Cascaded Deep Monocular 3D Human Pose Estimation with Evolutionary Training Data

Shichao Li¹, Lei Ke¹, Kevin Pratama¹, Yu-Wing Tai², Chi-Keung Tang¹, Kwang-Ting Cheng¹
¹The Hong Kong University of Science and Technology, ²Tencent

Abstract

End-to-end deep representation learning has achieved remarkable accuracy for monocular 3D human pose estimation, yet these models may fail for unseen poses with limited and fixed training data. This paper proposes a novel data augmentation method that: (1) is scalable for synthesizing massive amount of training data (over 8 million valid 3D human poses with corresponding 2D projections) for training 2D-to-3D networks, (2) can effectively reduce dataset bias. Our method evolves a limited dataset to synthesize unseen 3D human skeletons based on a hierarchical human representation and heuristics inspired by prior knowledge. Extensive experiments show that our approach not only achieves state-of-the-art accuracy on the largest public benchmark, but also generalizes significantly better to unseen and rare poses. Relevant files and tools are available at the project website

1. Introduction

Estimating 3D human pose from RGB images is critical for applications such as action recognition [32] and human-computer interaction, yet it is challenging due to lack of depth information and large variation in human poses, camera viewpoints and appearances. Since the introduction of large-scale motion capture (MC) datasets [56, 20], learning-based methods and especially deep representation learning have gained increasing momentum in 3D pose estimation. Thanks to their representation learning power, deep models have achieved unprecedented high accuracy [43, 41, 28, 34, 32, 60].

Despite their success, deep models are data-hungry and vulnerable to the limitation of data collection. This problem is more severe for 3D pose estimation due to two factors. First, collecting accurate 3D pose annotation for RGB images is expensive and time-consuming. Second, the collected training data is usually biased towards indoor environment and selected daily actions. Deep models can easily exploit these bias but fail for unseen cases in unconstrained environments. This fact has been validated by recent works [69, 66, 25, 64] where cross-dataset inference demonstrated poor generalization of models trained with biased data.

To cope with the domain shift of appearance for 3D pose estimation, recent state-of-the-art (SOTA) deep models adopt the two-stage architecture [68, 13, 14]. The first stage locates 2D human key-points from appearance information, while the second stage lifts the 2D joints into 3D skeleton employing geometric information. Since 2D pose annotations are easier to obtain, extra in-the-wild images can be used to train the first stage model, which effectively reduces the bias towards indoor images during data collection. However, the second stage 2D-to-3D model can still be negatively influenced by geometric data bias, yet not studied before. We focus on this problem in this work and our research questions are: are our 2D-to-3D deep networks influenced by data bias? If yes, how can we improve network
Our contributions are summarized as follows:

To answer these questions, we propose to analyze the training data with a hierarchical human model and represent human posture as collection of local bone orientations. We then propose a novel dataset evolution framework to cope with the limitation of training data. Without any extra annotation, we define evolutionary operators such as crossover and mutation to discover novel valid 3D skeletons in tree-structured data space guided by simple prior knowledge. These synthetic skeletons are projected to 2D and form 2D-3D pairs to augment the data used for training 2D-to-3D networks. With an augmented training dataset after evolution, we propose a cascaded model achieving state-of-the-art accuracy under various evaluation settings. Finally, we release a new dataset for unconstrained humans in-the-wild. Our contributions are summarized as follows:

- To our best knowledge, we are the first to improve 2D-to-3D network training with synthetic paired supervision.
- We propose a novel data evolution strategy which can augments an existing dataset by exploring 3D human pose space without intensive collection of extra data. This approach is scalable to produce 2D-3D pairs in the order of 10^7, leading to better model generalization power in unseen scenarios.
- We present TAG-Net, a deep architecture consisting of an accurate 2D joint detector and a novel cascaded 2D-to-3D network. It out-performs previous monocular models on the largest 3D human pose estimation benchmark in various aspects.
- We release a new labeled dataset for unconstrained human pose estimation in-the-wild.

Fig. 1 shows our model trained on an augmented dataset can handle rare poses while others such as [25] may fail.

2. Related Works

Monocular 3D human pose estimation. Single-image 3D pose estimation methods are conventionally categorized into generative methods and discriminative methods. Generative methods fit parametrized models to image observations for 3D pose estimation. These approaches represent humans by PCA models [2, 70], graphical models [8, 5] or deformable meshes [4, 30, 7, 42, 24]. The fitting process amounts to non-linear optimization, which requires good initialization and refines the solution iteratively. Discriminative methods [53, 1, 6] directly learn a mapping from image observations to 3D poses. Pertinent and recent deep neural networks (DNNs) employ two mainstream architectures: one-stage methods [66, 69, 32, 43, 41, 28, 60, 16] and two-stage methods [39, 34, 47, 68]. The former directly map from pixel intensities to 3D poses, while the latter first extract intermediate geometric representation such as 2D key-points and then lift them to 3D poses.

We adopt the discriminative approach and focus on the second stage. Instead of using a fixed training dataset, we evolve the training data to improve the performance of the 2D-to-3D network.

Weakly-supervised 3D pose estimation. Supervised training of DNNs demands massive data while 3D annotation is difficult. To address this problem, weakly-supervised methods explore other potential supervision to improve network performance when only few training data is available [44, 49, 50, 23, 11, 65, 27]. Multi-view consistency [44, 49, 50, 23, 11] is proposed and validated as useful supervisory signal when training data is scarce, yet a minimum of two views are needed. In contrast, we focus on effective utilization of scarce training data by synthesizing new data from existing ones and uses only single view.

Data augmentation for pose estimation. New images can be synthesized to augment indoor training dataset [51, 63]. In [63] new images were rendered using MC data and human models. Domain adaption was performed in [10] during training with synthetic images. Adversarial rotation and scaling were used in [46] to augment data for 2D pose estimation. These works produce augmented images while we focus on data augmentation for 2D-to-3D networks and produce geometric 2D-3D pairs.

Pose estimation dataset. Most large-scale human pose estimation datasets [67, 29, 3] only provide 2D pose annotations. Accurate 3D annotations [20, 56] require MC devices and these datasets are biased due to the limitation of data collection process. Deep models are prone to overfit to these biased dataset [61, 62, 26], failing to generalize in unseen situations. Our method can synthesize for free without human annotation large amount of valid 3D poses with more complete coverage in human pose space.

3. Dataset Evolution

From a given input image \( x_i \) containing one human subject, we aim to infer the 3D human pose \( \hat{p}_i \) given the image observation \( \phi(x_i) \). To encode geometric information as other 2D-to-3D approaches [34, 68, 25], we represent \( \phi(x) \) as the 2D coordinates of \( k \) human key-points \( (x_i, y_i)_{k=1}^k \) on the image plane. As a discriminative approach, we seek a regression function \( F(\phi(x_i), \Theta) \) that outputs 3D pose \( \hat{p}_i \) as \( \hat{p}_i = F(\phi(x_i), \Theta) \). This regression function is implemented as a DNN parametrized by \( \Theta \). Conventionally this DNN is trained on a dataset collected by MC devices [56, 20]. This dataset consists of paired images and 3D pose ground truths \( \{(x_i, p_i)\}_{i=1}^N \) and the DNN can be trained by gradient descent based on a loss function defined over the training dataset \( L = \sum_{i=1}^N E(p_i, \hat{p}_i) \).
where $E$ is the error measurement between the ground truth $p_i$ and the prediction $\hat{p}_i = F(\phi(x_i), \Theta)$.

Unfortunately, sampling bias exists during the data collection and limits the variation of the training data. Human 3.6M (H36M) [20], the largest MC dataset, only contains 11 subjects performing 15 actions under 4 viewpoints, leading to insufficient coverage of the training 2D-3D pairs $(\phi(x_i), p_i)$. A DNN can overfit to the dataset bias and become less robust to unseen $\phi(x)$. For example, when a subject starts street dancing, the DNN may fail since it is only trained on daily activities such as sitting and walking. This problem is even exacerbated for the weakly-supervised methods [44, 50, 11] where only a subset of training data is used to simulate the data scarcity scenario.

We take a non-stationary view toward the training data to cope with this problem. While conventionally the collected training data is fixed and the trained DNN is not modified during its deployment, here we assume the data and model can evolve during their life-time. Specifically, we synthesize novel 2D-3D pairs based on an initial training dataset and add them into the original dataset to form the evolved dataset. We then re-train the model with the evolved dataset. As shown in Fig. 2, model re-trained on the evolved dataset has consistently lower generalization error, comparing to a model trained on the initial dataset.

In the following we show that by using a hierarchical representation of human skeleton, the synthesis of novel 2D-3D pairs can be achieved by evolutionary operators and camera projection.

3.1. Hierarchical Human Representation

We represent a 3D human skeleton by a set of bones organized hierarchically in a kinematic tree as shown in Figure 3. This representation captures the dependence of adjacent joints using tree edges.

![Figure 3: Hierarchical human representation. Left: 3D keypoints organized in a kinematic tree where red arrows point from parent joints to children joints. Right: Zoom-in view of a local coordinate system.](image)

![Figure 4: Examples of applying evolution operators. Crossover and mutation take 2 and 1 random samples respectively to synthesize novel human skeletons.](image)

Each 3D pose $p$ corresponds to a set of bone vectors $\{b^1, b^2, \cdots, b^w\}$ and a bone vector is defined as

$$b^i = p^{child(i)} - p^{parent(i)}$$  \hfill (1)

where $p^j$ is the $j$th joint in the 3D skeleton and $parent(i)$ gives the parent joint index of the $i$th bone vector. A local coordinate system is attached at each parent node. For a parent node $p^{parent(i)}$, its local coordinate system is represented by the rotation matrix defined by three basis vectors $R^i = [i^i, j^i, k^i]$. The global bone vector is transformed into this local coordinate system as

$$b^i_{\text{local}} = R^i_T b^i_{\text{global}} = R^i_T (p^{child(i)} - p^{parent(i)})$$ \hfill (2)

For convenience, this local bone vector is further converted into spherical coordinates $b^i_{\text{local}} = (r_i, \theta_i, \phi_i)$. The posture of the skeleton can be described by the collection of bone orientations $\{(\theta_i, \phi_i)\}_{i=1}^w$ while the skeleton size is encoded into $\{r_i\}_{i=1}^w$.

---

3The coordinate system is detailed in our supplementary material.
3.2. Synthesizing New 2D-3D Pairs

We first synthesize new 3D skeletons \( D_{new} = \{ \mathbf{p}_j \}_{j=1}^M \) with an initial training dataset \( D_{old} = \{ \mathbf{p}_j \}_{j=1}^N \) and project 3D skeletons to 2D given camera intrinsics \( K \) to form 2D-3D pairs \( \phi(\mathbf{x}_j, \mathbf{p}_j) \) where \( \phi(\mathbf{x}_j) = K \mathbf{p}_j \).

When adopting the hierarchical representation, a dataset of articulated 3D objects is a population of tree-structured data in nature. Evolutionary operators [18] have constructive property [57] that can be used to synthesize new data [15] given an initial population. The design of operators is problem-dependent and our operators are detailed as follows.

**Crossover Operator** Given two parent 3D skeletons, crossover is defined as a random exchange of sub-trees. This definition is inspired by the observation that an unseen 3D pose might be obtained by assembling limbs from known poses. Formally, we denote the set of bone vectors for parent \( A \) and \( B \) as \( S_A = \{ \mathbf{b}_1^A, \mathbf{b}_2^A, \ldots, \mathbf{b}_w^A \} \) and \( S_B = \{ \mathbf{b}_1^B, \mathbf{b}_2^B, \ldots, \mathbf{b}_w^B \} \). A joint indexed by \( q \) is selected at random and the bones rooted at it are located for the two parents. These bones form the chosen sub-tree set \( S_{chosen} \)

\[
\{ \mathbf{b}_j : parent(j) = q \lor IsOff(parent(j), q) \} \tag{3}
\]

where \( IsOff(parent(j), q) \) is True if joint \( parent(j) \) is an offspring of joint \( q \) in the kinematic tree. The parent bones are split into the chosen and the remaining ones as \( S_X = S_{chosen}^X \cup S_{rem}^X \) where \( S_{chosen}^X = S_X - S_{chosen} \) and \( X = A \) or \( B \). Now the crossover operator gives two sets of children bones as

\[
S_C = S_{chosen}^A \cup S_{rem}^B \quad \text{and} \quad S_D = S_{chosen}^B \cup S_{rem}^A \tag{4}
\]

These two new sets are converted into two new 3D skeletons. The example in Fig. 4 shows the exchange of the right arms when the right shoulder joint is selected.

**Algorithm 1 Data evolution**

**Input:**
Initial set of 3D skeletons \( D_{old} = \{ \mathbf{p}_j \}_{j=1}^N \), noise level \( \sigma \), number of generations \( G \)

**Output:** Augmented set of skeletons \( D_{new} = \{ \mathbf{p}_j \}_{j=1}^M \)

1: \( D_{new} = D_{old} \)
2: for \( i=1:G \) do
3: \( \text{Parents} = \text{Sample}(D_{new}) \)
4: \( \text{Children} = \text{NaturalSelection} \left( \text{Mutation} \left( \text{Crossover} \left( \text{Parents} \right) \right) \right) \)
5: \( D_{new} = D_{new} \cup \text{Children} \)
6: end for
7: return \( D_{new} \)

**Mutation Operator** As the motion of human limbs is usually continuous, a perturbation of one limb of an old 3D skeleton may result in a valid new 3D pose. To implement this perturbation, our mutation operator modifies the local orientation of one bone vector to get a new pose. One bone vector \( \mathbf{b}_i = (r_i, \theta_i, \phi_i) \) for an input 3D pose is selected at random and its orientation is mutated by adding noise (Gaussian in this study):

\[
\theta_i' = \theta_i + g, \phi_i' = \phi_i + g \tag{5}
\]

where \( g \sim N(0, \sigma_{local}) \) and \( \sigma_{local} \) is a pre-defined noise level. One example of mutating the left leg is shown in Fig. 4. We also mutate the global orientation and bone length of the 3D skeletons to reduce the data bias of viewpoints and subject sizes, which is detailed in our supplementary material.

**Natural Selection** We use a fitness function to evaluate the goodness of synthesized data for selection as \( v(\mathbf{p}) \) which indicates the validity of the new pose. \( v(\mathbf{p}) \) can be any function that describes how anatomically valid a skeleton is, and we implement it by utilizing the binary function provided by [2]. We specify \( v(\mathbf{p}) = -\infty \) if \( \mathbf{p} \) is not valid to rule out all invalid poses.
Evolution Process The above operators are applied to $D_{old}$ to obtain a new generation $D_{new}$ by synthesizing new poses and merge with the old poses. This evolution process repeats several generations and is depicted in Algorithm 1. Finally, $D_{new}$ are projected to 2D key-points to obtain paired 2D-3D supervision.

4. Model Architecture

We propose a two-stage model as shown in Fig. 5. We name it TAG-Net, as the model’s focus transits from appearance to geometry. This model can be represented as a function

$$\hat{p} = TAG(x) = G(A(x))$$

(6)

Given an input RGB image $x$, $A(x)$ (the appearance stage) regresses $k = 17$ high-resolution probability heat-maps $H^k_{i=1}$ for $k$ 2D human key-points and map them into 2D coordinates $c = (x_i, y_i)_{i=1}^k, G(c)$ (the geometry stage) infers 3D key-point coordinates in the camera coordinate system from input 2D coordinates. Key designs are detailed as follows.

4.1. High-resolution Heatmap Regression

Synthesized 2D key-points are projected from 3D points and can be thought as perfect detections while real detections produced by heat-map regression models are noiser. We hope this noise can be as small as possible since we need to merge these two types of data as described in Section 3. To achieve this goal, we use HR-Net [59] as our backbone for image feature extraction. While the original model predicts heat-maps of size 96 by 72, we append a pixel shuffle layer [55] to the end and regress heat-maps of size 384 by 288. The original model uses hard arg-max to predict 2D coordinates, which results in rounding errors in our experiments. Instead, we use soft arg-max [40, 60] to obtain 2D coordinates. The average 2D key-point localization errors for H36M testing images are shown in Table 1. Our design choice improves the previous best model and achieves the highest key-point localization accuracy on H36M to date. The extensions add negligible amount of parameters and computation.

4.2. Cascaded Deep 3D Coordinate Regression

Since the mapping from 2D coordinates to 3D joints can be highly-nonlinear and difficult to learn, we propose a cascaded 3D coordinate regression model as

$$\hat{p} = G(c) = \sum_{t=1}^T D_t(i_t, \Theta_t)$$

(7)

where $D_t$ is the $t$th deep learner in the cascade parametrized by $\Theta_t$ whose input is $i_t$. As shown in the top of Fig. 5, the first learner $D_1$ in the cascade directly predicts 3D pose while the later ones predict the 3D refinement $\delta p = (\delta x_i, \delta y_i, \delta z_i)_{i=1}^k$. While cascaded coordinate regression has been adopted for 2D key-points localization [9, 48], hand-crafted image feature and classical weak learners such as linear regressors were used. In contrast, our geometric model $G(c)$ only uses coordinates as input and each learner is a DNN with residual connections [17].

The bottom of Fig. 5 shows the detail for each deep learner. One deep learner first maps the input 2D coordinates into a representation vector of dimension $d = 1024$, after which $R = 3$ residual blocks are used. Finally the representation is mapped by a fully-connected (FC) layer into 3D coordinates. After each FC layer we add batch normalization [19] and dropout [58] with dropout rate 0.5. The capacity of each deep learner can be controlled by $R$. This cascaded model is trained sequentially by gradient descent and the training algorithm is included in our supplementary material. Despite the number of parameters increase linearly with the cascade length, we found that the cascaded model is robust to over-fitting for this 3D coordinate prediction problem, which is also shared by the 2D counterparts [9, 48].

4.3. Implementation Details

We train $A(x)$ and $G(c)$ sequentially. The input size is 384 by 288 and our output heat-map has the same high resolution. The back-bone of $A(x)$ is pre-trained on COCO [29] and we fine-tune it on H36M with Adam optimizer using a batch size of 24. The training is performed on two NVIDIA Titan Xp GPUs and takes 8 hours for 18k iterations. We first train with learning rate 0.001 for 3k iterations, after which we multiply it by 0.1 after every 3k iterations. To train $G(c)$, we train each deep learner in the cascade using Adam optimizer with learning rate 0.001 for 200 epochs.

5. Experiments

To validate our data evolution framework and model architecture, we evolve from the training data provided in
H36M and conduct both intra- and cross-dataset evaluation. The camera intrinsics provided by H36M are used during data synthesis. We vary the size of initial population to demonstrate the effectiveness of synthetic data when the training data is scarce. Finally we present ablation study to analyze the influences of data augmentation and hyper-parameters.

5.1. Datasets and Evaluation Metrics

**Human 3.6M** (H36M) is the largest 3D human pose estimation benchmark with accurate 3D labels. We denote a collection of data by appending subject ID to S, e.g., S15 denotes data from subject 1 and 5. Previous works fix the training data while our method uses it as our initial population and evolves from it. We evaluate model performance with **Mean Per Joint Position Error** (MPJPE) measured in millimeters. Two standard evaluation protocols are adopted. **Protocol 1** (P1) directly computes MPJPE while **Protocol 2** (P2) aligns the ground-truth 3D poses with the predictions with a rigid transformation before calculating it. Protocol P1* uses ground truth 2D key-points as inputs and removes the influence of the first stage model.

**MPI-INF-3DHP** (3DHP) is a benchmark that we use to evaluate the generalization power of 2D-to-3D networks. We do not use its training data and conduct cross-dataset inference by feeding the provided key-points to $G(c)$. Apart from MPJPE, **Percentage of Correct Keypoints** (PCK) measures correctness of 3D joint predictions under a specified threshold, while **Area Under the Curve** (AUC) is computed for a range of PCK thresholds.

**Unconstrained 3D Poses in the Wild** (U3DPW) We collect by ourselves a new small dataset consisting of 300 challenging in-the-wild images with rare human poses, where 150 of them are selected from Leeds Sports Pose dataset [21]. The annotation process is detailed in our supplementary material. Similar to 3DHP, this dataset is used for validating model generalization for unseen 3D poses.

5.2. Comparison with state-of-the-art methods

**Comparison with weakly-supervised methods** Here we compare with weakly-supervised methods, which only use a small number of training data to simulate scarce data scenario. To be consistent with others, we utilize S1 as our initial population. While others fix S1 as the training dataset, we evolve from it to obtain an augmented training set. The comparison of model performance is shown in Table 2, where our model significantly out-performs others and demonstrates effective use of the limited training data. While other methods [50, 23] use multi-view consistency as extra supervision, we achieve comparable performance with only a single view by synthesizing useful supervision. Fig. 2 validates our method when the training data is extremely scarce, where we start with a small fraction of S1 and increase the data size by 2.5 times by evolution. Note that the model performs consistently better after dataset evolution. Compared to the temporal convolution model proposed in [45], we do not utilize any temporal information and achieve comparable performance. This indicates our approach can make better use of extremely limited data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Authors</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>Li et al. (ICCV’19) [27]</td>
<td>62.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>68.8</td>
</tr>
</tbody>
</table>

Table 2: Comparison with SOTA weakly-supervised methods. Average MPJPE over all 15 actions for H36M under two protocols (P1 and P2) is reported. P1* refers to protocol 1 evaluated with ground truth 2d key-points. Best performance is marked with bold font. Error for each action can be found in our supplementary material.

**Comparison with fully-supervised methods** Here we compare with fully-supervised methods that uses the whole training split of H36M. We use S15678 as our initial population and Table 3 shows the performance comparison. Under this setting, our model also achieves competitive performance compared with other SOTA methods, indicating that our approach is not limited to scarce data scenario.

<table>
<thead>
<tr>
<th>Method</th>
<th>Authors</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>Martinez et al. (ICCV’17) [34]</td>
<td>62.9</td>
</tr>
<tr>
<td></td>
<td>Yang et al. (CVPR’18) [66]</td>
<td>58.6</td>
</tr>
<tr>
<td></td>
<td>Zhao et al. (CVPR’19) [68]</td>
<td>57.9</td>
</tr>
<tr>
<td></td>
<td>Sharma et al. (ICCV’19) [54]</td>
<td>58.0</td>
</tr>
<tr>
<td></td>
<td>Moon et al. (ICCV’19) [38]</td>
<td>54.4</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>50.9</td>
</tr>
</tbody>
</table>

Table 3: Comparison with SOTA methods under fully-supervised setting. Same P1, P1* and P2 as in Table 2.

5.3. Cross-dataset Generalization

To validate the generalization ability of our 2D-to-3D network in unknown environment, Table 4 compares with other methods on 3DHP. In this experiment we evolve from S15678 in H36M to obtain an augmented dataset consisting of 8 million 2D-3D pairs. Without utilizing any training data of 3DHP, we achieve SOTA performance in this
benchmark. We obtain clear improvements comparing with [25], which also uses S15678 as the training data but fix it without data augmentation. The results indicate that our data augmentation approach improves model generalization effectively despite we start with the same biased training dataset. As shown in Fig. 6, the distribution of the augmented dataset indicates less dataset bias. Qualitative results on 3DHP and LSP are shown in Fig. 7. Note that these unconstrained poses are not well-represented in the original training dataset yet our model still gives good inference results. Qualitative comparison with [25] on some difficult poses in U3DPW is shown in Fig. 8 and our model shows better accuracy for these rare human poses.

Table 4: Testing results for the MPI-INF-3DHP dataset. A higher value is better for PCK and AUC while a lower value is better for MPJPE. MPJPE is evaluated without rigid transformation. CE denotes cross-dataset evaluation and the training data in MPI-INF-3DHP is not used.

Table 5: Ablation study on H36M. B: baseline. C: add cascade. E: add data evolution. Evolve() represents the data augmentation operation. Same P1 and P1* as in Table 2. Error reduction compared with the baseline follows the ↓ signs.

6. Conclusion

This paper presents a novel evolution framework to enrich the data distribution of an initial biased training set, leading to better intra-dataset and cross-dataset generalization of 2D-to-3D network. A novel monocular human pose estimation model is trained achieving state-of-the-art performance for single-frame 3D human pose estimation. There are a lot of fruitful directions remaining to be explored. First, extension to temporal domain, multi-view setting and multi-person scenarios are just three examples. Second, instead of using fixed evolution operators, we will investigate how the operators can also evolve during the data generation process.
Figure 7: Cross-dataset inferences of $G(c)$ on MPI-INF-3DHP (first row) and LSP (next two rows).

Figure 8: Cross-dataset inference results on U3DPW comparing with [25]. Video is included in our supplementary material.

Figure 9: Training and testing errors with varying number of cascade length and training data. The cascade effectively reduces training error and is robust to over-fitting.

Figure 10: MPJPE (P1*) before (BE) and after evolution (AE) with varying number of blocks $R$. Evolved training data can afford a deeper network. Best viewed in color.

Acknowledgments We gratefully acknowledge the support of NVIDIA Corporation with the donation of one Titan Xp GPU used for this research. This research is also supported in part by Tencent and the Research Grant Council of the Hong Kong SAR under grant no. 1620818.
References


