Global Adaptation meets Local Generalization: Unsupervised Domain Adaptation for 3D Human Pose Estimation

Wenhao Chai\textsuperscript{1} Zhongyu Jiang\textsuperscript{2} Jenq-Neng Hwang\textsuperscript{2} Gaoang Wang\textsuperscript{1, 2} \\
\textsuperscript{1} Zhejiang University \hspace{1cm} \textsuperscript{2} University of Washington

Abstract

When applying a pre-trained 2D-to-3D human pose lifting model to a target unseen dataset, large performance degradation is commonly encountered due to domain shift issues. We observe that the degradation is caused by two factors: 1) the large distribution gap over global positions of poses between the source and target datasets due to variant camera parameters and settings, and 2) the deficient diversity of local structures of poses in training. To this end, we combine global adaptation and local generalization in PoseDA, a simple yet effective framework of unsupervised domain adaptation for 3D human pose estimation. Specifically, global adaptation aims to align global positions of poses from the source domain to the target domain with a proposed global position alignment (GPA) module. And local generalization is designed to enhance the diversity of 2D-3D pose mapping with a local pose augmentation (LPA) module. These modules bring significant performance improvement without introducing additional learnable parameters. In addition, we propose local pose augmentation (LPA) to enhance the diversity of 3D poses following an adversarial training scheme consisting of 1) a augmentation generator that generates the parameters of pre-defined pose transformations and 2) an anchor discriminator to ensure the reality and quality of the augmented data. Our approach can be applicable to almost all 2D-3D lifting models. PoseDA achieves 61.3 mm of MPJPE on MPI-INF-3DHP under a cross-dataset evaluation setup, improving upon the previous state-of-the-art method by 10.2%.

1. Introduction

3D human pose estimation is an essential computer vision task which aims to estimate the coordinates of 3D joints from single-frame images or videos. This task can be further used for several downstream tasks in multiple object tracking [1, 15, 13, 21], person re-identification [45], action recognition [44], robot [47], human body reconstruction [14], sports application [64], etc. However, large-scale 3D-annotated datasets are hard to obtain. Existing methods are usually built on an off-the-shelf 2D pose estimators [9, 46] following two-stage schemes.

Deep learning methods [37, 41] have achieved great success in the mapping from 2D to 3D in the past decade. Despite their success in in-distribution data, these fully-supervised methods show poor performance in cross-dataset inference [18]. We argue that the real bottleneck lies in the domain gap of 3D pose data rather than the 2D-3D lifting network architecture or training strategy. Existing datasets either lack enough diversity in laboratorial environments [23] or lack adequate quantity and accuracy in the wild [38] due to the complex visual condition [59, 57, 58, 34, 33, 24]. We model the pose domain gap in terms of global position and local pose separately shown in Figure 1. As for the global position, the camera intrinsic and extrinsic parameters are completely different in different datasets, resulting in performance degradation in cross-dataset evaluation. And for the local pose, the lack of action diversity also limits the generalization ability of the model. Addressing the domain adaptation or generalization problem is a crucial step for 3D human pose estimation to move from toy experiments to real-world applications. Recent
works focus more on enhancing the generalization ability of the 2D-3D lifting networks or set camera view prediction as an auxiliary task [52, 54] to address adaptation problems. Some methods apply data augmentation in training images through image transformations [42, 39, 40] or human synthetics [8, 22, 49]. However, our proposed method does not rely on RGB or temporal information. Specifically, our method first generates transformed pose pairs from source dataset and then use them to train the 2D-3D lifting network, thus can fit any off-the-shelf model.

In this paper, we propose PoseDA, an unsupervised domain adaptation framework for 3D human pose estimation. Our method only requires non-sequence 2D poses (not images) and camera intrinsic parameters in target dataset as well as a large-scale 3D-annotated human pose dataset (e.g., Human3.6M [23]). In real-world scenarios, obtaining prior knowledge of the camera intrinsic parameters is often not a concern. This is due to the fact that such information is readily available from camera specifications or can be inferred from the input images or videos alone [48]. The basic idea behind the proposed method is to combine global adaptation and local generalization to address the issue of unsupervised domain adaptation for 3D human pose estimation. Global adaptation aims to align global positions of poses from source domain to target domain, and local generalization aims to enhance the diversity of local structures of poses. Therefore, our proposed method applies global position alignment strictly but only enhances the diversity of local poses. To be specific, we take a sample from the source dataset and apply transformations in terms of bone angle, bone length, and rotation. We use an augmentation generator to generate the parameters for these transformations and an anchor discriminator to ensure the realism and quality of these transformed pose pairs. As for the global position, we apply 2D global position alignment to ensure the alignment in both scales and 2D root positions between the projected 3D poses from the source domain with sampled 2D poses from the target domain. This process is solvable through geometry constraints with no additional learnable parameters. Finally, we use the transformed pose pairs to fine-tune the pre-trained 2D-3D lifting network and thus boost model performance on the target dataset without any use of 3D annotations. Our contributions are summarized as follows:

- Our approach is applicable to almost all 2D-3D lifting models. We achieve the state-of-the-art performance in Human3.6M-3DHP cross-dataset evaluation with 61.3 mm of MPIPE.

2. Related Work

2.1. Two-stage 3D Human Pose Estimation

Inspired by the rapid development of 2D human pose estimation algorithms, many works have tried to utilize 2D pose estimation results for 3D human pose estimation to improve in-the-wild performance [53, 62]. Two-stage 3D human pose estimation approaches, which first estimate 2D poses and then lift 2D poses to 3D poses, have been developed. Chen et al. [5] present a simple approach to 3D human pose estimation by performing 2D pose estimation, followed by 3D exemplar matching. Martinez et al. [37] propose a baseline focusing on lifting 2D poses to 3D with a simple yet effective neural network, which popularizes the research on lifting 2D pose to 3D space. Recently, semi/self-supervised learning based on geometry constraint [41, 6, 12, 25] has been used to train models without 3D annotations or by auxiliary losses.

2.2. Data Augmentation on 3D Human Poses

Data augmentation is widely used to improve deep model generalization ability by enhancing training data diversity. Some methods apply pose data augmentation on images [42, 40] or generate 3D synthetic data using graphics engines [8, 56, 17]. Other methods directly generate 2D-3D pose pairs by applying transformations on 3D skeleton [31, 18, 16]. Gong et al. [18] make this augmentation process differentiable and further learnable. Those transformations consist of bone angle, bone length, and rigid-body transformation.

2.3. Unsupervised Domain Adaptation for 3D Human Pose Estimation

Unsupervised domain adaptation [55] aims to transfer models from a fully-labeled source domain to an unlabeled target domain. Kundu et al. [30] address unsupervised domain adaptation problem by modeling pose uncertainty based on RGB images. Li et al. [31] are the first to generate corresponding 2D-3D pose pairs by applying skeleton transformations. Gong et al. [18] develop this data augmentation module with the differentiable form to jointly optimize the augmentation process with a end-to-end trained model. Gholami et al. [16] further utilize 2D pose in target domain to address the domain adaptation problem. However, all existing methods focus more on local pose augmentation or adaptation. In this paper, PoseDA utilizes both local pose augmentation and global position alignment, and achieves state-of-the-art performance in cross-domain tasks.
Figure 2. Our proposed unsupervised domain adaptation framework PoseDA consists of global position alignment (GPA) and local pose augmentation (LPA). The augmentation bone angle (BA), bone length (BL), and rotation (R) are applied on source 3D poses through an adversarial augmentation framework consists of an augmentation generator G and an anchor discriminator D_{3D}. Meanwhile, we also sample a 2D pose from target dataset and align the scale, x coordinate, and y coordinate in 2D screen between the target 2D box and projected source 2D box. Thus we solve the global 3D position by those geometric constraint. Finally, the augmented pose pairs combining with global 3D position and local 3D keypoints are used to train the lifting network \mathcal{P}.

3. Method

In this section, we introduce PoseDA, an unsupervised domain adaptation framework, as summarized in Figure 2, which consists of global position alignment (GPA) in Section 3.2 and local pose augmentation (LPA) in Section 3.3. Global position alignment module aims to align the 2D pose spatial distribution of both scale and location, \((x, y)\) coordinate, between source and target datasets, and local pose augmentation is designed to enhance the diversity of 3D-2D pose mapping. Finally, we formulate the training process with several loss functions. The pseudo-code for the overall training process of our method is given in Algorithm 1, and the transformation pipeline visualization of corresponding 2D-3D pose pairs is shown in Figure 3.

3.1. Problem Formulation

Let \(S = (J^{src}_{2D}, J^{src}_{3D})\) denote corresponding 2D-3D pose pairs from the source dataset, and \(J^{tar}_{2D}\) denote 2D pose from target dataset extracted by an off-the-shelf 2D pose estimator. Note that we only use \(J^{tar}_{3D}\) for evaluation but not for training. All the 3D poses \(J_{3D}\) in this paper are root-relative since we do not predict the global position.

Our method conducts data augmentation on pose pairs \(S = (J^{src}_{2D}, J^{src}_{3D})\) from source dataset. The 2D pose \(J^{tar}_{2D}\) and the camera intrinsic parameters from target dataset is used to guide this process:

\[
J^{aug}_{3D} = \mathcal{G}(J^{src}_{3D}; \theta_{cam}), \quad J^{aug}_{2D} = f_p(J^{aug}_{3D}; \theta_{cam})
\]  

where \((J^{aug}_{2D}, J^{aug}_{3D})\) denotes augmented pose pairs, \(\theta_{cam}\) denotes the camera intrinsic parameters from target dataset, and \(f_p\) denotes the projection function from 3D camera coordinates to 2D image coordinates. The augmented pose pairs are further used to train the pose lifting network.

We use several strong baseline methods (e.g., VideoPose3D [41] and SimpleBaseline [37]) for lifting 2D poses to 3D poses. Let \(\mathcal{P} : J^{2D}_{2D} \mapsto J^{3D}_{3D}\) denotes the lifting network with parameters \(\theta_{\mathcal{P}}\), which can be trained in fully-supervised paradigm as:

\[
\min_{\theta_{\mathcal{P}}} \mathcal{L}_{\mathcal{P}}(\mathcal{P}(J^{2D}_{2D}; \theta_{\mathcal{P}}), J^{3D}_{3D})
\]  

where \((J^{2D}_{2D}, J^{3D}_{3D})\) denotes paired 2D-3D pose pairs, which consist of both augmented pose pairs \((J^{aug}_{2D}, J^{aug}_{3D})\) and original pose pairs \((J^{src}_{2D}, J^{src}_{3D})\), the loss function \(\mathcal{L}_{\mathcal{P}}\) is typically defined as Mean Square Error (MSE), which is corresponding to the evaluation metric Mean Per Joint Position Error (MPJPE).

3.2. Global Position Alignment (GPA)

Global position alignment (GPA) is designed to eliminate the domain gap in viewpoints, which is simple yet efficient. By applying Monte Carlo sampling [7], the scale and location distributions of the 2D poses of the source dataset can be migrated to distributions of target dataset. We first construct pose pairs \((J^{src}_{3D}, J^{src}_{2D})\), which are pairs of root-relative 3D poses and 2D poses randomly sampled from source and target domains respectively.
Given a 2D pose $J_{2D}^{tar} = [x_{tar}, y_{tar}]^T \in \mathbb{R}^{2 \times J}$ and 3D pose $J_{3D}^{src} = [X_{src}, Y_{src}, Y_{src}]^T \in \mathbb{R}^{3 \times J}$ with the root at the origin $[0, 0, 0]^T$, GPA aims to estimate the translated 3D root position $J_r = [X_r, Y_r, Z_r]^T \in \mathbb{R}^{3 \times 1}$, to ensure the re-projected 2D pose $J_{2D}^{proj} = [x_{proj}, y_{proj}]^T$ from $J_{3D}^{src}$ + $J_r$ is close to $J_{2D}^{tar}$ as much as possible with respect to both position and scale. Denote the operation of GPA as $F$, 

$$
\hat{J}_r = F(J_{2D}^{tar}, J_{3D}^{src}),
$$

where $\hat{J}_r$ is the estimated 3D root position after translation.

We demonstrate how to obtain $\hat{J}_r$ as follows. Assume that the camera intrinsic parameters are given. The projection from 3D joints to 2D joints after translation should obey the perspective projection function as follows:

$$
x_i^{proj} = \frac{f_x(X_{src} + X_r)}{Z_{src} + Z_r} + c_x,
$$

$$
y_i^{proj} = \frac{f_y(Y_{src} + Y_r)}{Z_{src} + Z_r} + c_y,
$$

where $i$ denotes the $i$-th joint, $(f_x, f_y)$, $(c_x, c_y)$ denote focal length and principal point respectively. Note that $Z_{src} \ll Z_r$ since the absolute depth of the root joint of the person $Z_r$ is usually much larger than depth offset $Z_{src}$ of a certain joint relative to the root joint. Therefore, we assume that $Z_{src} + Z_r \approx Z_r$ holds. Then Eq. (4) becomes

$$
x_i^{proj} \approx \frac{f_x(X_{src} + X_r)}{Z_r} + c_x,
$$

$$
y_i^{proj} \approx \frac{f_y(Y_{src} + Y_r)}{Z_r} + c_y,
$$

To achieve the similar scale with $J_{2D}^{src}$ for $J_{2D}^{proj}$, we ensure that the 2D boxes should have similar size with the following constraint,

$$
\Delta x^{proj} + \Delta y^{proj} = \Delta x^{tar} + \Delta y^{tar},
$$

where $\Delta$ denotes the difference between the max-min coordinates of 2D joints, i.e., the width and height of the 2D box. Combine Eq. 5 and 6, we can get the approximated $Z_r$,

$$
\hat{Z}_r \approx \frac{f_x \Delta X_{src} + f_y \Delta Y_{src}}{\Delta x^{tar} + \Delta y^{tar}}.
$$

Then, we can align the global (root) position between $J_{2D}^{proj}$ and $J_{2D}^{tar}$ by

$$
x_r^{proj} = x_r^{tar}, \quad y_r^{proj} = y_r^{tar},
$$

where $[x_r, y_r]^T$ represents the 2D root joint. Combined with Eq. (5), we can get estimated $\hat{X}_r$ and $\hat{Y}_r$,

$$
\hat{X}_r = \frac{\hat{Z}_r(x_r^{tar} - c_x)}{f_x}, \quad \hat{Y}_r = \frac{\hat{Z}_r(y_r^{tar} - c_y)}{f_y}.
$$

Finally, we can obtain the translated root position $\hat{J}_r = [X_r, Y_r, \hat{Z}_r]^T$. The ultimate goal of Global Position Alignment is to align the 2D pose distributions of the target domain and the generated domain (after going through the full data augmentation pipeline) in terms of scale and position. There will be cases of large discrepancies between $J_{3D}^{src}$ and $J_{3D}^{aug}$ when pairing randomly on individuals. However, this is mitigated because 1) we only use box information rather than complete poses and 2) we randomly shuffle the pairing method between each epoch, which further increases diversity and avoids individual discrepancies in a statistical sense.

### 3.3. Local Pose Augmentation (LPA)

Inspired by PoseAug [18], We also apply local pose augmentation (LPA) to enhance the diversity of 2D-3D pose mappings. The augmentation transformation of 3D pose can be decoupled into perturbations of bone vector, bone length, and rotation. We design an adversarial augmentation framework, which consists of an augmentation generator to generate 3D pose transformation parameters and an anchor discriminator to ensure the realisticity, quality, and diversity of generated pose pairs. The generator and discriminator are jointly end-to-end trained following the GAN style.

**Augmentation generator.** Following PoseAug [18], we propose an augmentation generator denoted as $G$ with parameters $\theta_G$. Unlike vanilla GAN-style generator, we take a sample 3D pose from source dataset $J_{3D}^{src}$ as the condition $G_{cond}$, which is concatenated with a noise vector $z$ as the input of $G$, according to the suggestion in [16, 3]. The input 3D pose is converted to bone direction vectors representing the joint angle and bone length. Augmentation generator $G$ generates three types of 3D pose transformations: bone angle difference, bone length difference, and global rotation. The augmentation process can be represented as

$$
J_{3D}^{aug} = G(J_{3D}^{src}, z; \theta_G), \quad J_{2D}^{aug} = f_p(J_{3D}^{aug}; \theta_{cam})
$$

where $(J_{3D}^{aug}, J_{2D}^{aug})$ denotes the augmented paired 2D-3D pose pairs, $f_p$ denotes the camera projection function 4, and $\theta_{cam}$ denotes the given camera intrinsic parameters in target dataset.

**Anchor discriminator.** Let $D_{3D}$ denote the discriminator for 3D pose with parameters $\theta_{D_{3D}}$. It takes root-relative $J_{3D}$ sampled from both $J_{3D}^{src}$ and $J_{3D}^{aug}$ as input. Discriminator $D_{3D}$ works as an anchor discriminator to ensure the augmented pose $J_{3D}^{aug}$ is reasonable. Inspired by Kinematic Chain Space (KCS) [52, 51], we use KCS representation of 3D pose as input instead of 3D joints. With the help of KCS representation, the generated 3D poses $J_{3D}^{aug}$ are no longer limited in the source domain.
3.4. Training

Previous works [18, 16, 56] train the adversarial pose augmentation framework with the loss function of vanilla GAN [19] or least squares GAN (LSGAN) [36]. We argue that training based on Wasserstein distance [43, 11] can be trained stably and produce better augmented poses.

**Discriminator loss.** We adopt WGAN [2] loss for the anchor discriminator $L_{D_{3D}}$:

$$L_{D_{3D}} = \mathbb{E}_{x \sim J_{aug}^{3D}} [D_{3D}(x)] - \mathbb{E}_{x \sim J_{src}^{3D}} [D_{3D}(x)]$$  \hspace{1cm} (11)

**Generator loss.** The adversarial loss of the augmentation generator is the feedback from the anchor discriminator.

$$L_G = -\mathbb{E}_{x \sim G(J_{aug}^{3D}, z)} [D_{3D}(x)]$$  \hspace{1cm} (12)

The discriminator and generator are trained iteratively.

**Lifting network loss.** The standard Mean Squared Error (MSE) loss is adopted to the lifting network $\mathcal{P}$.

$$L_P = \| J_{GT} - J \|^2_2,$$  \hspace{1cm} (13)

where $J_{GT}$ and $J$ are ground truth and estimated 3D joints, respectively. The generator and the discriminator need to be warmed up for $w$ epoch before training lifting network.

4. Experiments

In this section, we conduct experiments on several popular human pose estimation benchmarks with comprehensive evaluation metrics. Since PoseDA is flexible regarding the architecture of the pose lifting network, we perform our framework on different strong baselines to show the universality. GT 2D pose is used as default. We also analyze the contribution of each component in ablation studies.

**Algorithm 1:** The training pipeline of our method

```
Input: $J = \{ J_{src}^{3D}, J_{aug}^{2D}, \theta_{cam} \}$
for $t \leftarrow 1$ to $T$
do
  for $i \leftarrow 1$ to $I$
do
    freeze $G$
sample and generate a batch data
    $J_{aug}^{3D}, J_{aug}^{2D} \leftarrow G(J_{src}^{3D}, z; \theta_G)$
  train discriminator
  $L_{D_{3D}} = \mathbb{E}_{x \sim J_{aug}^{3D}} [D_{3D}(x; \theta_{D_{3D}})] - \mathbb{E}_{x \sim J_{src}^{3D}} [D_{3D}(x; \theta_{D_{3D}})]$
  update $\theta_{D_{3D}}$
  train generator every $n$ iters
  if $i$ in every $n$ iters then
    freeze $\theta_{D_{3D}}$
generate a batch data
    $J_{aug}^{3D}, J_{aug}^{2D} \leftarrow G(J_{src}^{3D}, z; \theta_G)$
    $L_G = -\mathbb{E}_{x \sim J_{aug}^{3D}} [D_{3D}(x; \theta_{D_{3D}})]$
  update $\theta_G$
  warmup to ensure stable augmentation
if $t \geq w$ then
  mixup augmented and source data
  $J_{mix}^{3D}, J_{mix}^{2D} \leftarrow (J_{src}^{3D}, J_{src}^{3D}), (J_{aug}^{2D}, J_{aug}^{3D})$
  train lifting network with mixed data
  $L_P = \text{MSE} (\mathcal{P}(J_{mix}^{2D}; \theta_P), J_{mix}^{3D})$
  update $\theta_P$
```

4.1. Datasets and Metrics

The datasets used for quantitative evaluations are several widely used large-scale 3D human pose estimation benchmarks, including Human3.6M [23], MPI-INF-3DHP [38], and 3DPW [50].
Human3.6M (H3.6M) is one of the largest 3D human pose datasets captured in a laboratory environment. Following previous works [16, 41], there are two different settings of H3.6M: 1) using the S1, S5, S6, S7 and S8 as our source domain for cross-dataset evaluation. 2) using only S1 as source domain and others (S5, S6, S7, S8) as target domain.

MPI-INF-3DHP (3DHP) is a large-scale in-the-wild 3D human pose dataset with more diverse actions and motions. This dataset is closer to real-world scenarios and ideal for evaluating our method. Following previous works [29, 18], we use its test set, which includes 2,929 frames. We report the results of PoseDA using metrics of MPJPE, Percentage of Correct Keypoints (PCK), and Area Under the Curve (AUC).

3DPW is an in-the-wild dataset, unlike H3.6M or 3DHP, with uncontrolled motion and scene. Since it is a much more challenging dataset, we train each method on H3.6M and evaluate it on the 3DPW test set with the PA-MPJPE and MPJPE metric.

4.2. Implementation Details

Details of discriminators. For the anchor discriminator, we split 3D keypoints into five parts following [18], pass them through each of the five 4-layer residual MLPs with LeakyReLU, and finally concatenate the output. We use RMSProp optimizer for the anchor discriminators.

Details of generators. We sequentially build three 3-layer residual MLPs with LeakyReLU to generate bone angle, bone length, and rotation. Each network receives the output of the previous network as well as noise vector as input. These three networks are trained jointly by weighted adversarial losses. For every 6 iterations, we train the generator once and the discriminator 5 times.

Details of pose lifting network. We use VideoPose3D [41] (1-frame) as pose lifting network as well as other strong baselines [4, 37, 63] in ablation study. The pre-trained weights from the source dataset are used as initial weights for all the experiments.

Details of training. Starting with pre-trained weight, it takes 5 to 10 epochs (50 secs per epoch) to get convergence. In the experiments, we also found that WGAN not only has more stable training than LSGAN, but also prevents mode collapse, but the performance is not much different. We only train the generator and discriminator at the first 5 epochs for stable augmentation. Then, we mix the augmented data and original data in a ratio of 1:1 for pose lifting network training. The learning rate for all the networks is $1e^{-4}$. PoseDA is trained on NVIDIA RTX 3090. The training process takes two hours and consumes 2GB of GPU memory.

4.3. Results

Results on H3.6M. We compare PoseDA with state-of-the-art methods [41, 31, 18, 16] using ground truth 2D keypoints as input. The results on H3.6M are shown in Table 1, focusing on the generalization ability of the model over actions, since the distributions over global positions are relatively consistent between the different parts of the same dataset. Even though AdaptPose [16] utilizes temporal information, our method still achieves state-of-the-art performance.

Results on 3DHP. Our method achieves remarkable performance in all the metrics. We use single frame ground truth 2D keypoints as input and therefore compare against various recent methods with the same setting. PoseDA improves upon the previous state-of-the-art method by 10.2%, and is even competitively compared with some state-of-the-art fully supervised method.

Results on 3DPW [50]. We use ground truth 2D keypoints as input. PoseDA achieves the state-of-the-art performance without using any 3D annotations in 3DPW, as shown in Table 3, even is favorably compared with video-based methods.

Qualitative evaluation. Figure 4 shows the qualitative evaluation on 3DHP dataset. Compared with the baseline model without training with PoseDA, our method performs well for unusual human positions and challenging poses.
4.4. Ablation Study

Analysis on each components and board lifting network.

Since PoseDA is a data augmentation framework, with any pose lifting network architecture. Shown in Table 5, we conduct experiments on several strong baselines with different architectures, including MLP [37], Convolutional Network [41], and Graph Convolutional Network [4, 63]. We also check the effectiveness of each component in PoseDA. For fair comparison, we use the same weights of generator and discriminators in all the experiments. It shows that both global position alignment (GPA) and local pose augmentation (LPA) benefit the adaptive performance. Moreover, GPA significantly boosts performance without any extra training or use of additional learnable parameters.

Analysis on global pose alignment (GPA). We argue that the crop and normalization on 2D poses is an inaccurate method compared to our GPA module. Because the projection relation is not a linear operator and therefore does not have translation invariance. The crop out loses information about the relative positions of the camera and the person, and when the same 2d pose is cropped in a different position in the picture, the corresponding 3d pose is different (even if it is root-relative), there will be a one-to-many situation, which still increases ambiguity. We conduct experiments under four different pre-processing settings and with two backbones, our GPA based on screen normalized. We train the model on Human3.6M and test on 3DHP with the same pre-processing. Figure 5 shows different normalization operations.

4.5. Discussion Compared to SOTA Methods

The most related SOTA works are PoseAug and AdaptPose. Our proposed GPA has significant differences from the two methods in several aspects. In addition, our design of the discriminator is also different from these two methods. PoseAug is a simple domain generalization framework.

<table>
<thead>
<tr>
<th>Method</th>
<th>CD</th>
<th>MPJPE ((\downarrow))</th>
<th>PCK ((\uparrow))</th>
<th>AUC ((\uparrow))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi Person [39]</td>
<td></td>
<td>122.2</td>
<td>75.2</td>
<td>37.8</td>
</tr>
<tr>
<td>Mehta et al. [38]</td>
<td></td>
<td>117.6</td>
<td>76.5</td>
<td>40.8</td>
</tr>
<tr>
<td>VNect [40]</td>
<td></td>
<td>124.7</td>
<td>76.6</td>
<td>40.4</td>
</tr>
<tr>
<td>OriNet [35]</td>
<td></td>
<td>89.4</td>
<td>81.8</td>
<td>45.2</td>
</tr>
<tr>
<td>SimpleBaseline [37]</td>
<td></td>
<td>84.3</td>
<td>85.0</td>
<td>52.0</td>
</tr>
<tr>
<td>MixSTE [61]</td>
<td></td>
<td>57.9</td>
<td>94.2</td>
<td>63.8</td>
</tr>
<tr>
<td>LCN [10]</td>
<td>✓</td>
<td>-</td>
<td>74.0</td>
<td>36.7</td>
</tr>
<tr>
<td>HMR [27]</td>
<td>✓</td>
<td>113.2</td>
<td>77.1</td>
<td>40.7</td>
</tr>
<tr>
<td>SRNet [60]</td>
<td>✓</td>
<td>-</td>
<td>77.6</td>
<td>43.8</td>
</tr>
<tr>
<td>Li et al. [31]</td>
<td>✓</td>
<td>99.7</td>
<td>81.2</td>
<td>46.1</td>
</tr>
<tr>
<td>RepNet [52]</td>
<td>✓</td>
<td>92.5</td>
<td>81.8</td>
<td>54.8</td>
</tr>
<tr>
<td>PoseAug [18]</td>
<td>✓</td>
<td>73.0</td>
<td>88.6</td>
<td>57.3</td>
</tr>
<tr>
<td>AdapPose [16]</td>
<td>✓</td>
<td>68.3</td>
<td>90.2</td>
<td>59.0</td>
</tr>
<tr>
<td><strong>PoseDA (Ours)</strong></td>
<td>✓</td>
<td><strong>61.3</strong></td>
<td><strong>92.1</strong></td>
<td><strong>62.5</strong></td>
</tr>
</tbody>
</table>

Table 2. Results on 3DPW. CD denotes cross-domain evaluation (no CD denotes fully-supervision, i.e., trained and tested on the same 3DPW dataset). PCK, AUC and MPJPE are used as evaluation metrics. Source: H3.6M. Target: 3DPW.

<table>
<thead>
<tr>
<th>Method</th>
<th>CD</th>
<th>PA-MPJPE ((\downarrow))</th>
<th>MPJPE ((\downarrow))</th>
</tr>
</thead>
<tbody>
<tr>
<td>VideoPose3D [41]</td>
<td>✓</td>
<td>68.0</td>
<td>105.0</td>
</tr>
<tr>
<td>EFT [26]</td>
<td>✓</td>
<td>55.7</td>
<td>-</td>
</tr>
<tr>
<td>VIBE [28]</td>
<td>✓</td>
<td>51.9</td>
<td>82.9</td>
</tr>
<tr>
<td>Lin et al. [32]</td>
<td>✓</td>
<td>45.6</td>
<td>74.7</td>
</tr>
<tr>
<td>PoseAug [18]</td>
<td>✓</td>
<td>58.3</td>
<td>94.1</td>
</tr>
<tr>
<td>VIBE [28]</td>
<td>✓</td>
<td>56.5</td>
<td>93.5</td>
</tr>
<tr>
<td>BOA [20]</td>
<td>✓</td>
<td>49.5</td>
<td>77.2</td>
</tr>
<tr>
<td>AdapPose [16]</td>
<td>✓</td>
<td>46.5</td>
<td>81.2</td>
</tr>
<tr>
<td><strong>PoseDA (Ours)</strong></td>
<td>✓</td>
<td><strong>55.3</strong></td>
<td><strong>87.7</strong></td>
</tr>
</tbody>
</table>

Table 3. Results on 3DPW. CD denotes cross-domain evaluation (no CD denotes fully-supervision, i.e., trained and tested on the same 3DPW dataset), V denotes video-based method. PA-MPJPE and MPJPE are used as evaluation metrics. Source: H3.6M. Target: 3DPW.

<table>
<thead>
<tr>
<th>Method</th>
<th>CD</th>
<th>MPJPE ((\downarrow))</th>
<th>PCK ((\uparrow))</th>
<th>AUC ((\uparrow))</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleBaseline [37]</td>
<td></td>
<td>81.2</td>
<td>85.85</td>
<td>53.95</td>
</tr>
<tr>
<td>+ LPA</td>
<td></td>
<td>66.56 (‐16.7)</td>
<td>90.16 (‐4.3)</td>
<td>60.41 (‐6.5)</td>
</tr>
<tr>
<td>+ GPA</td>
<td></td>
<td>69.19 (‐12.0)</td>
<td>89.90 (‐4.1)</td>
<td>58.50 (‐4.6)</td>
</tr>
<tr>
<td>+ PoseDA</td>
<td></td>
<td><strong>64.79</strong> (‐16.4)</td>
<td><strong>90.55</strong> (‐4.7)</td>
<td><strong>61.32</strong> (‐7.4)</td>
</tr>
<tr>
<td>+ LPA</td>
<td></td>
<td>74.31 (‐6.8)</td>
<td>88.72 (‐2.9)</td>
<td>56.20 (‐2.4)</td>
</tr>
<tr>
<td>+ GPA</td>
<td></td>
<td>74.41 (‐6.8)</td>
<td>88.58 (‐2.7)</td>
<td>55.52 (‐1.7)</td>
</tr>
<tr>
<td>+ PoseDA</td>
<td></td>
<td><strong>69.50</strong> (‐11.7)</td>
<td><strong>90.15</strong> (‐4.2)</td>
<td><strong>58.56</strong> (‐4.8)</td>
</tr>
<tr>
<td>VideoPose3D [41]</td>
<td></td>
<td>82.55</td>
<td>85.71</td>
<td>53.35</td>
</tr>
<tr>
<td>+ LPA</td>
<td></td>
<td>66.65 (‐15.9)</td>
<td>90.05 (‐4.3)</td>
<td>60.24 (‐6.9)</td>
</tr>
<tr>
<td>+ GPA</td>
<td></td>
<td>66.07 (‐16.5)</td>
<td>90.87 (‐5.2)</td>
<td>60.07 (‐6.7)</td>
</tr>
<tr>
<td>+ PoseDA</td>
<td></td>
<td><strong>61.36</strong> (‐21.2)</td>
<td><strong>92.05</strong> (‐6.3)</td>
<td><strong>62.52</strong> (‐9.2)</td>
</tr>
</tbody>
</table>

Table 4. Ablation study on components and pose lifting network of our method. LPA denotes local pose augmentation, GPA denotes global position alignment. PoseDA combine GPA and LPA in a unified framework. Source: H3.6M. Target: 3DPH.

Figure 5. Illustration of different types of normalization operations.
that uses two discriminators, called $D_{3D}$ and $D_{2D}$. Their goal is to regulate the poses generated by the generator, making them similar to the 3D and 2D poses of the source domain. AdaptPose is a domain adaptation framework that also uses these two discriminators, but the goal of $D_{2D}$ is to make the generated poses more similar to the 2D poses of the target domain. However, the assumption of AdaptPose is not supported after careful and comprehensive experiments. In the human pose domain adaptation problem, we abandon $D_{2D}$ and only use $D_{3D}$ to regulate the quality of the generated poses to achieve better results. The reason we give is that: 1) Compared to PoseAug, making the generated 2D poses unnecessary similar to the source domain brings greater diversity. 2) Compared to AdaptPose, we believe that forcing the generated 2D poses to be similar to the target domain does not guarantee that the corresponding 3D poses are also similar. In other words, this is also because the 2D-3D mapping has large ambiguity. As a result, with our carefully designed GPA and LPA modules, global adaptation and local generalization are well combined, which is also our major contribution.

We conduct extensive and convincing experiments on the input selection of local pose augmentation module. All the experiments in this section are based on pre-trained VideoPose3D and global position alignment. Following [16, 18], we design a 2D pose discriminator of 5-layer MLP. Note that the condition of generator is used only to generate transformation, we still apply those transformation on the 3D pose from the source dataset. The 3D pose of target dataset used in 3D discriminator is the prediction of corresponding 2D pose by the pre-trained lifting network. As shown in Table 7, these experiments lead to two important conclusions: 1) the design of 2D pose discriminator is unnecessary, and 2) there is a performance drop no matter either the target domain information is involved in the generator or the discriminator to adapt the characteristics of local pose.

### 5. Conclusion

This paper addresses the problem of unsupervised cross-domain adaptation for 3D human pose estimation. To reduce the domain gap, we propose global position alignment and local pose augmentation. We argue that global position alignment is simple yet effective, and the local pose augmentation enhances the diversity of 2D-3D pose mapping. The proposed global position alignment module significantly boosts performance with no additional learnable parameters needed. We also show that adversarial pose augmentation based on Wasserstein distance can further obtain stable, diverse, and high-quality pose pairs. With extensive and convincing experiments and ablation studies, PoseDA can be flexibly applied on any 2D-3D pose lifting network and make a significant step towards solving domain adaptation problems for 3D human pose estimation.

### Limitations and Future Work

Although we show that generalization performs better than adaptation on domain gap over local pose, it is still possible to design a method that can adapt 3D pose without hurting the diversity of 2D-3D mappings.

### Acknowledgement

This work is supported by the National Key R&D Program of China No.2022ZD0162000, and National Natural Science Foundation of China No.62106219.

---

**Table 5. Ablation study on Global Position Alignment over other normalization techniques. Source: H3.6M. Target: 3DHP.**

<table>
<thead>
<tr>
<th>Method</th>
<th>MPJPE (↓)</th>
<th>PCK (↑)</th>
<th>AUC (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No GPA</td>
<td>66.07</td>
<td>90.87</td>
<td>60.07</td>
</tr>
<tr>
<td>RR</td>
<td>66.88</td>
<td>90.77</td>
<td>59.31</td>
</tr>
<tr>
<td>BL</td>
<td>65.81</td>
<td>90.94</td>
<td>59.91</td>
</tr>
<tr>
<td>RR + BL</td>
<td>64.99</td>
<td>91.07</td>
<td>60.49</td>
</tr>
<tr>
<td>PoseDA</td>
<td>61.36</td>
<td>92.05</td>
<td>62.52</td>
</tr>
</tbody>
</table>

**Table 6. Ablation study on local pose augmentation with different pre-defined pose augmentation methods. RR denotes random rotation along vertical axial and BL denotes random bone length transformation. Source: H3.6M. Target: 3DHP.**

<table>
<thead>
<tr>
<th>Method</th>
<th>MPJPE (↓)</th>
<th>PCK (↑)</th>
<th>AUC (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Aug</td>
<td>66.07</td>
<td>90.87</td>
<td>60.07</td>
</tr>
<tr>
<td>RR</td>
<td>66.88</td>
<td>90.77</td>
<td>59.31</td>
</tr>
<tr>
<td>BL</td>
<td>65.81</td>
<td>90.94</td>
<td>59.91</td>
</tr>
<tr>
<td>RR + BL</td>
<td>64.99</td>
<td>91.07</td>
<td>60.49</td>
</tr>
<tr>
<td>PoseDA</td>
<td>61.36</td>
<td>92.05</td>
<td>62.52</td>
</tr>
</tbody>
</table>

**Table 7. The input selection of generator condition $G_{cond}$. 3D pose discriminator $D_{3D}$, and 2D pose discriminator $D_{2D}$ in LPA. $S, T$ denote the pose from source or target domain. Source: H3.6M. Target: 3DHP.**

<table>
<thead>
<tr>
<th>$G_{cond}$</th>
<th>$D_{3D}$</th>
<th>$D_{2D}$</th>
<th>MPJPE (↓)</th>
<th>PCK (↑)</th>
<th>AUC (↑)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$T$</td>
<td>$T$</td>
<td>66.07</td>
<td>90.87</td>
<td>60.07</td>
</tr>
<tr>
<td>$T$</td>
<td>$S$</td>
<td>$T$</td>
<td>66.34</td>
<td>90.81</td>
<td>59.72</td>
</tr>
<tr>
<td>$T$</td>
<td>$S$</td>
<td>$-$</td>
<td>65.91</td>
<td>91.02</td>
<td>59.92</td>
</tr>
<tr>
<td>$T$</td>
<td>$S$</td>
<td>$-$</td>
<td>65.37</td>
<td>90.98</td>
<td>60.30</td>
</tr>
<tr>
<td>$S$</td>
<td>$T$</td>
<td>$T$</td>
<td>64.20</td>
<td>91.64</td>
<td>60.85</td>
</tr>
<tr>
<td>$S$</td>
<td>$T$</td>
<td>$-$</td>
<td>63.66</td>
<td>91.57</td>
<td>61.34</td>
</tr>
<tr>
<td>$S$</td>
<td>$S$</td>
<td>$S$</td>
<td>73.55</td>
<td>88.96</td>
<td>56.41</td>
</tr>
<tr>
<td>$S$</td>
<td>$S$</td>
<td>$T$</td>
<td>65.46</td>
<td>91.27</td>
<td>60.03</td>
</tr>
<tr>
<td>$S$</td>
<td>$S$</td>
<td>$-$</td>
<td>61.36</td>
<td>92.05</td>
<td>62.52</td>
</tr>
</tbody>
</table>
References


[24] Jingxia Jiang, Tian Ye, Jinbin Bai, Sixiang Chen, Wenhao Chai, Shi Jun, Yun Liu, and Erkang Chen. Five a’s (+) net-


