Self Adversarial Training for Human Pose Estimation

Chia-Jung Chou , Jui-Ting Chien , and Hwann-Tzong Chen
Department of Computer Science
National Tsing Hua University, Taiwan
{jessie33321, ydnaandy123}@gmail.com htchen@cs.nthu.edu.tw

Abstract

This paper presents a deep learning based approach to the problem of human pose estimation. We employ generative adversarial networks as our learning paradigm in which we set up two stacked hourglass networks with the same architecture, one as the generator and the other as the discriminator. The generator is used as a human pose estimator after the training is done. The discriminator distinguishes ground-truth heatmaps from generated ones, and back-propagates the adversarial loss to the generator. This process enables the generator to learn plausible human body configurations and is shown to be useful for improving the prediction accuracy.

1. Introduction

Human pose estimation from a single image is a challenging problem due to the limited information of 2D images and the large variations in configuration and appearance of body parts. Early work often tackles the problem using graphical models [2, 13, 22] and random field inference [23, 35] with handcrafted image features. Despite the improvements made by those intriguing designs of models and algorithms, the bottleneck seems to be the lack of effective feature representations that are capable of characterizing different levels of visual cues and accounting for the varieties in appearance of people.

The situation has been changed along with the popularity of deep learning in computer vision. Deep neural nets have the ability to learn better feature representations. For example, a recent approach, *stacked hourglass network* [28], achieves state-of-the-art performance without the use of hand designed priors or graphical-model-style inference. The well-designed architecture, which supports repeated bottom-up, top-down inference across scales for large receptive field, helps the model to capture some correlations among human body parts. However, the model might predict human pose with implausible configuration due to severe occlusion or overlapping with other people nearby. In these situations, the model is forced to find some similar features which might be in the background or belong to another person. These challenging cases are much easier for human vision to recognize. Humans have the concepts of the structure and constraint of body parts, and are also good at associating these concepts with observed image features. Inspired by the success of *generative adversarial networks* on many topics, we incorporate a discriminator to take charge of checking the structural constraints of human body. We maintain the original pose estimator as the generator to capture important image features. It is worth noting that the architectures of our discriminator and generator are exactly the same. We use the adversarial training strategy to enable the discriminator to distinguish implausible poses and simultaneously to guide the generator. After the training is done, the generator is used as a pose estimator and the discriminator can be removed.

(a) (b)

Figure 1. Motivation. (a) A deep network may produce incorrect estimations due to occlusion. (b) After incorporating adversarial training, the structural constraints of human body parts can be learned.

The main contribution of this work is two folds: First, we design a deep ConvNet model to learn the structure and configuration of human body parts via adversarial training.

978-988-14768-5-2 ©2018 APSIPA

The training techniques of generative adversarial networks are used to train the proposed model for solving the human pose estimation problem. Second, we evaluate our method using LSP, MPII, and LIP datasets, with detailed analysis on the effects of different components in our design, and the experimental results show improved accuracy on all of those datasets.

2. Related Work

2.1. Human Pose Estimation

Many recent methods on human pose estimations use deep neural nets to predict the keypoints of human body in an image. DeepPose [38], one of the earliest deep-learning based approaches to human pose estimation, formulates the pose estimation problem as a regression problem using a standard convolutional architecture, and its performance is higher than classical approaches [2, 12, 13, 21, 31, 41]. Latest methods mostly aim to predict structural outputs, usually called heatmaps or support maps that characterize the probabilities of observing each keypoint at different locations. The exact location of a keypoint is further estimated by finding the maximum in an aggregation of heatmaps. Compared with direct-regression methods, heatmap-based methods better leverage the distributed properties of convolutional networks and are considered more suitable for training.

Some works incorporate graphical models, *e.g*. CRF, MRF, which may be used as a post-processing step [9] or embedded into the network for end-to-end training [11, 40]. Powerful CNN architectures have been developed to capture the important cues and evidences of human parts. In [39] and [28], a multi-stage scheme is employed to make the receptive field large enough for learning the long-range spatial relationships. Also, intermediate supervision is used to produce intermediate confidence maps and let them be refined through different stages. Several recent methods focus on solving the multi-person pose estimation problem. The methods of [19, 32] estimate poses of multiple people in a single image. They use deep networks to generate keypoint candidates and run integer linear programming (ILP) to group joints candidates for each person. The approach of Cao *et al*. [7] predicts the multi-person keypoint heatmaps and the part affinity fields, and then uses a greedy algorithm to group the joints that belong to the same person.

2.2. Generative Adversarial Networks

Generative adversarial networks (GANs) flourish in generating natural images such as human faces and indoor scenes. With the introduction by Goodfellow *et al*. [16], the two-player minimax game allows unsupervised training of generative models and avoids the blur effect of using variational autoencoders. However, people concern about GANs being unstable and hard to train. Radford *et al*. [33] introduce DCGAN, an all convolutional architecture which is easier to train. They propose some elements to increase the model stability such as eliminating the fully connected layer and employing batch normalization to prevent from losing diversity, *i.e*., mode collapsing. DCGAN uses an effective network configuration to make the training of GAN more feasible.

Recently, Arjovsky *et al*. [3] propose Wasserstein GAN (WGAN), which does not require a special network design like DCGAN. WGAN uses the Wasserstein distance to replace the original loss function in GAN and solves the unreliable gradient problem in the original GAN. Using Wasserstein distance also provides an estimate of the quality of the generated samples. However, since WGAN satisfies the K-Lipschitz constraint by weight clipping, it pushes weights towards two values (the extremes of the clipping range) and is hard to tune the clipping parameters. Gulrajani *et al*. [17] replace the weight clipping strategy by gradient penalty. Gradient penalty is an additional term in the loss function that directly enforces the discriminator's gradient norm around K. The result shows that the improved training strategy of [17] is much faster and more stable than WGAN. Another branch of work uses autoencoders as discriminators such as EBGAN [42]. EBGAN aims to match the autoencoder loss distribution while typical GANs try to match the data distribution. EBGAN still suffers from the same problem of classical GANs. Inherited from [42], Berthelot *et al*. [5] present an equilibrium term, which is based on *proportional control theory*, to balance the discriminator and the generator. It also provides a convergence measure that can be used to determine if the model has collapsed or reached its final state.

Due to the success of GAN on generating images, it also draws attention to the field of supervised learning. The concept of *conditional GAN* [27] is introduced for incorporating class information. Several methods combine the conditional GAN loss and the L1 or L2 distance between generated data and ground-truth data. The methods of [20, 24, 30] use this solution to perform tasks of super-resolution, image inpainting, and image translation. They get promising results with respect to human vision. The examples described above are still all about generating natural images. They either generate a whole image based on certain constraints or generate an image patch. Another type of task is about generating heatmaps of labels as in semantic segmentation [26], saliency [29] or human pose estimation [10]. Adding the adversarial training strategy to this type of task seems to bring some benefits to it. In our work, we also try to use adversarial training techniques [5]. to improve the performance of pose estimator.

3. Adversarial Training with the Stacked Hourglass Networks

Our model splits into two networks, the generator and the discriminator. The first network, *generator*, is a fully convolutional network with residual blocks and a conv-deconv architecture. After feeding forward through the generator, we get a set of heatmaps that indicate the confidence score at every location for each keypoint. The second network, *discriminator*, has the same architecture as the generator but it encodes the heatmaps along with the RGB image and decodes them into a new set of heatmaps in order to distinguish real heatmaps from fake ones. The framework of our model is illustrated in Fig. 2.

3.1. Generator

The goal of the generative network is to learn a mapping from a color image to keypoint heatmaps. The deep convolutional architecture allows itself to learn contextual feature representation from the input images. Additionally, the adversarial loss from the discriminative network is introduced and combined with the error between the generated heatmaps and the ground-truth heatmaps. This process enables the generator to learn not only the features and spatial dependencies from images but also the plausible human body configurations.

3.1.1 Network Architecture

We use the state-of-the-art hourglass architecture [28] as our base network. It is a fully convolutional network with residual modules as its building blocks. The network starts with an initial process of a 7×7 convolution with stride 2, followed by several residual modules and max-pooling layers. The initial process reduces the resolution of the feature maps from 256 to 64. Then, a sequence of hourglass modules are stacked to predict the keypoint heatmaps. A single hourglass module is a bottom-up and top-down design to extract the features at every scale. For human pose estimation, it is essential to explore both the local evidence, such as a small region around the wrist, and the long-range relationships between joints. To maintain the information and to integrate global and local context concurrently, skip connections are used, and features at each resolution can be better preserved. A 4-stack hourglass architecture is shown in Fig. 3

3.1.2 Training the Generator

Training the generator is done by back-propagating the loss $\mathcal{L}_{\text{MSE}}$ from generator itself and the adversarial loss $\mathcal{L}_{\text{adv}}$ from the discriminator.

The generator consists of N stacks of hourglass modules. The expected output of each hourglass module contains M heatmaps, each of which is a 64×64 map with a Gaussian peaked at the ground-truth location of the jth joint. The supervision is conducted at the end of each hourglass. The loss from the generator itself can be expressed as

$$\mathcal{L}_{\text{MSE}} = \sum_{i=1}^{N} \sum_{j=1}^{M} (C_{ij} - \hat{C}_{ij})^2 , \tag{1}$$

where C_{ij} is the ground-truth heatmap of jth joints at the ith stack, and $\hat{C}_{ij}$ is the generated heatmap. We calculate the mean square error between them to enforce the generator to learn the image features that are important for localizing the keypoints. In early stacks, local evidence is used since the receptive field is restricted to a small area. In later stacks, long-range spatial relationships will be considered since the receptive field has been enlarged through the numerous sequential convolutional operations. This training scheme is illustrated in Fig. 4.

In addition to the traditional supervised loss described above, we add an adversarial loss, which can urge the generator to produce reasonable poses. The adversarial loss from the discriminator can be expressed as

$$\mathcal{L}_{\text{adv}} = \sum_{j=1}^{M} (\hat{C}_j - D(\hat{C}_j, X))^2 , \tag{2}$$

where $\hat{C}_j$ is the output heatmaps of the generator's last hourglass stack, D is the discriminator, and X is an input image. The loss computes the error between generated heatmaps and reconstructed heatmaps. The detail of this equation will be explained in the next section.

The total loss for generator is defined by

$$\mathcal{L}_G = \mathcal{L}_{\text{MSE}} + \lambda_G \mathcal{L}_{\text{adv}} , \tag{3}$$

where λ_G is a hyperparameter to control the weight of adversarial loss.

3.2. Discriminator

The objective of the discriminator is to distinguish real data from generated data. The input of the discriminator contains either ground-truth heatmaps or generated heatmaps, and both of them are concatenated with the corresponding color image of the person. From the input pair, the discriminator should learn whether the pose described by the heatmaps is correct and corresponds to the person in the input color image. The discriminator attempts to reconstruct a new set of heatmaps. The qualities of the reconstructed heatmaps are determined by how they are similar to the input heatmaps, following the same notion as autoencoder. The loss is computed as the error between the input heatmaps and the reconstructed heatmaps.

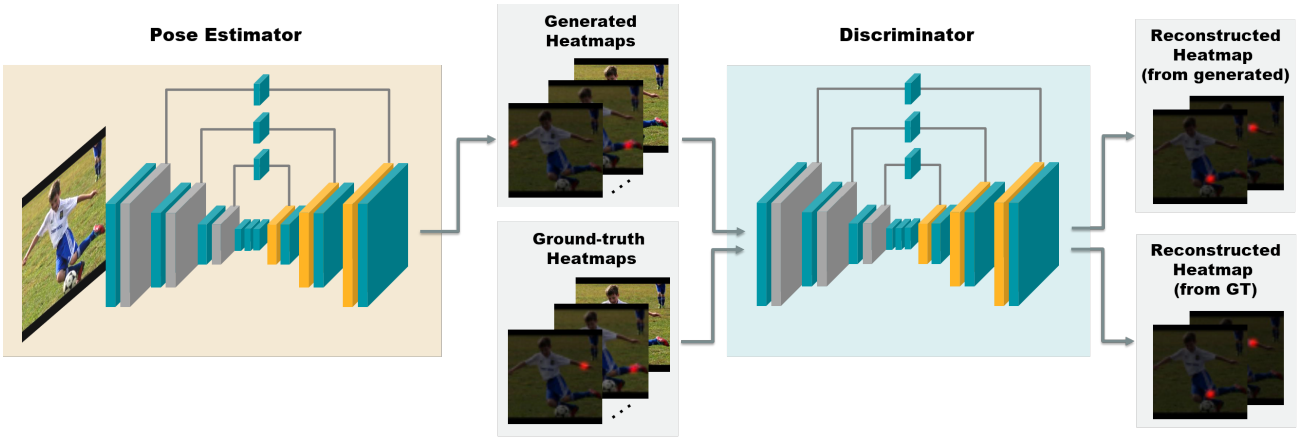


Figure 2. The framework of our adversarial networks. We incorporate a ConvNet-based pose estimator as the generator (on the left) with a discriminator (on the right) that aims to distinguish the generated heatmaps from the ground-truth heatmaps by reconstructing the input heatmaps. The generator and the discriminator have the same architecture.

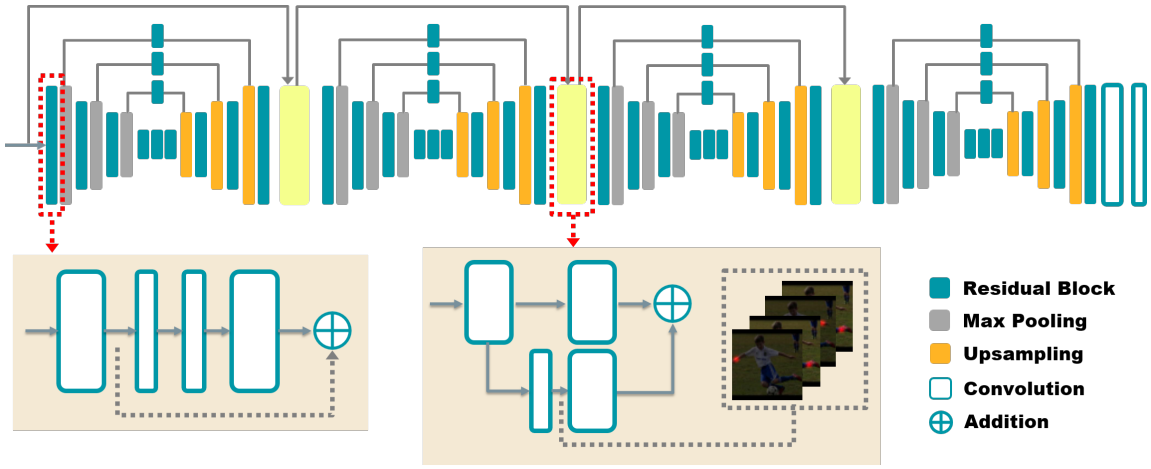


Figure 3. The architecture of 4-stack hourglass. The hourglass module consists of residual blocks (zoomed-in at bottom-left), pooling layers, upsampling layers, and skip connections. Between each pair of consecutive hourglass stacks, there is a transition block (yellow box) which produces intermediate heatmaps and adds them to the main trunk of the network.

3.2.1 Training the Discriminator

For each training image, the generated and ground-truth heatmaps will be fed to the discriminator separately. Two sets of heatmaps will be reconstructed for computing $\mathcal{L}_{\text{real}}$ and $\mathcal{L}_{\text{fake}}$. In other words, at each iteration, the discriminator is updated using the accumulated gradient, which is computed with respect to $\mathcal{L}_{\text{real}}$ and $\mathcal{L}_{\text{fake}}$.

When the input comprises ground-truth heatmaps, the discriminator is trained to recognize it and reconstruct a similar one, *i.e.*, to minimize the error between the ground-truth heatmaps and the reconstructed ones. On the other hand, if the input comprises generated heatmaps, the discriminator is trained to reconstruct totally different ones, *i.e.*, to drive the error between the generated heatmaps and the reconstructed ones as large as possible. The loss is expressed as

$$\begin{aligned} \mathcal{L}_{\text{real}} &= \sum_{j=1}^{M} (C_j - D(C_j, X))^2, \\ \mathcal{L}_{\text{fake}} &= \sum_{j=1}^{M} (\hat{C}_j - D(\hat{C}_j, X))^2, \\ \mathcal{L}_D &= \mathcal{L}_{\text{real}} - k_t \, \mathcal{L}_{\text{fake}} \, . \end{aligned} \tag{4}$$

The discriminator is optimized by the per-pixel loss $\mathcal{L}_D$. Given a set of heatmaps, which can refer to a particular pose, the discriminator will give a value to each pixel. The value is the error between the input and output heatmaps of the discriminator. The value means how good the confidence of this pixel is, in the discriminator's opinion. For example, if the confidence of the right knee is high nearby

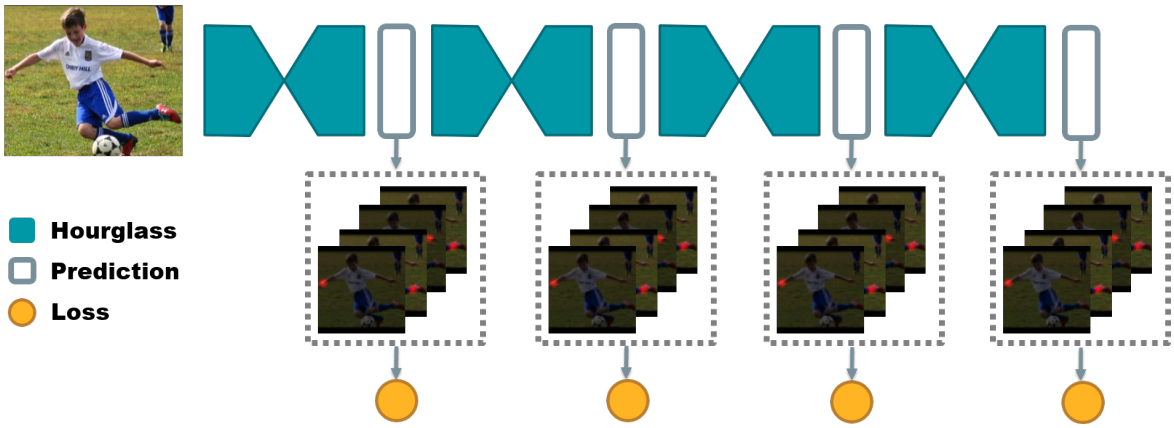


Figure 4. An illustration of intermediate supervision. The mean squared error (MSE) loss is applied at the end of each hourglass module.

the left knee, a well-trained discriminator will produce a heatmap of the right knee that has a larger error at the location of left knee.

Since the discriminator is like a critic, it offers detailed 'comments' on the input heatmaps and suggests which parts in the heatmaps do not yield a real pose. This is different from the conventional GAN, which only judges the whole input being good or bad.

As mentioned in many papers, GAN is unstable and hard to train since the network easily collapses when the discriminator gets too good too quickly. Inspired by [5], we use a variable k_t to control the balance between generator and discriminator. The variable is updated at every iteration t. The adaptive term k_t is defined by

$$k_{t+1} \;=\; k_t + \lambda_k \left(\gamma\, \mathcal{L}_{\text{real}} - \mathcal{L}_{\text{fake}}\right), \qquad (5)$$

where k_t is bounded between 0 and 1, and λ_k is a hyperparameter. As in Eq. (4), k_t means how much emphasis is put on $\mathcal{L}_{\text{fake}}$. When the generator gets better than the discriminator, *i.e.*, $\mathcal{L}_{\text{fake}}$ is smaller than $\gamma\, \mathcal{L}_{\text{real}}$, the generated heatmaps are real enough to fool the discriminator. Hence, k_t will increase, to make the term $\mathcal{L}_{\text{fake}}$ more dominant, and thus the discriminator will be trained more on recognizing the generated heatmaps. The proportion it accelerates to train on $\mathcal{L}_{\text{fake}}$ is according to how far the discriminator falls behind the generator, *i.e.*, $\gamma\, \mathcal{L}_{\text{real}} - \mathcal{L}_{\text{fake}}$. Similarly, when the discriminator gets better than the generator, k_t decreases, to slow down the training on $\mathcal{L}_{\text{fake}}$ so that the generator can keep up with it.

3.3. Adversarial Training

Based on *generative adversarial networks* (GANs), our training scheme is supervised learning plus a two-player game. The terms $\mathcal{L}_{\text{fake}}$ in Eq. (4) and Eq. (2) have the same value except for the sign. The generator aims to minimize the distance between $\hat{C}$ and $D(\hat{C}, X)$ while the discriminator tries to maximize it. This is the adversarial part of this learning procedure. To distinguish poses, the discriminator seeks to capture the essential factor of real pose distribution during the process of reconstruction. At the same time, the generator seeks to produce higher-quality heatmaps of human pose so that it can deceive the discrminator and pass the inspection to let discriminator reconstruct similar heatmaps. In addition to the unsupervised training game, we preserve the traditional supervised learning to make the generator learn quicker and prevent the network from collapsing. Algorithm 1 summarizes the adversarial training process.

Algorithm 1: The adversarial training process.

Input : An image X of a person and the corresponding ground-truth heatmaps C

do
- Forward discriminator by $D(C, X)$
- Compute gradient ∇f_D w.r.t. Eq. (4)
- Forward generator by $\hat{C} = G(X)$
- Compute gradient ∇f_G w.r.t. Eq. (1)
- Forward discriminator by $D(\hat{C}, X)$
- Accumulate gradient ∇f_D w.r.t. Eq. (4)
- Update discriminator with ∇f_D
- Accumulate gradient ∇f_G w.r.t. Eq. (2)
- Update generator with ∇f_G

while $\hat{C}$ *still improves*;

3.3.1 Inference

After the training is done, the discriminator can be removed. We use the generated heatmaps $\hat{C} = G(X)$ to infer the final result. To stabilize the predictions, we evaluate both the original image and its flipped version, and average their output heatmaps. As in the training phase, the output heatmap size of a joint is 64×64. We first extract the location with the largest confidence score in each joint's heatmap. Then,

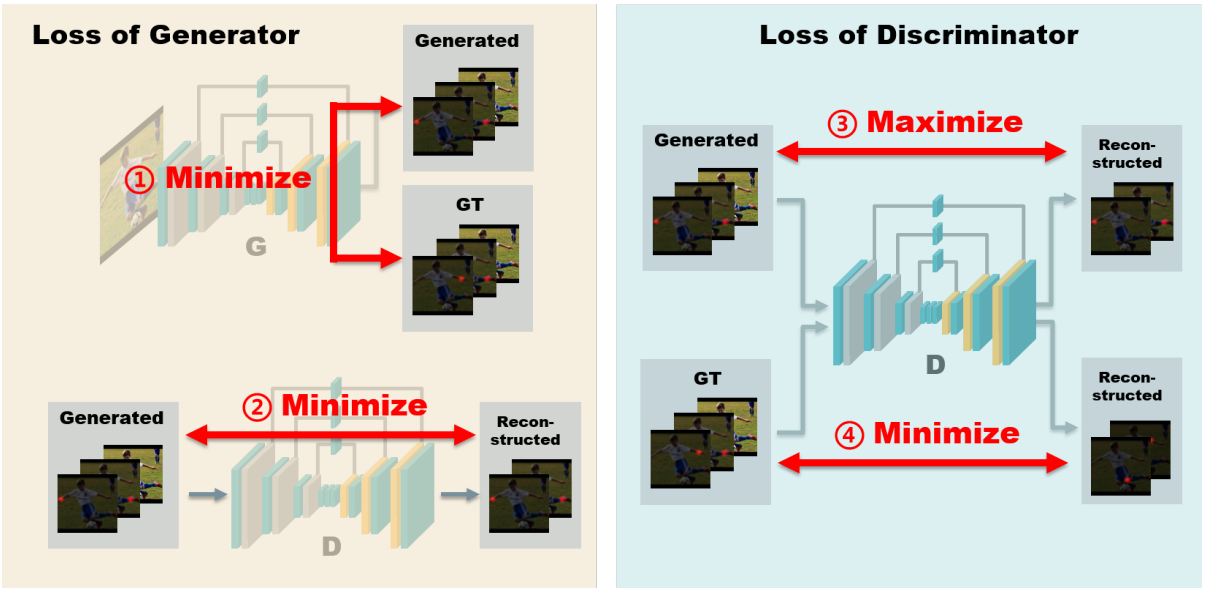


Figure 5. Summary of $\mathcal{L}_G$ and $\mathcal{L}_D$. Losses in the orange box are used to update the generator. Losses in the blue box are used to update the discriminator.

we transform the location back to the original coordinate space with respect to the input image size.

4. Experiments

4.1. Datasets

We evaluate our method on three benchmark datasets, Leeds Sports Pose Dataset (LSP) [21], MPII Human Pose Dataset [1], and Look Into Person Dataset (LIP) [15]. In the following experiments, we use the same preprocessing and data augmentation settings. We randomly flip an input image horizontally, rotate it by an angle in $[-30, 30]$ degrees, and scale it with factors in $[0.75, 1.25]$. During testing, we scale the LSP and LIP images uniformly across the whole datasets to make the person a suitable size in the image. For MPII images, we use the scale and center annotations provided with the images. We implement our methods using Torch7 libraries for deep learning. We set a batch size of 6 and train the network from scratch using the RMSprop optimization algorithm. The experiments are performed on a Titan X GPU.

- **Leeds Sports Pose Dataset** (LSP):

  Our results of LSP are trained on the LSP plus LSP-extended dataset. LSP consists of 11,000 poses for training and 1,000 for testing. The images are gathered from Flickr and contain people who are doing sports such as baseball, parkour, tennis, and so on. Each image is annotated with 14 keypoint locations. To make it easier to integrate with other datasets, we calculate the center and scale of annotated person and use it at the training phase. The label of this dataset is a little noisy since some occluded joints may not have location information or the location might be wrong. The noisy labels and the variations in poses of humans doing sports make this dataset quite challenging.

- **MPII Human Pose Dataset** (MPII) :

  MPII dataset contains about 25,000 images and over 40,000 annotated people. These data are divided into 30,000 images for training and 10,000 images for testing. Each person is annotated with 16 joints. The images are extracted from YouTube videos where the contents are everyday human activities. In comparison with other pose datasets, MPII has richer information such as activity label and fully unannotated video frames, and has higher image resolution. We only use keypoint locations during training.

- **Look Into Person Human Pose Dataset** (LIP) :

  LIP is the newest and largest dataset for human pose estimation. It contains 50,000 images with 19 semantic human part labels and 16 human keypoints. In the following experiments, we only use keypoints information. The dataset divides into 30,462 images for training set, 10,000 images for validation set, and 10,000 for test set. The images may contain full body, half body, or part of body, with heavy occlusions and of low resolution. The dataset is also used in CVPR 2017 workshop 'Visual Understanding of Humans in Crowd Scene' and the first 'Look Into Person (LIP) Challenge'.

4.2. Evaluation Metrics

The evaluations are based on two metrics. We use PCK to measure performance on LSP and LIP. For MPII, we use PCKh.

- **Percentage of Correct Keypoints** (PCK) [41]:

  PCK reports the percentage of correct detection that falls within a tolerance range. The tolerance range is a fraction of torso size. The equation can be expressed as

  $$\frac{\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2}{\|\mathbf{y}_{\text{lhip}} - \mathbf{y}_{\text{rsho}}\|_2} \leq r\,, \tag{6}$$

  where $\mathbf{y}_i$ is the ground-truth location of the ith keypoint and $\hat{\mathbf{y}}_i$ is the predicted location of the ith keypoint. The fraction r is bounded between 0 and 1.

- **Percentage of Correct Keypoints with respect to head** (PCKh) [1]:

  PCKh is almost the same as PCK except for the tolerance range is a fraction of head size.

4.3. Results

We show in Fig. 6 some qualitative results obtained using our method. Fig. 7 shows a visualization of heatmaps. It can be seen in Fig. 7(a) that the predictions produced by the stacked hourglass network [28] are mostly accurate, but the model is not very sure about its answers according to the heatmaps. Our method is able to refine the heatmaps, as shown in Fig. 7(b).

- LSP: The comparisons between our results and others are reported in Table 1. Our model shown in this table is trained with external data from the MPII training set. The score is computed at $r = 0.2$. As shown in Fig. 4.3, our approach gets the highest detection rate across all tolerance range. Furthermore, the improvement is even more obvious at tighter distance (between 0.05 and 0.1).

- MPII: Table 2 shows the PCKh performance of our method and previous methods at $r = 0.5$. Our model shown in this table is trained with external data from the LSP training set.

- LIP: Table 3 shows the final list of the CVPR 2017 LIP Human Pose Estimation Challenge. The challenge is finished and our method achieves the best result. For reference, both BUPTMM-POSE and Hybrid Pose Machine use methods that merge the predictions of Newell *et al.* [28] and others.

4.4. Analysis

In this section, we present the effects of several components in our model. We conduct the experiments on the test set of the LSP dataset. We observe the accuracy through training iterations.

4.4.1 GAN and Conditional GAN

We experiment on several network configurations. The settings differ in the number of stacks of the generator. The size of the discriminator is fixed (1-stack). As shown in Fig. 10, we find that GAN and conditional GAN perform almost equally in both 1-stack (Fig. 10(a)) and 2-stack (Fig. 10(b)). The discriminator seems to perform well even when the image of the person is not provided. A possible reason is that the implausible pose could be recognized by merely the pose information. The image of the person is an extra information, but the discriminator does not always need it.

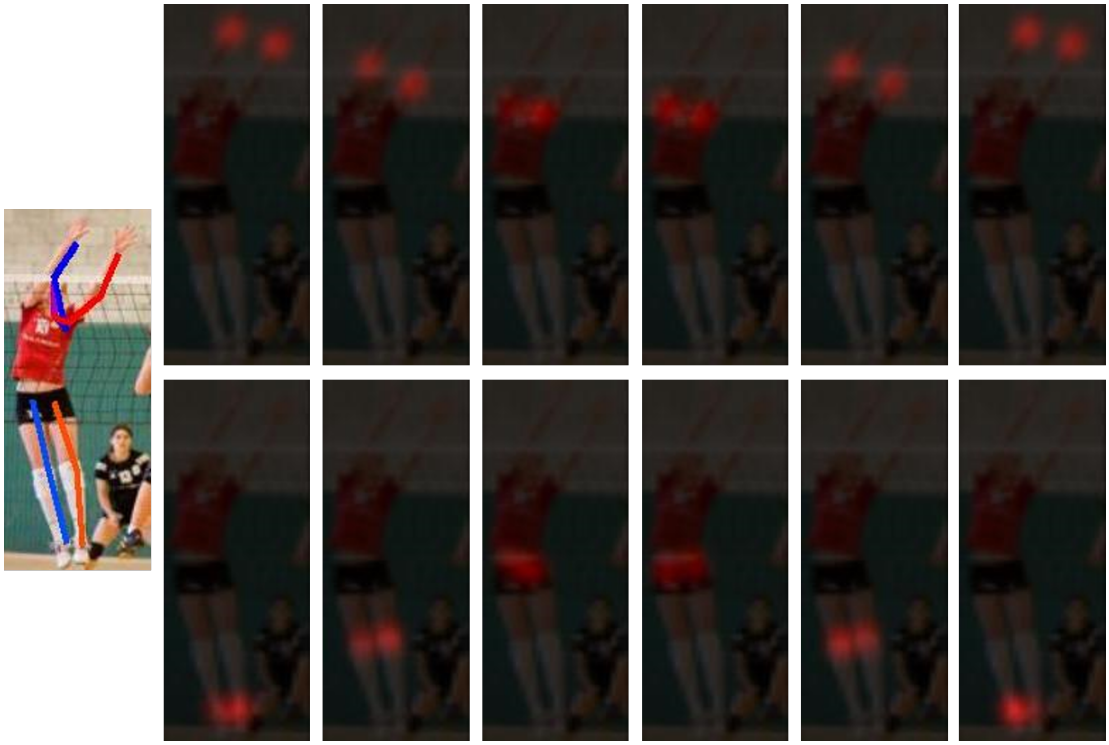
4.4.2 With or without Adversarial Training

To investigate the benefit of adversarial training, we compare our method with the original stacked hourglass network. In Fig. 11(a), the improvement of adding adversarial training is significant. Our method converges faster and ends at a higher accuracy. But when it comes to 2-stack hourglass, in Fig. 11(b), the gain of adversarial training does not seem so obvious like 1-stack hourglass. The lines are staggered across training iterations, although at the end our method is a little higher than the original hourglass. The 8-stack hourglass is the best setting released by the authors of [28], but in our experiment, in Fig. 11(c), 4-stack hourglass plus a discriminator is a better choice. In this setting we decrease the learning rate by 10^{-1} at epoch 60. In Fig. 11(d), we zoom in the part of curve after epoch 60. We find that the strategy of learning rate decay is helpful for both methods, but ours is a bit more stable and achieves better performance in the end.

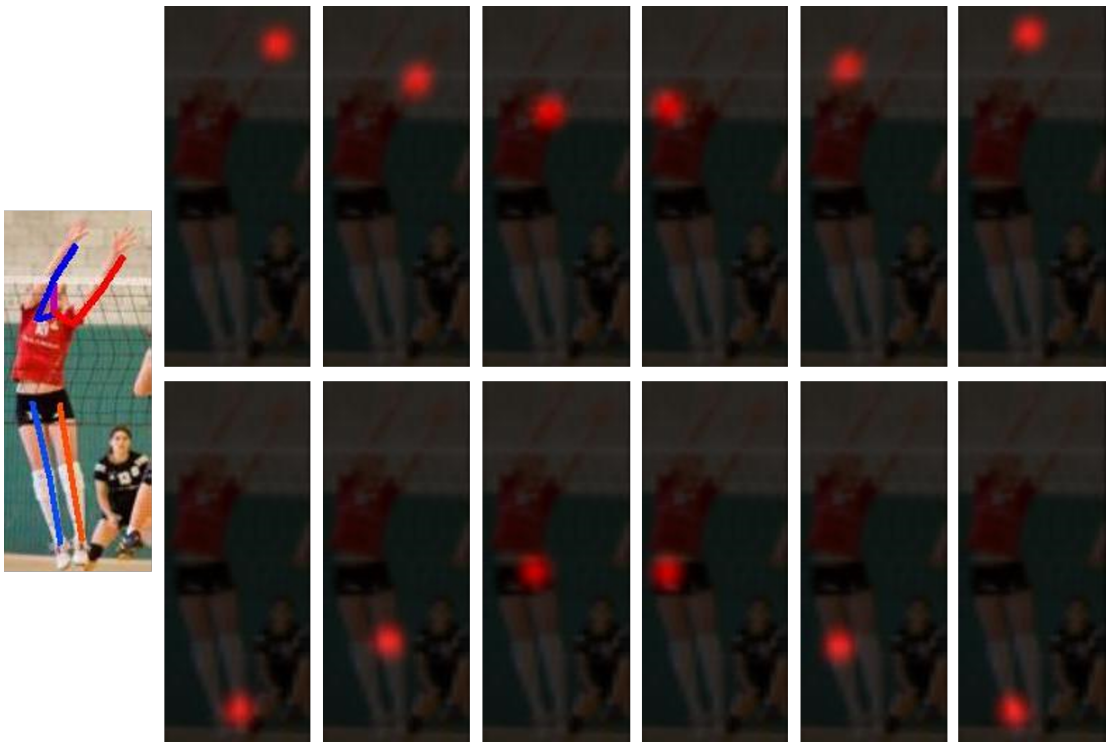
5. Conclusion

We present an adversarial network to solve the human pose estimation problem. The network is composed of a generator and a discriminator with the same architecture. The generator is responsible for predicting the heatamps of human body keypoints based on the image features, and the discriminator plays the role of critic that can distinguish implausible poses and give the generator useful hints to improve the heatmaps. The additional discriminator can be removed after the training is done, and therefore it does not affect the inference time. We evaluate our approach on three standard benchmark datasets and the results show that our approach is useful for improving the prediction accuracy.

Figure 6. Qualitative results. The red and orange lines indicate the left side, and the blue line indicates the right side. Our method can generate more plausible and structural poses than [28].



(a)



(b)

Figure 7. Heatmaps visualization on the LSP dataset. (a) The predictions produced by the stacked hourglass network [28] are mostly accurate, but the heatmaps show that the model is not very sure about its answers. (b) Our method further refines the heatmaps and corrects the position of right shoulder. The heatmaps from left to right, top to bottom are left wrist, left elbow, left shoulder, right shoulder, right elbow, right wrist, left ankle, left knee, left hip, right hip, right knee, and right ankle.

Table 1. Human pose estimation on the LSP dataset. (PCK)

Methods	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Total
Lifshitz *et al.* [25], ECCV'16	97.8	93.3	85.7	80.4	85.3	76.6	70.2	85.0
Pishchulin *et al.* [32] , CVPR'16	97.0	91.0	83.8	78.1	91.0	86.7	82.0	87.1
Insafutdinov *et al.* [19], ECCV'16	96.8	**95.2**	89.3	84.4	88.4	83.4	78.0	88.5
Wei *et al.* [39], CVPR'16	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Bulat *et al.* [6], ECCV'16	97.2	92.1	88.1	85.2	92.2	91.4	88.7	90.7
Chu *et al.* [11], CVPR'17	98.1	93.7	89.3	86.9	93.4	94.0	92.5	92.6
Ours	**98.2**	94.9	**92.2**	**89.5**	**94.2**	**95.0**	**94.1**	**94.0**

Table 2. Human pose estimation on the MPII dataset. (PCKh)

Methods	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Total
Pishchulin *et al.* [31], ICCV'13	74.3	49.0	40.8	34.1	36.5	34.4	35.2	44.1
Tompson *et al.* [37], NIPS'14	95.8	90.3	80.5	74.3	77.6	69.7	62.8	79.6
Carreira *et al.* [8], CVPR'16	95.7	91.7	81.7	72.4	82.8	73.2	66.4	81.3
Tompson *et al.* [36], CVPR'15	96.1	91.9	83.9	77.8	80.9	72.3	64.8	82.0
Hu *et al.* [18], CVPR'16	95.0	91.6	83.0	76.6	81.9	74.5	69.5	82.4
Pishchulin *et al.* [32], CVPR'16	94.1	90.2	83.4	77.3	82.6	75.7	68.6	82.4
Lifshitz *et al.* [25], ECCV'16	97.8	93.3	85.7	80.4	85.3	76.6	70.2	85.0
Gkioxary *et al.* [14], ECCV'16	96.2	93.1	86.7	82.1	85.2	81.4	74.1	86.1
Rafi *et al.* [34], BMVC'16	97.2	93.9	86.4	81.3	86.8	80.6	73.4	86.3
Belagiannis *et al.* [4], FG'17	97.7	95.0	88.2	83.0	87.9	82.6	78.4	88.1
Insafutdinov *et al.* [19], ECCV'16	96.8	95.2	89.3	84.4	88.4	83.4	78.0	88.5
Wei *et al.* [39], CVPR'16	97.8	95.0	88.7	84.0	88.4	82.8	79.4	88.5
Bulat *et al.* [6], ECCV'16	97.9	95.1	89.9	85.3	89.4	85.7	81.7	89.7
Newell *et al.* [28], ECCV'16	98.2	96.3	91.2	87.1	90.1	87.4	83.6	90.9
Chu *et al.* [11], CVPR'17	**98.5**	96.3	91.9	**88.1**	90.6	88.0	**85.0**	91.5
Ours	98.2	**96.8**	**92.2**	88.0	**91.3**	**89.1**	84.9	**91.8**

References

[1] M. Andriluka, L. Pishchulin, P. V. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 6, 7

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009. 1, 2

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *CoRR*, abs/1701.07875, 2017. 2

[4] V. Belagiannis and A. Zisserman. Recurrent human pose estimation. *CoRR*, abs/1605.02914, 2016. 10

[5] D. Berthelot, T. Schumm, and L. Metz. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017. 2, 5

[6] A. Bulat and G. Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *ECCV*, 2016. 10

[7] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. *CoRR*, abs/1611.08050, 2016. 2

[8] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. *CoRR*, abs/1507.06550, 2015. 10

[9] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014. 2

[10] Y. Chen, C. Shen, X. Wei, L. Liu, and J. Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. *CoRR*, abs/1705.00389, 2017. 2

[11] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. *CoRR*, abs/1702.07432, 2017. 2, 10

[12] M. Dantone, J. Gall, C. Leistner, and L. J. V. Gool. Human pose estimation using body parts dependent joint regressors. In *CVPR*, 2013. 2

[13] P. F. Felzenszwalb, D. A. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 1, 2

[14] G. Gkioxari, A. Toshev, and N. Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, 2016. 10

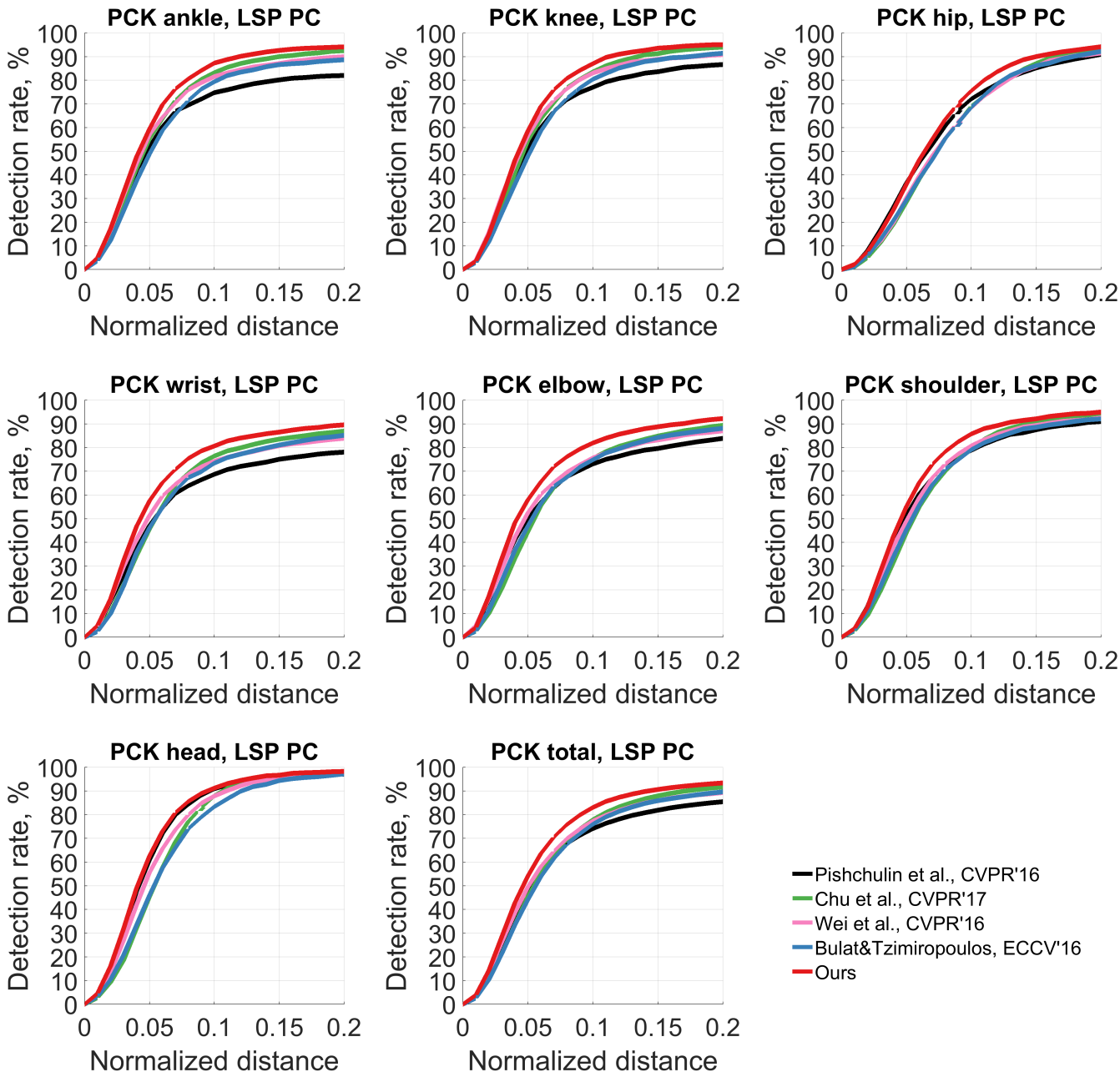


Figure 8. Percentage of Correct Keypoints (PCK) on the LSP dataset. All methods are trained with external data from the MPII training set, in addition to the LSP training set. PC refers to the person-centric annotation.

Table 3. Human pose estimation on the LIP dataset. (PCK)

Methods	Head	Sho.	Elb.	Wri.	Hip	Knee	Ank.	Total
Hybrid Pose Machine	71.7	87.1	82.3	78.2	69.2	77.0	73.5	77.2
BUPTMM-POSE	90.4	87.3	81.9	78.8	68.5	75.3	75.8	80.2
Pyramid Stream Network	91.1	88.4	82.2	79.4	70.1	80.8	81.2	82.1
Ours	**94.9**	**93.1**	**89.1**	**86.5**	**75.7**	**85.5**	**85.7**	**87.4**

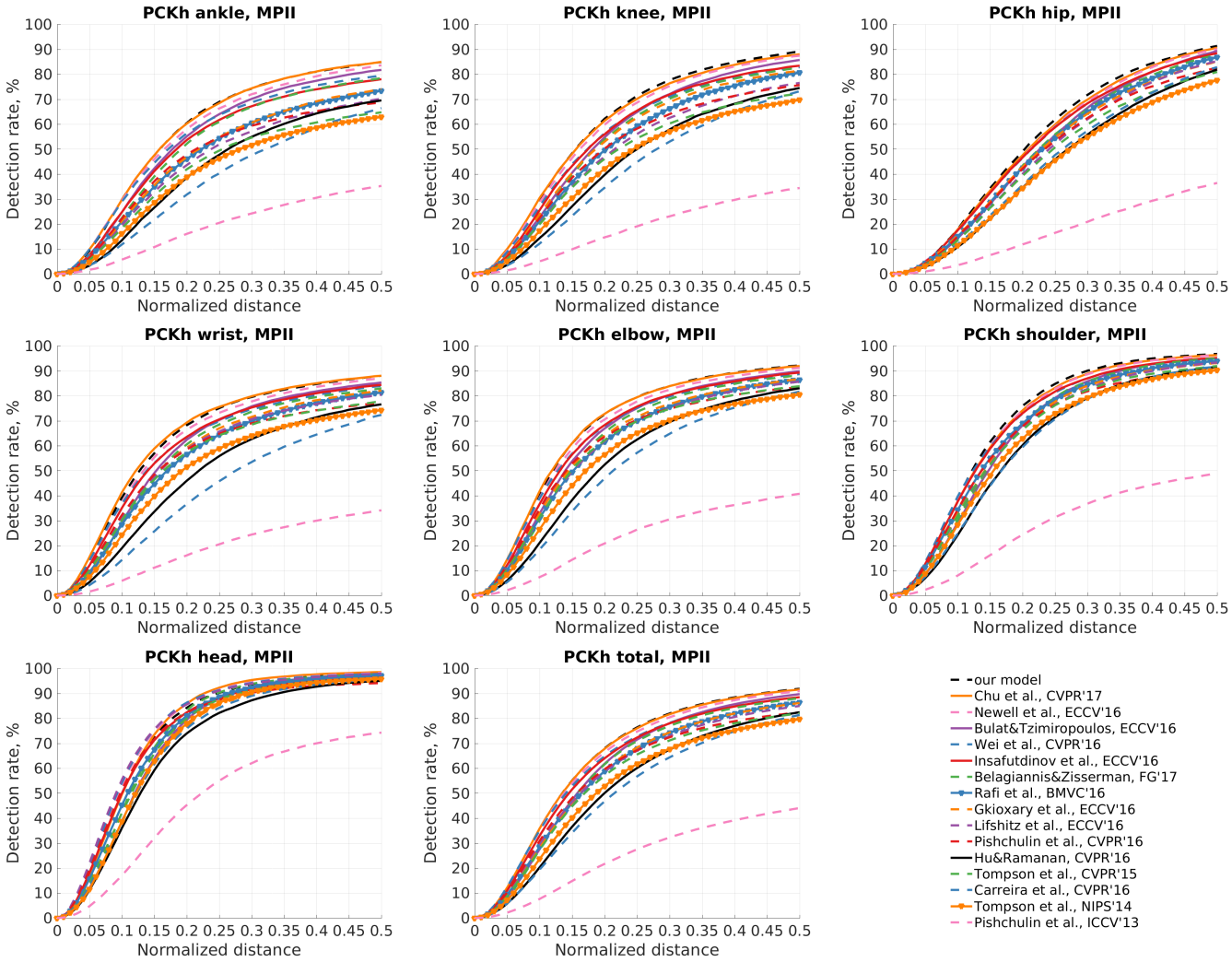


Figure 9. PCKh on the MPII dataset.

[15] K. Gong, X. Liang, X. Shen, and L. Lin. Look into person: Self-supervised structure-sensitive learning and A new benchmark for human parsing. *CoRR*, abs/1703.05446, 2017. 6

[16] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial networks. *CoRR*, abs/1406.2661, 2014. 2

[17] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017. 2

[18] P. Hu and D. Ramanan. Bottom-up and top-down reasoning with hierarchical rectified gaussians. In *CVPR*, 2016. 10

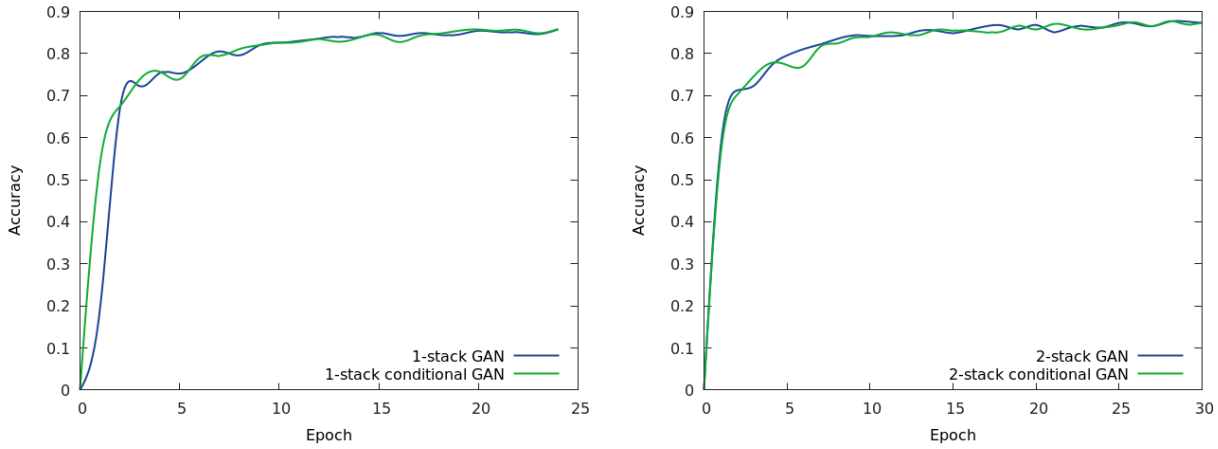
[19] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. *CoRR*, abs/1605.03170, 2016. 2, 10

[20] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 2

[21] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. 2, 6

[22] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, 2011. 1

[23] L. Ladicky, P. H. S. Torr, and A. Zisserman. Human pose estimation using a joint pixel-wise and part-wise formulation. In *CVPR*, 2013. 1



(a) (b)

Figure 10. PCK on the LSP dataset. The blue line is the accuracy of GAN while the green line is of conditional GAN. (a) 1-stack hourglass. (b) 2-stack hourglass.

[24] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *CoRR*, abs/1609.04802, 2016. 2

[25] I. Lifshitz, E. Fetaya, and S. Ullman. Human pose estimation using deep consensus voting. *CoRR*, abs/1603.08212, 2016. 10

[26] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic segmentation using adversarial networks. *CoRR*, abs/1611.08408, 2016. 2

[27] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 2

[28] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *CoRR*, abs/1603.06937, 2016. 1, 2, 3, 7, 8, 9, 10, 14

[29] J. Pan, C. Canton-Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giró i Nieto. Salgan: Visual saliency prediction with generative adversarial networks. *CoRR*, abs/1701.01081, 2017. 2

[30] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[31] L. Pishchulin, M. Andriluka, P. V. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *ICCV*, 2013. 2, 10

[32] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016. 2, 10

[33] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 2

[34] U. Rafi, B. Leibe, J. Gall, and I. Kostrikov. An efficient convolutional network for human pose estimation. In *BMVC*, 2016. 10

[35] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In *ECCV*, 2014. 1

[36] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *CVPR*, 2015. 10

[37] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014. 10

[38] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014. 2

[39] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016. 2, 10

[40] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, 2016. 2

[41] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 2, 7

[42] J. J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *CoRR*, abs/1609.03126, 2016. 2

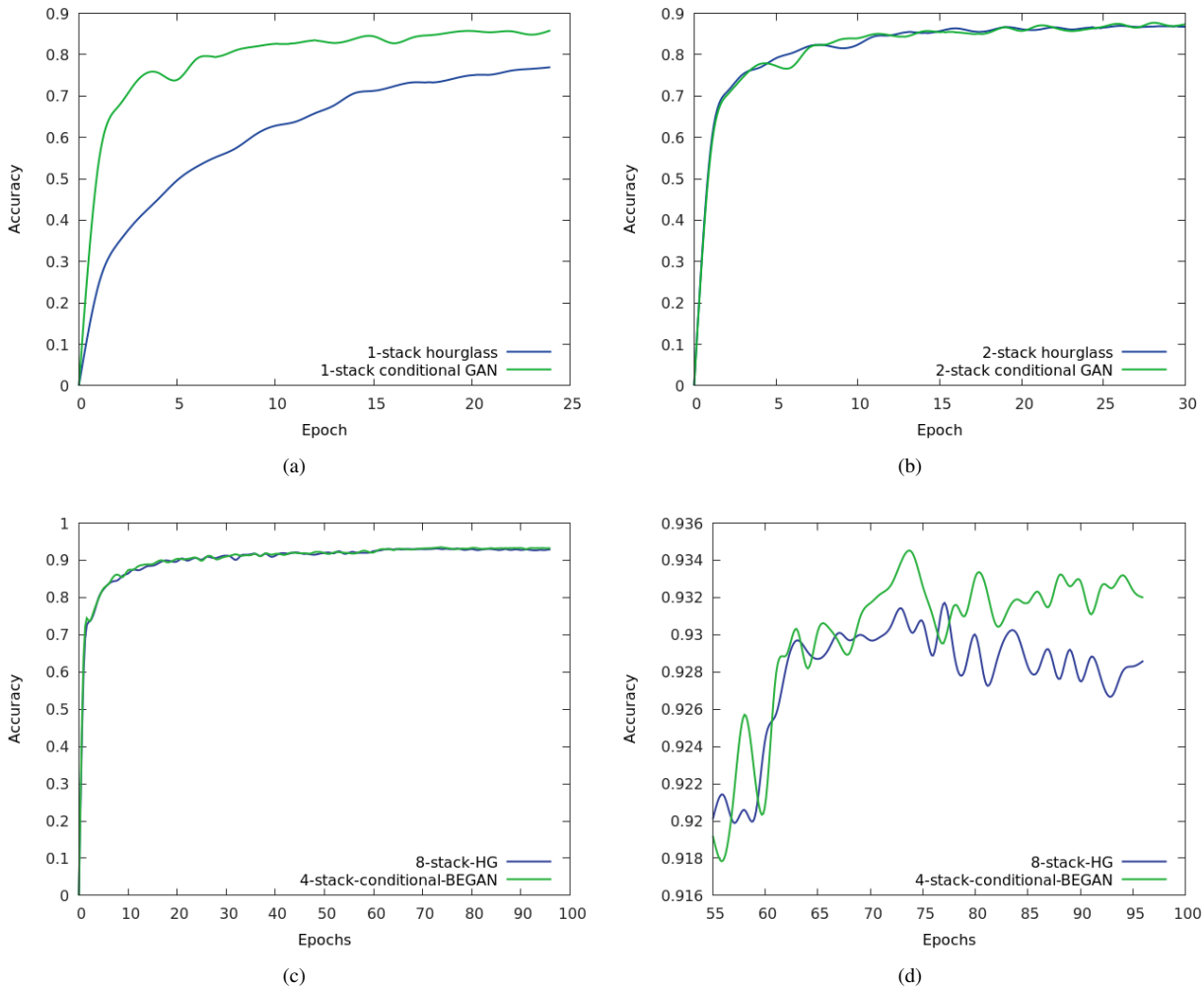


Figure 11. PCK on the LSP dataset. The blue line is the approach of [28] while the green line is ours. (a) 1-stack hourglass. (b) 2-stack hourglass. (c) 8-stack standard hourglass versus 4-stack hourglass plus a discriminator. In this setting we decrease the learning rate by 10^{-1} at epoch 60. (d) We zoom in the part of curve after epoch 60. We find that the strategy of learning rate decay is helpful for both methods, but ours is a bit more stable and achieves better performance in the end.