JointPose: Jointly Optimizing Evolutionary Data Augmentation and Prediction Neural Network for 3D Human Pose Estimation

Zhiwei Yuan and Songlin Du(✉)

Southeast University, Nanjing, China
sdu@seu.edu.cn

Abstract. 3D human pose estimation plays important roles in various human-machine interactive applications, but lacking diversity in existing labeled 3D human posture dataset restricts the generalization ability of deep learning based models. Data augmentation is therefore an important method to solve this problem. However, data augmentation and pose estimation network training are usually treated as two isolated processes, limiting the performance of pose estimation network. In this paper, we developed an improved data augmentation method which jointly performs pose network estimation and data augmentation by designing a reward/penalty strategy for effective joint training, making model training and data augmentation improve each other. In particular, an improved evolutionary data augmentation method is proposed to generate the distribution of nodes in crossover and rotation angles in mutation through the process of the evolution. Extensive experiments show that our approach not only significantly improves state-of-the-art models without additional data efforts but also is extremely competitive with other advanced methods.

Keywords: 3D human pose estimation · Evolutionary data augmentation · Joint neural network

1 Introduction

Human pose estimation (HPE) aims to restore the human body posture and build human body representation (such as, body skeleton and body shape) from input data such as images and videos. And 3D HPE has been applied to a wide range of applications, (e.g., motion recognition and analysis, human-computer interaction, virtual reality(VR), security identification, etc.). However, due to the limited information provided by a single image and the complexity and diversity of 3D human pose estimation, this task is extremely challenging. Thanks to their representation learning power, the deep learning method greatly improves the accuracy of the model [9,11,12,14,15], and makes deep learning based human pose estimation have a better prospect.

Despite such success, the training of deep learning model requires a large amount of labeled data hence the training data directly determines the upper limit of accuracy of the model. In particular, this is more severe for 3D HPE as during the process of obtaining the human body posture dataset, collecting accurate 3D pose annotation of the human dataset require a lot of manpower and time cost, and the collection of human

I. Farkaš et al. (Eds.): ICANN 2021, LNCS 12893, pp. 368–379, 2021.
https://doi.org/10.1007/978-3-030-86365-4_30

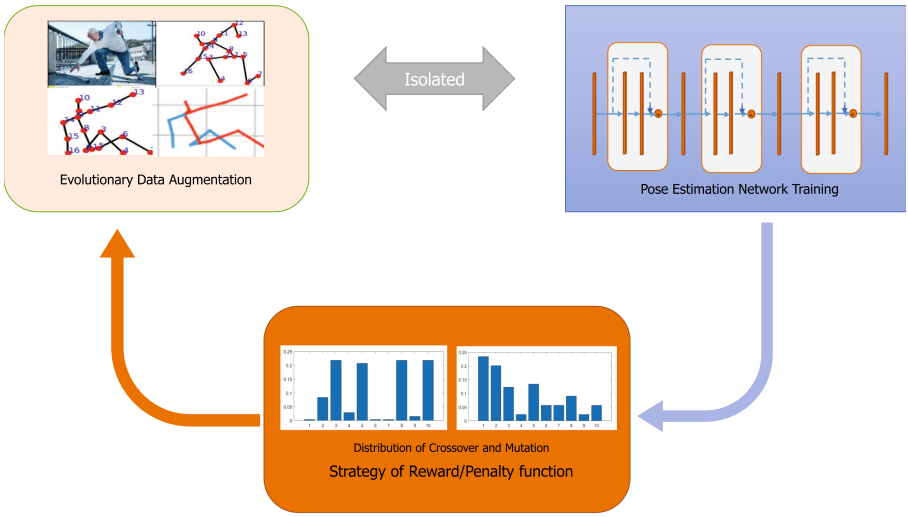

Fig. 1. Data augmentation and network training are usually isolated. We propose to bridge the two by designing a reward/penalty strategy online. In train epoch of 3D HPE model, we use the pre-trained model information (e.g., the value of loss function) to adjust the operation distribution of evolutionary data augmentation.

posture is under the fixed scene. Therefore, the problem of 3D human posture dataset has become a major bottleneck restricting the performance improvement of the depth model.

In order to solve the limitation caused by lack of labeled data, Li *et al.* [7] proposed a method to synthesize massive amount of 3D human skeletons with evolutionary computation. Human postures are first represented as tree-like structures, and new samples are then synthesised by crossover (exchanging two parts of two parental generations) and mutation (randomly rotating local bones). The synthesised evolutionary data is then used to train the pose estimation network, achieving the state-of-the-art performance. However, the data augmentation process and pose estimation network training are carried out separately. This paper investigates whether these two parts can be combined together to jointly train a more effective network.

In this paper, we answer the above question by proposing a new approach that jointly optimizes 3D HPE network and data augmentation by designing a reward/penalty strategy for effective joint training [17]. In original data augmentation, the operation parameters of crossover and mutation are randomly selected, which makes the effect of the algorithm very unsatisfactory. Given the priori knowledge of human pose, there is a natural beauty of symmetry between the joints of different individuals. The random exchange destroys this pattern, so we use the pre-trained information to generate cross-distribution in the evolutionary algorithm to minimize the impact. Similarly, there is a harmonious beauty to the length between the various parts of the human posture, and we use the distribution of variation rather than random parameters to maintain this pattern. Compared with original method, the pose synthesised by our approach is more diverse and efficient.

To realize this idea, we combined data augmentation with 3D HPE model training. In each train epoch of 3D HPE model, there is a pre-trained 3D HPE model, and the we used the pre-training model information to adjust the operation of data augmentation. In the original data augmentation, the human pose's cross nodes were randomly selected during the crossover operation, on the contrary the distribution of the cross nodes was generated through the performance of the pre-training process in our improved model. Similarly, the rotation of bone vectors was selected randomly during the mutation operation, while the distribution of bone vectors in our improved model was also generated using the performance of the pre-training process. Then the data obtained by the improved data augmentation method are trained in the 3D HPE network. In this way, we combine the data augmentation process with the pose estimation network training process. Figure 1 shows our approach jointly performs pose network estimation and data augmentation by designing a reward/penalty strategy. The main contributions of this paper are summarized as follows:

– To the best of our knowledge, we are the first to investigate the joint optimization of evolutionary data augmentation and network training in 3D human pose estimation.
– The pose estimation and data augmentation in network training are combined to jointly improve the performances of both data augmentation and model training.
– Strong performance on the 3D HPE network using the data synthesised by the improved evolutionary data augmentation method, which validate our method substantially.

2 Related Work

3D Human Pose Estimation: Using the identification method to estimate 3D human posture is a direct mapping from image observation to 3D pose. The related and latest depth 3D pose estimation networks mainly adopt two frameworks: the one-stage methods and the two-stage methods. The one-stage methods directly map from the image to 3D pose, while the two-stage methods [3,24] first extract the 2D pose from the image, and then establishes the mapping from 2D pose to 3D pose. In this paper, we take a two-stage approach. In the first stage, in the process of extracting 2D pose from images, the training dataset is not deficient compared with the 3D human pose data set, and the regression accuracy has been relatively ideal, so we directly adopted the model completed by pre-training. The focus of this paper is on the improved data augmentation in the second stage. New 2D-3D data pairs are synthesised by improving the method of evolutionary data augmentation. See Sect. 3 for details.

Human Pose Data Augmentation: There are many methods for data augmentation. For example, in [19,22], new indoor images can be synthesized to augment as to extend additional training dataset. During training with synthetic images, domain adaption was performed in [2]. Adversarial rotation and scaling were used in [17] to augment data for 2D HPE. These works produce augmented images, on the contrary, our approach focus on data augmentation for 2D-to-3D networks and produce the distribution of data augmentation operations for geometric 2D-3D pairs.

Hard Example Mining: The idea is widely used in training SVM models for object detection [21]. It aims to perform an alternative optimization between model training and data selection. Contrary to this idea, the proposed method focuses on mining the distribution P that synthesizes more efficient data in evolutionary data augmentation, rather than hard example for network training, in order to improve the evolutionary data augmentation algorithm. The reasons are as follows: 1) Considering the special structure of the human body, it is unreasonable to randomly select crossover and mutation nodes, so the probability distribution is used to select the operated nodes. 2) As a parameter of the evolutionary data augmentation algorithm, the improved probability distribution directly makes the synthesized data more efficient for network training.

3 Evolutionary Data Augmentation (EDA)

In this section, we use evolutionary algorithm for data augmentation, then combine the pose estimation network training process and data augmentation process and finally generate the distribution of augmentation operations, so as to propose an improved evolutionary data augmentation method.

3.1 3D Human Skeleton Representation

We represent a 3D human skeleton with a set of bones organized hierarchically in a kinematic tree, as shown in Fig. 2. For a given 3D human skeleton, we use a set of vectors $\{b_1, b_2, \cdots b_w\}$ to represent it, and the definition of skeleton vector is

$$b^k = p^{child\_node(k)} - p^{parent\_node(k)}, \tag{1}$$

where b^i is the ith bone in the 3D bone vector, and the direction of the ith bone vector is from the ith child node to the ith parent node. At the same time, for convenience, each skeleton vector is further transformed locally to spherical coordinate system, i.e.

$$b^k_{local} = \{r_k, \theta_k, \phi_k\}, \tag{2}$$

where $\{(\theta_k, \phi_k)\}^{\omega}_1$ represents the direction of the bone vector, and $\{(r_k)\}^{\omega}_1$ represents the length of the bone vector. Such a 3D representation of the human skeleton of a tree structure provides convenience for our data-augmentation evolutionary manipulation. In the data evolution operation, the representation of skeleton vector provides the possibility of crossover operation, and the transformation to the local spherical coordinate system makes the mutation operation more convenient.

3.2 Evolutionary Data Augmentation

We use evolutionary data augmentation (EDA) [7] to synthesise a new dataset, as shown in Fig. 2. The original dataset is set as the initial population. A new generation is obtained by go through tectonic evolutionary operators, and then natural selection and the evolution of the generation after generation. Finally, the augmented data can be used for training, in the evolutionary algorithm of 3D human body posture, evolutionary operations are designed as follows.

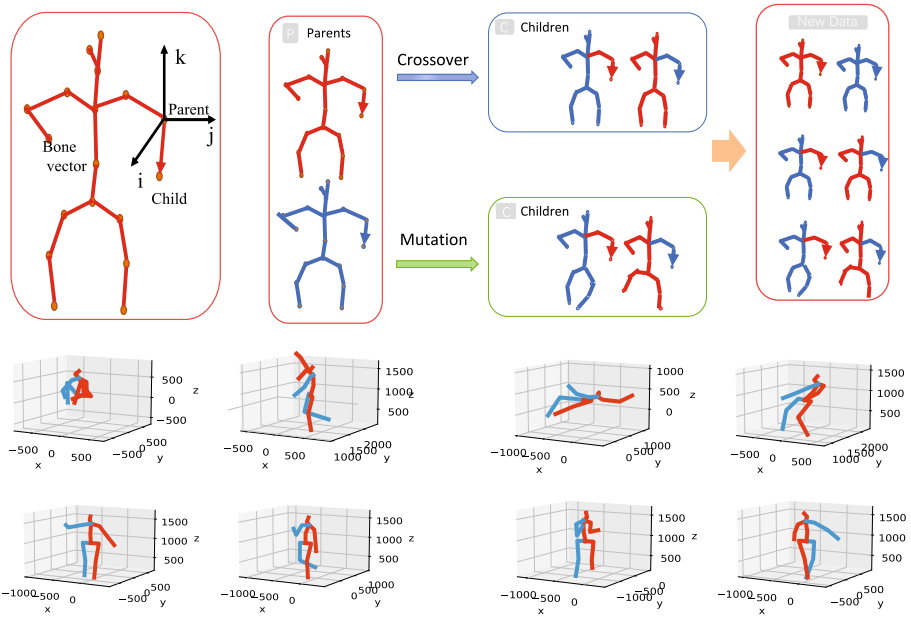

Fig. 2. Top: The human skeleton vector is converted to local spherical coordinates (Left). The process of evolutionary data augmentation. For example, the arms of the two parents are exchanged in crossover; the right and left legs of the two parents are rotated respectively in mutation (Right). **Bottom:** Visualize of initial population (Left) and evolved population (Right) in EDA.

Crossover Operator: Given two parents, we choose a node as a cross node. For example, when the node we choose is the right shoulder, we exchange the bone vector corresponding to the right arm of the two parents. Therefore, the definition the vector selected for crossover operation is

$$\{b^k : parent(k) = q \; and \; F(parent(k), q)\}, \tag{3}$$

where the selection of cross joint k is not random, but based on the result of pre-training, the enhanced distribution is generated. When the performance of a cross node in pre-training is better, the corresponding distribution of this node will be higher.

Mutation Operator: The mutation operation refers to the rotation of bone vectors to increase the diversity of data. In the local spherical coordinate system, $\{(\theta_k, \phi_k)\}$ represents the direction of the bone vector, and the mutation operator is to change the direction of the bone vector, so the definition of the mutation operator is

$$\theta'_k = \theta_k + g, \; \phi'_k = \phi_k + g, \tag{4}$$

where the selection of g is not random, but an enhanced distribution generated according to the information of pre-training model. Similarly, the better the performance of a rotation angle in pre-training model, the higher the distribution of corresponding rotation angle will be.

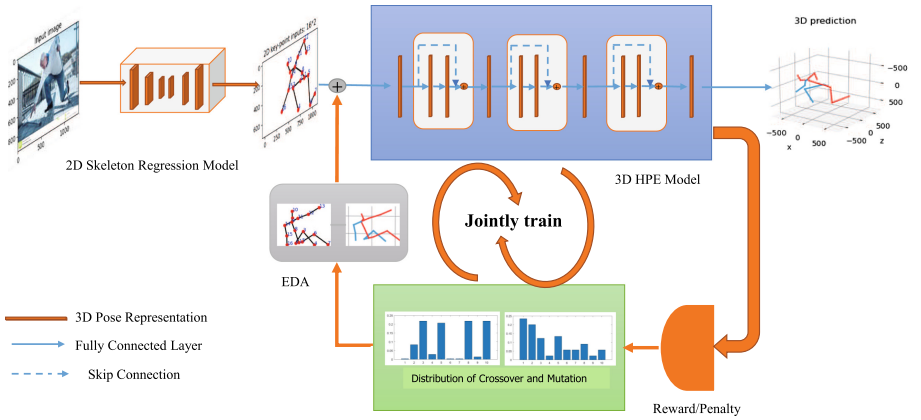

Fig. 3. Overview of our proposed model architecture of jointly optimize improved evolutionary data augmentation and 3D HPE network training. We propose a strategy of reward and penalty functions to jointly train EDA and 3D HPE models.

Natural Selection: We design a simple yet effective way to select the generations. In particular, we use a fitness function to evaluate the rationality of 3D human pose [1]. When 3D human posture is not reasonable, the fitness function $v(p) = -\infty$.

4 Joint Training of EDA and HPE

In this section, we propose a strategy of using reward/penalty function to jointly train EDA and 3D HPE models, as shown in Fig. 3. The proposed traning strategy which jointly optimizing EDA and 3D HPE not only enhances EDA, but also improves the generalization performance of the 3D HPE model.

4.1 Pre-training of HPE Model

In order to enhance the diversity and efficiency of the data synthesised by EDA, a numerical indicator is needed to evaluate these data. Generally, the data synthesised by EDA is a long-tailed distribution, that is, the data that is very effective for the training of the 3D HPE model does not often appear in the synthesized data. If a method can be designed to discover these valid data, then the efficiency of EDA can be improved.

The pre-trained model just meets these requirements. Therefore, using a pre-trained 3D HPE model to back-check the data that is highly efficient for 3D HPE model training. In particular, the proposed method not directly generates the pixels of the 3D human pose picture, but fits the distribution of the nodes of the crossover and mutation operations in the EDA, which greatly reduces the complexity of the algorithm and the consumption of computing resources.

4.2 Strategy of Reward/Penalty Function

When the pre-trained model is obtained, we can optimize the efficiency of the distribution of crossover and mutation operations in EDA. In the training process of the 3D HPE model, after the model is trained with a dataset, if the performance of the 3D HPE model is improved, that is, the loss function of the 3D HPE model on the training dataset is relatively small, then this dataset is highly efficient. On the contrary, if the 3D HPE model is trained with a set of data, its loss function on the training dataset is relatively large, then this dataset is not highly efficient. The strategy we propose is to reward and penalize based on whether the data synthesized by a distributed is efficient. In a certain training process of the 3D HPE model, if the pre-trained 3D HPE model finds that the data synthesized by the distribution P is efficient, then we update P by rewarding

$$P_i' = P_i + \alpha, P_j' = P_j - \frac{\alpha}{n-1}, \forall j \neq i, \tag{5}$$

Similarly, if the pre-trained 3D HPE model finds that the data synthesized by the distribution P is not efficient, then we update P by penalizing

$$P_i' = P_i - \beta, P_j' = P_j + \frac{\beta}{n-1}, \forall j \neq i, \tag{6}$$

where $0 < \alpha, \beta \leq 1$ are hyperparameters that controls the amount of reward and penalty. The greater the value of α and β, the greater the degree of reward and penalty. And n is the number of synthetic data distributions in a set of 3D HPE model training.

Discussion. In reward/penalty strategy, it is important to determine the reference for judging whether synthetic data is efficient. The loss function value of the 3D HPE model on the training dataset is a usable indicator. However, in different training stages, the numerical changes of the loss function are different. For example, the same set of data will cause a rapid decline in the loss function value in early stage of model training, but can only cause a slow decline in later stage of training. Therefore, we use the average of the loss function values of different groups at the same stage as a reference. In addition, when evaluating the efficiency of the distribution, a set of data is used as a unit instead of an individual, which reduces the deviation that may be caused by randomness. Algorithm 1 shows the details of the distribution update process with strategies of reward/penalty function. During the training process, the EDA and 3D HPE model training are alternately carried out, so that the two can improve their own efficiency while also making the other's effect better. Algorithm 2 shows the details of training scheme for joint optimization of EDA and 3D HPE model.

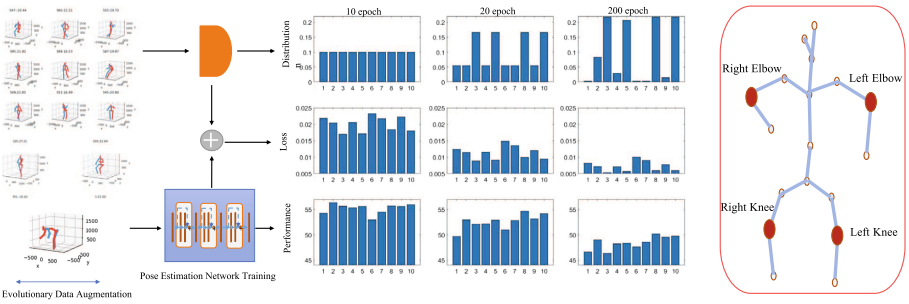

Fig. 4. Network training status visualization: **Left:** distribution of crossover and mutation operation nodes in EDA (**Top**), loss function of 3D HPE model (**Middle**) and performance of 3D HPE model (**Bottom**). **Right:** Location of 4 nodes with high distribution in EDA.

Algorithm 1:
Strategies of reward/penalty
Input: The distribution **P**,
loss function **Loss**.
Output: The updated distribution **P'**.

$Loss_{average} = \frac{1}{n}\sum_{i=1}^{n} Loss_i$
for i = 1 : n
  if $Loss_i < Loss_{average}$
    $P_i' = P_i + \alpha$
    $P_j' = P_j - \frac{\alpha}{n-1}, \forall j \neq i$
  if $Loss_i > Loss_{average}$
    $P_i' = P_i - \beta$
    $P_j' = P_j + \frac{\beta}{n-1}, \forall j \neq i$
Return P'

Algorithm 2:
Scheme for joint optimization
Input: Dataset **X**, Model **HPE**,
Distribution **P**.
Output: New synthesised dataset **X'**,
New model **HPE'**,
New distribution **P'**.

1 Train HPE using X;
2 Calculate the Loss of X in HPE;
3 Update P to P' according to algorithm 1;
4 Perform EDA with P' to get X';
5 Train HPE using X'.

5 Experiments

In this section, we first show the visualization of network training states to verify the motivation of jointly optimizing EDA and 3D HPE network training. Then we quantitatively evaluate the effectiveness of the method in different scenes and further compare with state-of-the-art approaches.

5.1 Datasets, Evaluation Metrics and Implementation Details

H36M is the largest and most accurate body posture dataset with 3D annotation, which is the body posture of 11 people collected by motion sensor [4,5]. The H36M dataset has 7 subject ID (1, 5, 6, 7, 8, 9, 11), we denote a collection of data by appending subject ID to S, e.g., S1 denotes data from subject 1, S15678 denotes data from subject 1, 5, 6, 7 and 8. Mean Per Joint Position Error (MPJPE) was used as the performance index to evaluate the 3D pose estimation model, as follows

$$MPJPE = \frac{1}{N}\sum_{i=1}^{N} \|J_i - J_i^*\|_2, \tag{7}$$

where N is the number of all joints, J_i and J_i^* are respectively the ground truth position and the estimated position of the ith joint. P1 was directly calculated, P2 was calculated with the ground-truth of 3D pose and the predicted value after rigid transformation. P1* is the second-stage model performance without considering the impact of the first-stage model performance and directly using the 2D key points as input. In our experiment, P1 and P2 refer to average MPJPE over all 15 actions for H36M under protocols P1 and P2. P1* is to use ground truth 2D key points for evaluation under Protocol 1.

We jointly optimize EDA and 3D HPE. The train is performed on RTX 2080 Ti GPU and takes about 336 h. To perform EDA, the parameters α and β gradually decrease from 0.1 to 0.01, and it takes about 40 min to get a new synthetic dataset. To train 3D HPE network, we train the cascade model using Adam optimizer with learning rate 0.001 for 200 epochs. Every 20 epochs we perform a data augmentation operation update based on the penalty and reward function strategy. During the test, our model runs at an average of 31.4 fps.

5.2 Comparison with State-of-the-Art Methods

In this set of experiments, we sequentially simulate the situation of data scarcity and data-rich. S1 and S15678 is used as the initial data respectively. The improved EDA method is then used to synthesis new more datasets for jointly optimization in the training model. In 3D HPE network, we directly adopt the network trained on COCO [10] in the first stage, while in the second stage, we use a deep network consisting of three residual modules and a full connection layer. Compared with others, our approach is also applicable to the case with more data, as shown in Table 1 and Table 2.

5.3 Visualization of the Training Status

In this experiment, we are interested in how the strategies of reward/penalty function deal with the influence of loss function. Taking the experiment of the 3D HPE model on S1 of H36M dataset as an example, Fig. 4 visualizes the distribution in the EDA, the loss function value and the performance of the 3D HPE model in different training stages. In EDA, the distribution of 4 nodes (left elbow, right elbow, left knee and right knee) has been significantly increased, as shown in Fig. 4(Right). Our proposed approach believes that crossover and mutation operations on these nodes can synthesize efficient data. Figure 5 shows some examples of our method applied to 3D HPE.

Table 1. The model performance compared with SOTA methods. Best performance of model is marked with bold font.

Supervision	Method Authors	Performance		
		P1	P1*	P2
Weakly	Use Multi-view			
	Rhodin et al. (CVPR18) [18]	–	–	64.6
	Kocabas et al. (CVPR19) [6]	65.3	–	57.2
	Use Temporal information			
	Pavllo et al. (CVPR19) [16]	64.7	–	–
	Single-Image Method			
	Li et al. (ICCV19) [8]	88.8	–	66.5
	Li et al. (CVPR20) [7]	62.9	50.5	47.5
	Ours	**61.4**	**48.5**	**47.3**
Fully	Martinez et al. (ICCV17) [12]	62.9	45.5	47.7
	Yang et al. (CVPR18) [23]]	58.6	–	37.7
	Zhao et al. (CVPR19) [24]	57.6	43.8	–
	Sharma et al. (ICCV19) [20]	58.0	–	40.9
	Moon et al. (ICCV19) [13]	54.4	35.2	–
	Li et al. (CVPR20) [7]	50.9	34.5	38.0
	Ours	**50.4**	**32.1**	**37.7**

Table 2. Performance on S1 of H36M dataset compared with SOTA method over all 15 actions under weakly supervision.

Method Authors	Performance							
	Direct	Discuss	Eat	Greet	Phone	Photo	Pose	Purchase
Rhodin et al. (CVPR18) [19]	78.90	92.80	82.09	86.34	94.10	113.21	83.75	110.55
Li et al. (ICCV19) [11]	70.44	83.61	76.59	77.91	85.43	106.14	72.26	102.93
Ours	**53.86**	**57.78**	**57.17**	**58.21**	**63.55**	**73.51**	**54.73**	**60.26**
	Sit	SitDown	Smoke	Wait	WalkDog	Walk	WalkPair	**Average**
Rhodin et al. (CVPR18) [18]	125.45	185.76	90.57	82.24	99.83	67.04	79.86	97.72
Li et al. (ICCV19) [20]	115.79	164.99	82.43	74.34	94.61	60.15	70.65	88.77
Ours	**65.18**	**81.48**	**59.13**	**57.05**	**68.10**	**52.03**	**59.11**	**61.41**

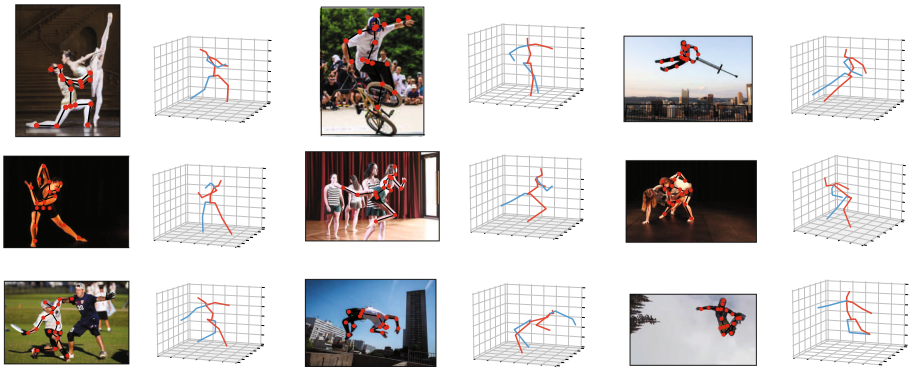

Fig. 5. Some 3D human pose estimation examples of our model. In each example, from left to right are the input image with 2D node coordinates and 3D human pose predicted by the model.

6 Conclusion

In this paper, a joint training strategy with reward and penalty function is proposed to optimize EDA and 3D HPE network during training. Sufficient results on publicly available datasets show that the proposed approach achieves higher performance compared with state-of-the-art methods. In the future research, the proposed method can be extended to other aspects of human pose estimation, such as time-series pictures, multi-person scenes, and multi-view pose estimation. Meanwhile, the data augmentation process and the pose estimation network training process can be further integrated to improve the performance of the pose estimation network.

Acknowledgements. This work was jointly supported by the National Natural Science Foundation of China under grant 62001110, the Natural Science Foundation of Jiangsu Province under grant BK20200353, the Guangdong Basic and Applied Basic Research Foundation under grant 2020A1515110145, the Shenzhen Science and Technology Program under grant RCBS20200714114858072, and the Fundamental Research Funds for the Central Universities under grant 2242021R10115.

References

1. Akhter, I., Black, M.J.: Pose-conditioned joint angle limits for 3D human pose reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1446–1455 (2015)
2. Chen, W., et al.: Synthesizing training images for boosting human 3D pose estimation. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 479–488. IEEE (2016)
3. Cheng, Y., Yang, B., Wang, B., Yan, W., Tan, R.T.: Occlusion-aware networks for 3D human pose estimation in video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 723–732 (2019)
4. Ionescu, C., Li, F., Sminchisescu, C.: Latent structured models for human pose estimation. In: 2011 International Conference on Computer Vision, pp. 2220–2227. IEEE (2011)
5. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2013)
6. Kocabas, M., Karagoz, S., Akbas, E.: Self-supervised learning of 3d human pose using multi-view geometry. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1077–1086 (2019)
7. Li, S., Ke, L., Pratama, K., Tai, Y.W., Tang, C.K., Cheng, K.T.: Cascaded deep monocular 3D human pose estimation with evolutionary training data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6173–6183 (2020)
8. Li, Z., Wang, X., Wang, F., Jiang, P.: On boosting single-frame 3D human pose estimation via monocular videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2192–2201 (2019)
9. Lin, M., Lin, L., Liang, X., Wang, K., Cheng, H.: Recurrent 3d pose sequence machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 810–819 (2017)
10. Lin, T.Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48

11. Luvizon, D.C., Picard, D., Tabia, H.: 2D/3D pose estimation and action recognition using multitask deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5137–5146 (2018)
12. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3D human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2640–2649 (2017)
13. Moon, G., Chang, J.Y., Lee, K.M.: Camera distance-aware top-down approach for 3D multi-person pose estimation from a single RGB image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10133–10142 (2019)
14. Nie, B.X., Wei, P., Zhu, S.C.: Monocular 3D human pose estimation by predicting depth on joints. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 3467–3475. IEEE (2017)
15. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3D human pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7025–7034 (2017)
16. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3D human pose estimation in video with temporal convolutions and semi-supervised training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7753–7762 (2019)
17. Peng, X., Tang, Z., Yang, F., Feris, R.S., Metaxas, D.: Jointly optimize data augmentation and network training: adversarial data augmentation in human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2226–2234 (2018)
18. Rhodin, H., et al.: Learning monocular 3D human pose estimation from multi-view images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8437–8446 (2018)
19. Rogez, G., Schmid, C.: Mocap-guided data augmentation for 3D pose estimation in the wild. In: Advances in Neural Information Processing Systems, pp. 3108–3116 (2016)
20. Sharma, S., Varigonda, P.T., Bindal, P., Sharma, A., Jain, A.: Monocular 3D human pose estimation by generation and ordinal ranking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2325–2334 (2019)
21. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
22. Varol, G., et al.: Learning from synthetic humans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 109–117 (2017)
23. Yang, W., Ouyang, W., Wang, X., Ren, J., Li, H., Wang, X.: 3D human pose estimation in the wild by adversarial learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5255–5264 (2018)
24. Zhao, L., Peng, X., Tian, Y., Kapadia, M., Metaxas, D.N.: Semantic graph convolutional networks for 3D human pose regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3425–3435 (2019)