end, which ensures stable training and produces better performance.

## 4. Experiments

We study four questions in experiments. 1) Is PoseAug able to improve performance of 3D pose estimator for both intra-dataset and cross-dataset scenarios? 2) Is PoseAug effective at enhancing diversity of training data? 3) Is PoseAug consistently effective for different pose estimators and cases with limited training data? 4) How does each component of PoseAug take effect? We experiment on H36M, 3DHP and 3DPW. Throughout the experiments, unless otherwise stated we adopt single-frame version of VPose [34] as pose estimator.

### 4.1. Datasets

**Human3.6M (H36M)** [16] Following previous works [26, 53], we train our model on subjects S1, 5, 6, 7, 8 of H36M and evaluate on subjects S9 and S11. We use two evaluation metrics: Mean Per Joint Position Error (MPJPE) in millimeters and MPJPE over aligned predictions with GT 3D poses by a rigid transformation (PA-MPJPE).

**MPI-INF-3DHP (3DHP)** [29] It is a large 3D pose dataset with 1.3 million frames, presenting more diverse motions than H36M. We use its test set to evaluate the model’s generalization ability to unseen environments, using metrics of MPJPE, Percentage of Correct Keypoints (PCK) and Area Under the Curve (AUC).

**3DPW** [44] It is an in-the-wild dataset with more complicated motions and scenes. To verify generalization of the proposed method to challenging in-the-wild scenarios, we use its test set for evaluation with PA-MPJPE as metric.

**MPII** [2] and **LSP** [17] They are in-the-wild datasets with only 2D body joint annotations and used for qualitatively evaluating model generalization for unseen poses.

## 4.2. Results

**Results on H36M** We compare PoseAug with state-of-the-art methods [53, 39, 34, 30, 22] on H36M. Similar to [22], we use 2D poses from HR-Net [41] as inputs. As shown in Table 1, our method outperforms SOTA methods [53, 39, 34, 30] by a large margin, indicating its effectiveness. Notably, compared with the previous best augmentation method [22], our PoseAug achieves lower MPJPE even though it uses external bone length data for data augmentation and nearly  $3\times$  more data than ours for model training. This clearly verifies advantages of PoseAug’s online augmentation scheme—it can generate more diverse and informative data that better benefit model training.

Table 1: **Results on H36M** in terms of MPJPE and PA-MPJPE. Best results are shown in **bold**.

Method	MPJPE (↓)	PA-MPJPE (↓)
SemGCN (CVPR’19) [53]	57.6	-
Sharma <i>et al.</i> (CVPR’19) [39]	58.0	40.9
VPose (CVPR’19) [34] (1-frame)	52.7	40.9
Moon <i>et al.</i> (ICCV’19) [30]	54.4	-
Li <i>et al.</i> (CVPR’20) [22]	50.9	<b>38.0</b>
Ours	<b>50.2</b>	39.1

Table 2: **Results on 3DHP**. CE denotes cross-scenario evaluation. PCK, AUC and MPJPE are used for evaluation.

Method	CE	PCK (↑)	AUC (↑)	MPJPE (↓)
Mehta <i>et al.</i> [27]		76.5	40.8	117.6
VNect [29]		76.6	40.4	124.7
Multi Person [28]		75.2	37.8	122.2
OriNet [24]		81.8	45.2	89.4
LCN [8]	✓	74.0	36.7	-
HMR [18]	✓	77.1	40.7	113.2
SRNet [50]	✓	77.6	43.8	-
Li <i>et al.</i> [22]	✓	81.2	46.1	99.7
RepNet [45]	✓	81.8	54.8	92.5
Ours	✓	<b>88.6</b>	<b>57.3</b>	<b>73.0</b>
Ours(+Extra2D)	✓	89.2	57.9	71.1

**Results on 3DHP (cross-scenario)** We then evaluate how PoseAug facilitates model generalization to cross-scenario datasets. We compare PoseAug against various state-of-the-art methods, including the latest one using offline data augmentation [22], the ones exploiting complex network architecture [8, 50] and weakly-supervised learning [18, 45] and the ones trained on the training set of 3DHP [27, 29, 28, 24]. From Table 2, we can observe our method achieves the best performance w.r.t. all the metrics, outperforming previous approaches by a large margin. This verifies the effectiveness of PoseAug in improving model generalization to unseen scenarios. Moreover, PoseAug can further improve the performance (from 73.0 to 71.1 in MPJPE) by using additional in-the-wild 2D poses (MPII) to train the 2D discriminator. This demonstrates its extensibility in leveraging extra 2D poses to further enrich the diversity of augmented data.

**Results on 3DPW (cross-scenario)** We train four 3D pose estimators [26, 53, 3, 33] without and with PoseAug on H36M and compare their generalization performance on 3DPW. As shown in Table 4, on average, PoseAug brings 10.8% improvements for all the models.

Table 3: Results in PA-MPJPE for four estimators on 3DPW.

Method	PA-MPJPE (↓)
SemGCN [53]	152.3
+ PoseAug	<b>140.0</b> (-12.3)
SimpleBaseline [26]	145.2
+ PoseAug	<b>130.3</b> (-14.9)
ST-GCN [3](1-frame)	154.3
+ PoseAug	<b>129.7</b> (-24.6)
VPose [34] (1-frame)	146.3
+ PoseAug	<b>132.8</b> (-13.4)

**Qualitative results** For subjective evaluation, we choose four challenging datasets, *i.e.*, LSP, MPII, 3DHP and 3DPW, with large varieties of postures, body sizes, and view points between their data and the data from H36M. Results are shown in Fig. 5. We can see our method performs fairly well, even for those unseen difficult poses.

## 4.3. Analysis on PoseAug

**Applicability to different estimators** Our PoseAug framework is generic and applicable to different 3D pose estimators. To demonstrate this, we employ four representative 3D pose estimators as backbones: 1) SemGCN [53], a graph-based 3D pose estimation network; 2) SimpleBaseline [26], an effective MLP-based network; 3) ST-GCN [3] (1-frame), a pioneer network that uses GCN-based architecture to encode global and local joint relations; and 4) VPose [34] (1-frame), a fully-convolutional network with SOTA performance. We train these models on the H36M dataset using 2D poses from four different 2D pose detectors, including CPN [7], DET [13], HR-Net [41] and groundtruth (GT). We evaluate these models on the test set of H36M and 3DHP

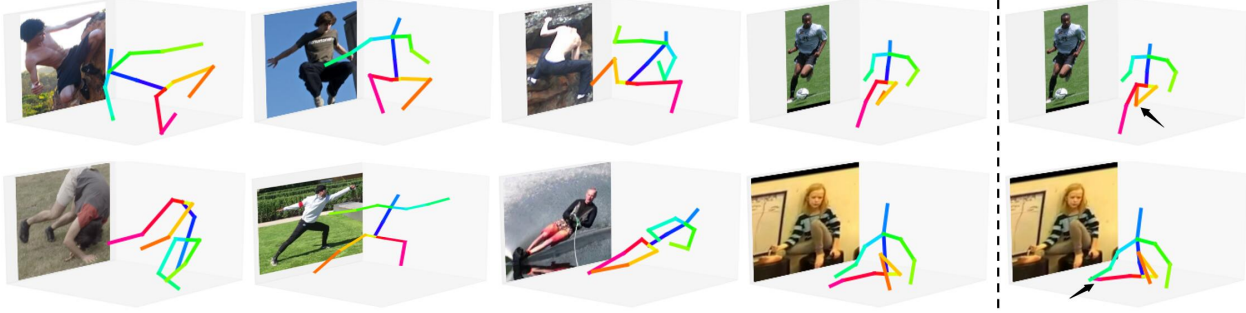


Figure 5: Example 3D pose estimations from LSP, MPII, 3DHP and 3DPW. Our results are shown in the left four columns. The rightmost column shows results of *Baseline—VPose* [34] trained w/o PoseAug. Errors are highlighted by black arrows.

Table 4: Performance comparison in MPJPE for various pose estimators trained w/o and with PoseAug on H36M and 3DHP datasets. DET, CPN, HR and GT denote 3D pose estimation model trained on different 2D pose sources, respectively. We evaluate the model on H36M test set with the corresponding 2D pose sources. On 3DHP test set, we use GT 2D poses as input for evaluating model’s generalization. We can observe PoseAug consistently decreases errors for all datasets and estimators.

Method	H36M				3DHP			
	DET	CPN	HR	GT	DET	CPN	HR	GT
SemGCN [53]	67.5	64.7	57.5	44.4	101.9	98.7	95.6	97.4
+ PoseAug	<b>65.2</b> (-2.3)	<b>60.0</b> (-4.8)	<b>55.0</b> (-2.5)	<b>41.5</b> (-2.8)	<b>89.9</b> (-11.9)	<b>89.3</b> (-9.4)	<b>89.1</b> (-6.5)	<b>86.1</b> (-11.2)
SimpleBaseline [26]	60.5	55.6	53.0	43.3	91.1	88.8	86.4	85.3
+ PoseAug	<b>58.0</b> (-2.5)	<b>53.4</b> (-2.2)	<b>51.3</b> (-1.7)	<b>39.4</b> (-3.9)	<b>78.7</b> (-12.4)	<b>78.7</b> (-10.1)	<b>76.4</b> (-10.1)	<b>76.2</b> (-9.1)
ST-GCN [3] (1-frame)	61.3	56.9	52.2	41.7	95.5	91.3	87.9	87.8
+ PoseAug	<b>59.8</b> (-1.5)	<b>54.5</b> (-2.4)	<b>50.8</b> (-1.5)	<b>36.9</b> (-4.8)	<b>83.5</b> (-12.1)	<b>77.7</b> (-13.6)	<b>76.6</b> (-11.3)	<b>74.9</b> (-12.9)
VPose [34] (1-frame)	60.0	55.2	52.7	41.8	92.6	89.8	85.6	86.6
+ PoseAug	<b>57.8</b> (-2.2)	<b>52.9</b> (-2.3)	<b>50.2</b> (-2.5)	<b>38.2</b> (-3.6)	<b>78.3</b> (-14.4)	<b>78.4</b> (-11.4)	<b>73.2</b> (-12.4)	<b>73.0</b> (-13.6)

w.r.t. MPJPE metric. On H36M, we use the corresponding 2D poses for evaluation; while on 3DHP, we evaluate these models with GT 2D poses to filter out the influence of 2D pose detectors. The results are shown in Table 4. We can see PoseAug brings clear improvements to all models on both H36M and more challenging 3DHP datasets. Notably, they obtain more than 13.1% average improvement on 3DHP when trained with PoseAug.

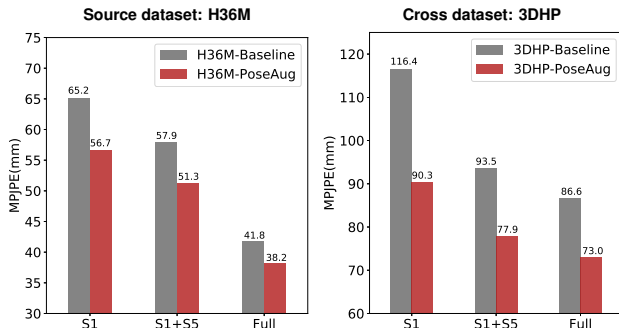


Figure 6: Ablation study on limited data setup. We report MPJPE for evaluation. Best viewed in color.

**Effectiveness for limited training data cases** 3D pose annotations are expensive to collect, making limited training

data a common challenge. To demonstrate the effectiveness of our method on addressing such cases, we use pose data from H36M S1 and S1+S5 for model training which only contain 16% and 41% training samples, respectively. The results in Fig. 6 show PoseAug consistently improves model performance with varying amounts of training data, on both H36M and 3DHP. Meanwhile, the improvements brought by our method are more significant for cases with less training data (*e.g.*, MPJPE in 3DHP, S1: 116.4  $\rightarrow$  90.3, Full: 86.6  $\rightarrow$  73.0). Moreover, in cross-scenario generalization, our method trained with only S1 achieves the comparable result (MPJPE: 90.3) to baseline trained using full dataset (MPJPE: 86.6), and our method trained with S1+S5 can outperform baseline trained using full dataset by a large margin (77.9 vs 86.6 in MPJPE).

**Analysis on the augmentor** We then check the effectiveness of each module in augmentor. Table 5 summarizes the results. By gradually adding the BA, RT and BL operations, the pose estimation error can be monotonically decreased from 41.8/86.6 to 38.8/73.5 (on H36M/3DHP). Moreover, incorporating the error feedback guidance can further improve performance to 38.2 for H36M and 73.0 for 3DHP. These verify the effectiveness of each module of the augmentor in producing more effective augmented

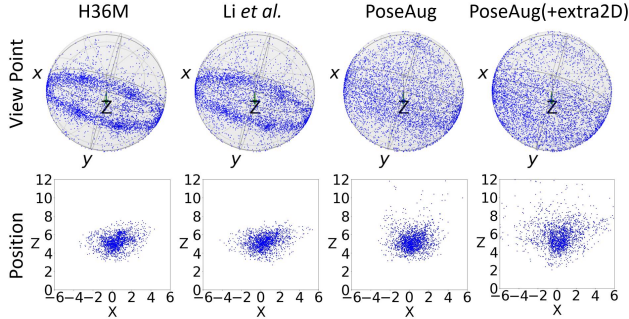


Figure 7: Distribution on view point (top row) and position (bottom row) for original data H36M, and augmented data from Li *et al.* [22], PoseAug (3rd column) and PoseAug with extra 2D poses. This distribution shows PoseAug significantly improves diversity of view point and position.

samples. Among these modules, RT contributes the most to cross-scenario performance, which implies it benefits data diversity most effectively.

Table 5: Ablation study on components of the augmentor. We report MPJPE on H36M and 3DHP datasets.

Method	BA	RT	BL	Feedback	H36M ( $\downarrow$ )	3DHP ( $\downarrow$ )
Baseline					41.8	86.6
Variant A	✓				39.7 (-2.1)	85.2 (-1.4)
Variant B		✓			39.2 (-2.6)	75.9 (-10.7)
Variant C	✓	✓			39.1 (-2.7)	75.5 (-11.1)
Variant D	✓	✓	✓		38.8 (-3.0)	73.5 (-13.1)
PoseAug	✓	✓	✓	✓	<b>38.2</b> (-3.6)	<b>73.0</b> (-13.6)

**Analysis on diversity improvement** To demonstrate effectiveness of PoseAug in enhancing data diversity, considering RT operation which augments the view point and position contributes the most to cross-scenario performance, as shown in Table 5, we make diversity analysis on view point and position distribution. Fig. 7 demonstrates the distributions of view point and position of H36M and the augmented data generated by Li *et al.* [22] and our method. For H36M data, one can observe their view points concentrate near to the xz-plane with a limited diversity along the y-axis; and their positions form a small and concentrated cluster, also showing a limited diversity. This explains why the model trained on H36M hardly generalizes to in-the-wild scenarios. Similarly, we observe small divergence for the view point and position distribution of augmented data from Li *et al.* [22]. This implies the diversity improvement from the handcrafted rule is limited. Comparably, our PoseAug can offer more plausible view points and positions using the learnable augmentor, with a much greater diversity. In addition, the diversity on human positions can be further improved with extra 2D poses, which also explains its resulted improved generalization ability in Table 2.

**Analysis on the discriminator** We here demonstrate the effectiveness of plausibility guidance from the 2D and 3D discriminators. Table 6 summarizes the results. By adding one of the 2D or 3D discriminators, the performance of baseline can be boosted by 2.2/5.8 and 2.2/7.0 on H36M/3DHP, respectively. Including both discriminators into PoseAug training can further boost the performance by 3.6/13.6 on H36M/3DHP, which clearly verify the effectiveness of both discriminators and also the importance of plausibility (in augmented poses) for estimator performance.

Table 6: Ablation study on the discriminators  $\mathcal{D}_{2D}$  and  $\mathcal{D}_{3D}$  on H36M and 3DHP. MPJPE is used for evaluation.

Method	$\mathcal{D}_{2D}$	$\mathcal{D}_{3D}$	H36M ( $\downarrow$ )	3DHP ( $\downarrow$ )
Baseline			41.8	86.6
Variant A	✓		39.6 (-2.2)	80.8 (-5.8)
Variant B		✓	39.6 (-2.2)	79.6 (-7.0)
PoseAug	✓	✓	<b>38.2</b> (-3.6)	<b>73.0</b> (-13.6)

**Analysis on part-aware KCS (PA-KCS)** To verify its effectiveness, we replace it in PoseAug with KCS [45]. Table 7 summarizes the results. PA-KCS clearly outperforms KCS on both 3DHP and 3DPW. This verifies our PA-KCS provides better guidance than KCS during training.

Table 7: Ablation study on part-aware KCS (PA-KCS). We report MPJPE on 3DHP and PA-MPJPE on 3DPW.

Method	KCS	PA-KCS	3DHP ( $\downarrow$ )	3DPW ( $\downarrow$ )
Baseline			86.6	145.2
Variant A	✓		77.7 (-8.9)	133.9 (-11.3)
PoseAug		✓	<b>73.0</b> (-13.6)	<b>130.3</b> (-14.9)

## 5. Conclusion

In this paper, we develop an auto-augmentation framework, PoseAug, that learns to enrich the diversity of training data and improves performance of the trained pose estimation models. The PoseAug effectively integrates three components including the augmentor, estimator and discriminator and makes them fully interacted with each other. Specifically, the augmentor is designed to be differentiable and thus can learn to change major geometry factors of the 2D-3D pose pair to suit the estimator better by taking its training error as feedback. The discriminator can ensure the plausibility of augmented data based on a novel part-aware KCS representation. Extensive experiments justify PoseAug can augment diverse and informative data to boost estimation performance for various 3D pose estimators.

**Acknowledgement** This research was partially supported by AISG-100E-2019-035, MOE2017-T2-2-151, NUS\_ECRA\_FY17\_P08 and CRP20-2017-0006. JZ would like to acknowledge the support of NVIDIA AI Tech Center (NVAITC) to this research project.



## References

- [1] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *CVPR*, 2015.
- [2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014.
- [3] Yujun Cai, Lihao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *ICCV*, 2019.
- [4] Ching-Hang Chen, Amrith Tyagi, Amit Agrawal, Dylan Drover, Rohith MV, Stefan Stojanov, and James M Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *CVPR*, 2019.
- [5] Wenzheng Chen, Huan Wang, Yangyan Li, Hao Su, Zhenhua Wang, Changhe Tu, Dani Lischinski, Daniel Cohen-Or, and Baoquan Chen. Synthesizing training images for boosting human 3d pose estimation. In *3DV*, 2016.
- [6] Xipeng Chen, Kwan-Yee Lin, Wentao Liu, Chen Qian, and Liang Lin. Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation. In *CVPR*, 2019.
- [7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018.
- [8] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *ICCV*, 2019.
- [9] Rishabh Dabral, Anurag Mundhada, Uday Kusupati, Safeer Afaque, Abhishek Sharma, and Arjun Jain. Learning 3d human pose from structure and motion. In *ECCV*, 2018.
- [10] Dylan Drover, Rohith MV, Ching-Hang Chen, Amit Agrawal, Amrith Tyagi, and Cong Phuoc Huynh. Can 3d pose be learned from 2d projections alone? In *ECCVw*, 2018.
- [11] Andrew Errity. Human-computer interaction. *An Introduction to Cyberpsychology*, 241, 2016.
- [12] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *AAAI*, 2018.
- [13] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. <https://github.com/facebookresearch/detectron>, 2018.
- [14] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *CVPR*, 2019.
- [15] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.
- [16] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2014.
- [17] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010.
- [18] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.
- [19] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *CVPR*, 2019.
- [20] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019.
- [21] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Pointaugument: an auto-augmentation framework for point cloud classification. In *CVPR*, 2020.
- [22] Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In *CVPR*, 2020.
- [23] Yang Li, Kan Li, Shuai Jiang, Ziyue Zhang, Congzhen-tao Huang, and Richard Yi Da Xu. Geometry-driven self-supervised method for 3d human pose estimation. In *AAAI*, 2020.
- [24] Chenxu Luo, Xiao Chu, and Alan Yuille. Orinet: A fully convolutional network for 3d human pose estimation. *BMVC*, 2018.
- [25] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.
- [26] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017.
- [27] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3DV*, 2017.
- [28] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *3DV*, 2018.
- [29] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Trans. on Graphics*, 36(4):44, 2017.
- [30] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In *ICCV*, 2019.
- [31] Aiden Nibali, Zhen He, Stuart Morgan, and Luke Prendergast. 3d human pose estimation with 2d marginal heatmaps. In *WACV*, 2019.
- [32] Xuecheng Nie, Jianfeng Zhang, Shuicheng Yan, and Jiashi Feng. Single-stage multi-person pose machines. In *ICCV*, 2019.
- [33] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Al-

- ban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPSw*, 2017.
- [34] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, 2019.
- [35] Aleksis Pirinen, Erik Gärtner, and Cristian Sminchisescu. Domes to drones: Self-supervised active triangulation for 3d human pose reconstruction. In *NeurIPS*, 2019.
- [36] Helge Rhodin, Victor Constantin, Isinsu Katircioglu, Mathieu Salzmann, and Pascal Fua. Neural scene decomposition for multi-person motion capture. In *CVPR*, 2019.
- [37] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *ECCV*, 2018.
- [38] Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. In *NeurIPS*, 2016.
- [39] Saurabh Sharma, Pavan Teja Varigonda, Prashast Bindal, Abhishek Sharma, and Arjun Jain. Monocular 3d human pose estimation by generation and ordinal ranking. In *ICCV*, 2019.
- [40] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *CVPR*, 2019.
- [41] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.
- [42] Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Direct prediction of 3d body poses from motion compensated sequences. In *CVPR*, 2016.
- [43] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.
- [44] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018.
- [45] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *CVPR*, 2019.
- [46] Luyang Wang, Yan Chen, Zhenhua Guo, Keyuan Qian, Mude Lin, Hongsheng Li, and Jimmy S Ren. Generalizing monocular 3d human pose estimation in the wild. In *ICCVw*, 2019.
- [47] Zhe Wang, Daeyun Shin, and Charless C Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. In *ECCVw*, 2020.
- [48] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.
- [49] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *CVPR*, 2018.
- [50] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In *ECCV*, 2020.
- [51] Jianfeng Zhang, Xuecheng Nie, and Jiashi Feng. Inference stage optimization for cross-scenario 3d human pose estimation. In *NeurIPS*, 2020.
- [52] Jianfeng Zhang, Dongdong Yu, Jun Hao Liew, Xuecheng Nie, and Jiashi Feng. Body meshes as points. In *CVPR*, 2021.
- [53] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, 2019.
- [54] Kun Zhou, Xiaoguang Han, Nianjuan Jiang, Kui Jia, and Jiangbo Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In *ICCV*, 2019.
- [55] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *ICCV*, 2017.