Regressive Domain Adaptation for Unsupervised Keypoint Detection

Junguang Jiang[1], Yifei Ji[1], Ximei Wang[1], Yufeng Liu[2], Jianmin Wang[1], Mingsheng Long[1] (✉)
[1]School of Software, BNRist, Tsinghua University, China
[2]Y-tech, Kuaishou Technology
{jjg20,jiyf17,wxm17}@mails.tsinghua.edu.cn, {jimwang,mingsheng}@tsinghua.edu.cn

Abstract

Domain adaptation (DA) aims at transferring knowledge from a labeled source domain to an unlabeled target domain. Though many DA theories and algorithms have been proposed, most of them are tailored into classification settings and may fail in regression tasks, especially in the practical keypoint detection task. To tackle this difficult but significant task, we present a method of regressive domain adaptation (RegDA) for unsupervised keypoint detection. Inspired by the latest theoretical work, we first utilize an adversarial regressor to maximize the disparity on the target domain and train a feature generator to minimize this disparity. However, due to the high dimension of the output space, this regressor fails to detect samples that deviate from the support of the source. To overcome this problem, we propose two important ideas. First, based on our observation that the probability density of the output space is sparse, we introduce a spatial probability distribution to describe this sparsity and then use it to guide the learning of the adversarial regressor. Second, to alleviate the optimization difficulty in the high-dimensional space, we innovatively convert the minimax game in the adversarial training to the minimization of two opposite goals. Extensive experiments show that our method brings large improvement by 8% *to* 11% *in terms of PCK on different datasets.*

1. Introduction

Many computer vision tasks have achieved great success with the advent of deep neural networks in recent years. However, the success of deep networks relies on a large amount of labeled data [14], which is often expensive and time-consuming to collect. Domain adaptation (DA) [21], which aims at transferring knowledge from a labeled source domain to an unlabeled target domain, is a more economical and practical option than annotating sufficient target samples, especially in the keypoint detection tasks. The fast development of computer vision applications leads to huge increases in demand for keypoint detection but the annotations of this task are more complex than classification tasks, requiring much more labor work especially when the objects are partially occluded. On the contrary, accurately labeled synthetic images can be obtained in abundance by computer graphics processing at a low cost [27, 23]. Hence, enabling domain adaptation in regression settings for unsupervised keypoint detection has a promising future.

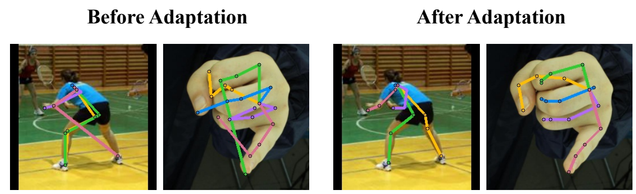


Figure 1. **Visualization** before and after adaptation on the unlabeled target domain. (Left) The wrong predictions before adaptation are usually located at other keypoints. (Right) The predictions of the adapted model look more like hands or bodies.

There are many effective DA methods for classification [17, 6, 22, 30], but we empirically found that few methods work on regression. One possible reason is that there exist explicit task-specific boundaries between classes in classification. By applying domain alignment, the margins of boundaries between different classes on the target domain are enlarged, thereby helping the model generalize to the unlabeled target domain. However, the regression space is usually continuous on the contrary, *i.e.*, there is no clear decision boundary. Meanwhile, although images have limited pixels, the keypoint is still in a *large* discrete space due to a combination of different axes, posing another huge challenge for most DA methods.

To solve the issues caused by the large output space, we delved into the predictions of a source-only keypoint detection model. We observed that when the predictions on the unlabeled domain are wrong, they are *not equally distributed* on the image. For example, if the position of a *right* ankle is mistaken (Figure 1), the wrong prediction is most likely at the position of the left ankle or other keypoints, instead of somewhere in the background as we expected. This unexpected observation reveals that the output space is *sparse* in the sense of probability. Consider an extremely sparse case where the predicted position is always located at a keypoint, then a specific ankle detection problem becomes a K-way classification problem, and we can reduce the domain gap by enlarging the decision boundary between keypoints. This extreme case gives us a strong hint that if we can constrain the output space from a whole image space into a smaller one with only K keypoints, it may be possible to bridge the gap between regression and classification.

This paper aims to enable regressive domain adaptation (RegDA). Inspired by the latest domain adaptation theory—disparity discrepancy (DD) [30], we first use an adversarial regressor to maximize the disparity on the target domain and train a feature generator to minimize this disparity. Based on the aforementioned observations and analyses, we introduce a spatial probability distribution to describe the sparsity and use it to guide the optimization of the adversarial regressor. It can somewhat avoid the problems caused by the large output space and reduce the gap between keypoint detection and classification in domain adaptation. Besides, we also found that maximizing the disparity of two regressors is unbelievably difficult (see Section 5.2.4). To this end, we convert the minimax game in DD [30] into *minimization of two opposite goals*. This conversion has effectively overcame the optimization difficulty of adversarial training in RegDA. Our contributions are summarized as follows:

- We discovered the sparsity of regression output space in the sense of probability, which provides a hint to bridge the gap between regression and classification.
- We proposed RegDA, an effective regression method, which converts the minimax game between two regressors into the minimization of two opposite goals.
- We conducted rich experiments on various keypoint detection tasks and validate that our method can bring performance gains by 8% to 11% in terms of PCK.

2. Related Work

Domain Adaptation. Most deep neural networks suffer from performance degradation due to the domain shift [20]. Domain Adaptation (DA) is proposed to transfer knowledge from the source domain to the target domain. DAN [17] adopts adaptation layers to minimize an optimal MK-MMD [7] between domains. DANN [6] first introduces adversarial training into domain adaptation. MCD [22] uses two task-specific classifiers to approximate the $\mathcal{H}\Delta\mathcal{H}$-distance [2] between source and target and minimizes it by feature adaptation. MDD [30] extends the theories of domain adaptation to multiclass classification and proposes a novel measurement of domain discrepancy. These methods mentioned above are insightful and effective in classification problems. *But few of them work on regression problems.* In our work, we propose a novel training method for domain adaptation in keypoint detection, which is a typical regression problem.

Keypoint Detection. 2D keypoint detection has become a popular research topic these years for its wide use in computer vision applications. Tompson et al. [25] propose a multi-resolution framework that generates heatmaps representing per-pixel likelihood for keypoints. Hourglass [18] develops a repeated bottom-up, top-down architecture, and enforces intermediate supervision by applying loss on intermediate heatmaps. Xiao et al. [29] propose a simple and effective model that adds a few deconvolutional layers on ResNet [9]. HRNet [24] maintains high resolution through the whole network and achieves notable improvement. *Note that our method is not intended to further refine the network architecture*, but to solve the problem of domain adaptation in 2D keypoint detection. Thus our method is compatible with any of these heatmap-based networks.

Some previous works have explored DA in keypoint detection, but most in 3D keypoints detection. Cai et al. [3] propose a weakly-supervised method with the aid of depth images and Zhou et al. [32] conduct weakly-supervised domain adaptation with a 3D geometric constraint-induced loss. These methods assume 2D ground truth available on target domain and use a *fully-supervised method* to get 2D heatmap. Zhou et al. [33] utilize view-consistency to regularize predictions from unlabeled target domain in 3D keypoints detection, but depth scans and images from different views are required on the target domain. *Our problem setup is completely different from the above works since we only have unlabeled 2D data on the target domain.*

Loss Functions for Heatmap Regression. Heatmap regression is widely adopted in keypoint detection problems, which computes the the mean squared error between the predicted heatmap and the ground truth one [25, 28, 4, 5, 16, 24]. Besides, Mask R-CNN [8] adopts the cross-entropy loss, where the ground truth is a one-hot heatmap. Some other works [10, 19] take the problem as a binary classification for each pixel. Differently, *we present a new loss function based on KL-divergence*, which is suitable for RegDA.

3. Preliminaries

3.1. Learning Setup

In supervised 2D keypoint detection, we have n labeled samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ from $\mathcal{X} \times \mathcal{Y}^K$, where $\mathcal{X} \in \mathcal{R}^{H\times W\times 3}$ is the input space, $\mathcal{Y} \in \mathcal{R}^2$ is the output space and K is the number of keypoints for each input. The samples independently drawn from the distribution D are denoted as $\widehat{D}$. The goal is to find a regressor $f \in \mathcal{F}$ that has the lowest error rate $\text{err}_D = \mathbb{E}_{(x,y)\sim D} L(f(\boldsymbol{x}), y)$ on D, where L is a loss function we will discuss in Section 4.1.

In unsupervised domain adaptation, there exists a labeled source domain $\widehat{P} = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^n$ and an unlabeled target domain $\widehat{Q} = \{\boldsymbol{x}_i^t\}_{i=1}^m$. The objective is to minimize err_Q.

3.2. Disparity Discrepancy

Definition 1 (Disparity [30]). *Given two hypothesis $f, f' \in \mathcal{F}$, we define the disparity between them as*

$$\mathrm{disp}_D(f', f) \triangleq \mathbb{E}_D L(f', f). \tag{1}$$

Definition 2 (Disparity Discrepancy, DD [30]). *Given a hypothesis space $\mathcal{F}$ and a specific regressor $f \in \mathcal{F}$, the Disparity Discrepancy (DD) is defined by*

$$d_{f,\mathcal{F}}(P, Q) \triangleq \sup_{f' \in \mathcal{F}} (\mathrm{disp}_Q(f', f) - \mathrm{disp}_P(f', f)). \tag{2}$$

It has been proved that when L satisfies the triangle inequality, the expected error $\mathrm{err}_Q(f)$ on the target domain is **strictly** bounded by the sum of four terms: empirical error on the source domain $\mathrm{err}_{\widehat{P}}(f)$, empirical disparity discrepancy $d_{f,\mathcal{F}}(\widehat{P}, \widehat{Q})$ between source and target, the ideal error λ and complexity terms [30]. Thus our task becomes

$$\min_{f \in \mathcal{F}} \mathrm{err}_{\widehat{P}}(f) + d_{f,\mathcal{F}}(\widehat{P}, \widehat{Q}). \tag{3}$$

We train a feature generator network ψ (see Figure 2) which takes inputs $\boldsymbol{x}$, and regressor networks f and f' which take features from ψ. We approximate the supremum in Equation (2) by maximizing disparity discrepancy (DD):

$$\begin{aligned}\max_{f'} \mathcal{D}(\widehat{P}, \widehat{Q}) = \mathbb{E}_{\boldsymbol{x}^t \sim \widehat{Q}} L((f' \circ \psi)(\boldsymbol{x}^t), (f \circ \psi)(\boldsymbol{x}^t)) \\ - \mathbb{E}_{\boldsymbol{x}^s \sim \widehat{P}} L((f' \circ \psi)(\boldsymbol{x}^s), (f \circ \psi)(\boldsymbol{x}^s)).\end{aligned} \tag{4}$$

When the regressor f' is close to the supremum, minimizing the following terms will decrease err_Q effectively,

$$\min_{\psi, f} \mathbb{E}_{(\boldsymbol{x}^s, \boldsymbol{y}^s) \sim \widehat{P}} L((f \circ \psi)(\boldsymbol{x}^s), \boldsymbol{y}^s) + \eta \mathcal{D}(\widehat{P}, \widehat{Q}), \tag{5}$$

where $\eta > 0$ is the trade-off coefficient.

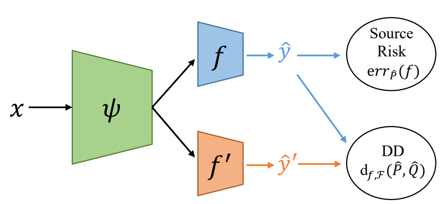


Figure 2. DD architecture under the keypoint detection setting.

4. Method

4.1. Supervised Keypoint Detection

Most top-performing methods on keypoint detection [29, 24, 18] generate a likelihood heatmap $\mathcal{H}(\boldsymbol{y}_k) \in R^{H' \times W'}$ for each keypoint $\boldsymbol{y}_k$. The heatmap usually has a 2D Gaussian blob centered on the ground truth location $\boldsymbol{y}_k$. Then we can use L_2 distance to measure the difference between the predicted heatmap $f(\boldsymbol{x}^s)$ and the ground truth $\mathcal{H}(\boldsymbol{y}^s)$. The final prediction is the point with the maximum probability in the predicted map $\boldsymbol{h}_k$, i.e. $\mathcal{J}(\boldsymbol{h}_k) = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} \boldsymbol{h}_k(\boldsymbol{y})$. Heatmap learning shows good performance in the supervised setting. However, when we apply it to the minimax game for domain adaptation, we empirically find that it will lead to a numerical explosion. The reason is that $f(\boldsymbol{x}^t)$ is not bounded, and the maximization will increase the value at all positions on the predicted heatmap.

To overcome this issue, we first define the *spatial probability distribution* $\mathcal{P}_{\mathrm{T}}(\boldsymbol{y}_k)$, which normalizes the heatmap $\mathcal{H}(\boldsymbol{y}_k)$ over the spatial dimension,

$$\mathcal{P}_{\mathrm{T}}(\boldsymbol{y}_k)_{h,w} = \frac{\mathcal{H}(\boldsymbol{y}_k)_{h,w}}{\sum_{h'=1}^{H'} \sum_{w'=1}^{W'} \mathcal{H}(\boldsymbol{y}_k)_{h',w'}}. \tag{6}$$

Denote by σ the spatial softmax function,

$$\sigma(\mathbf{z})_{h,w} = \frac{\exp(\mathbf{z}_{h,w})}{\sum_{h'=1}^{H'} \sum_{w'=1}^{W'} \exp(\mathbf{z}_{h',w'})}. \tag{7}$$

Then we can use the KL-divergence to measure the difference between the predicted spatial probability $\widehat{\boldsymbol{p}}^s = (\sigma \circ f)(\boldsymbol{x}^s) \in R^{K \times H \times W}$ and the ground truth label $\boldsymbol{y}^s$,

$$L_{\mathrm{T}}(\boldsymbol{p}^s, \boldsymbol{y}^s) \triangleq \frac{1}{K} \sum_{k}^{K} \mathrm{KL}(\mathcal{P}_{\mathrm{T}}(\boldsymbol{y}_k^s) || \boldsymbol{p}_k^s). \tag{8}$$

In supervised setting, models trained with KL-divergence achieve comparable performance with models trained with L_2 loss since both models are provided with pixel-level supervision. Since $\sigma(\mathbf{z})$ sums to 1 in the spatial dimension, the maximization of $L_{\mathrm{T}}(\boldsymbol{p}^s, \boldsymbol{y}^s)$ will not cause the numerical explosion. In our next discussion, KL is used by default.

4.2. Sparsity of the Spatial Density

Compared with classification models, the output space of the keypoint detection models is much larger, usually of size 64×64. Note that the optimization objective of the adversarial regressor f' is to maximize the disparity between the predictions of f' and f on the target domain, and minimize the disparity on the source domain. In other words, we are looking for an adversarial regressor f' which predicts correctly on the source domain, *while making as many mistakes as possible on the target domain*. However, in the experiment on *dSprites* (detailed in Section 5.1), we find that increasing the output space of the adversarial regressor f' will worsen the final performance on the target domain. Therefore, the dimension of the output space has a huge impact on the adversarial regressor. *It would be hard to find the adversarial regressor f' that does poorly only on the target domain when the output space is too large.*

Thus, how to reduce the size of the output space for the adversarial regressor has become an urgent problem. As we mentioned previously (see Figure 1), when the model makes a mistake on the unlabeled target domain, the probability of different positions is not the same. For example, when the model incorrectly predicts the position of the right ankle (see Figure 3), most likely the position of the left ankle is predicted, occasionally other keypoints predicted, and rarely positions on the background are predicted. Hence, *when the input is given, the output space, in the sense of probability, is not uniform.* This spatial density is sparse, i.e., some positions have a larger probability while most positions have a probability close to zero. To explore this space more efficiently, f' should pay more attention to positions with high probability. Since wrong predictions are often located at **other** keypoints, we sum up their heatmaps,

$$\mathcal{H}_{\mathrm{F}}(\widehat{\boldsymbol{y}}_k)_{h,w} = \sum_{k' \neq k} \mathcal{H}(\widehat{\boldsymbol{y}}_{k'})_{h,w}, \tag{9}$$

where $\widehat{\boldsymbol{y}}_k$ is the prediction by the main regressor f. Then we normalize the map $\mathcal{H}_{\mathrm{F}}(\boldsymbol{y}_k)$ into *ground false* distribution,

$$\mathcal{P}_{\mathrm{F}}(\widehat{\boldsymbol{y}}_k)_{h,w} = \frac{\mathcal{H}_{\mathrm{F}}(\widehat{\boldsymbol{y}}_k)_{h,w}}{\sum_{h'=1}^{H'} \sum_{w'=1}^{W'} \mathcal{H}_{\mathrm{F}}(\widehat{\boldsymbol{y}}_k)_{h',w'}}. \tag{10}$$

We use $\mathcal{P}_{\mathrm{F}}(\widehat{\boldsymbol{y}}_k)$ to approximate the *spatial probability distribution* that the model makes mistakes at different locations and we will use it to guide the exploration of f' in Section 4.3. The size of the output space of the adversarial regressor is reduced *in the sense of expectation*. Essentially, we are making use of the sparsity of the spatial density to ease the minimax game in a high-dimensional output space.

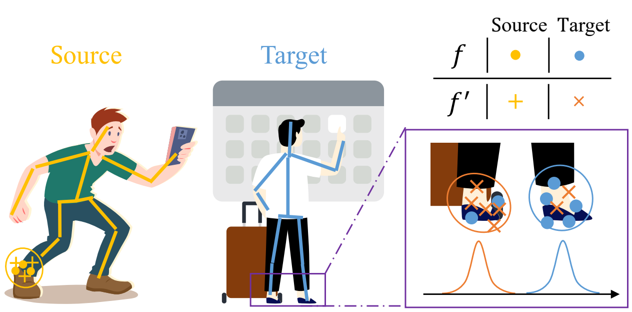


Figure 3. The task is to predict the position of the right ankle. Predictions of f and f' on the source domain (in yellow) are near the right ankle. Predictions of f on the target domain (in blue) are sometimes wrong and located at the left ankle or other keypoints. The predictions of f' on the target domain (in orange) are encouraged to locate at other keypoints in order to detect samples far from the support of the right ankle.

4.3. Minimax of Target Disparity

Besides the problem discussed above, there is still one problem in the minimax game of the target disparity. Theoretically, the minimization of KL-divergence between two distributions is unambiguous. As the probability of each location in the space gets closer, two probability distributions will also get closer. *But the maximization of KL-divergence will lead to uncertain results.* Because there are many situations where the two distributions are different, for instance, the variance is different or the mean is different.

In the keypoint detection, we usually use PCK (detailed in Section 5.2.3) to measure the quality of the model. As long as the output of the model is near the ground truth, it is regarded as a correct prediction. Therefore, we are more concerned about the target samples whose prediction is far from the true value. In other words, we hope that after maximizing the target disparity, there is a big difference between the mean of the predicted distribution ($\widehat{\boldsymbol{y}}'$ should be different from $\widehat{\boldsymbol{y}}$ in Figure 4). However, experiments show that $\widehat{\boldsymbol{y}}'$ and $\widehat{\boldsymbol{y}}$ are almost the same during the adversarial training (see Section 5.2.4). In other words, *maximizing KL mainly changes the variance of the output distribution.* The reason is that KL is calculated point by point in the space. When we maximize KL, the probability value of the peak point ($\widehat{\boldsymbol{y}}'$ in Figure 4) is reduced, and the probability of other positions will increase uniformly. Ultimately the variance of the output distribution increases, but the mean of the distribution does not change significantly, which is completely inconsistent with our expected behavior. Since the final predictions of f' and f are almost the same, it's hard for f' to detect target samples that deviate from the support of the source domain. Thus, the minimax game takes little effect.

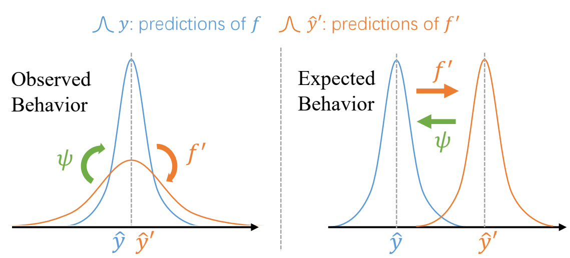


Figure 4. When we maximize the KL-divergence between the predictions by f' and f (fixed), we expect to maximize the mean difference, but what actually changes is often *only* the variance.

Since minimization cannot get our expected behavior, can we avoid using it and only use minimization in the adversarial training? The answer is yes. The reason that we had to maximize before was that we only had one optimization goal. *If we have two goals with opposite physical meanings, then the minimization of these two goals can play the role of minimax game.* Our task now is to design two opposite goals for the adversarial regressor and the feature generator. The goal of the feature generator is to minimize the target disparity or minimize the KL-divergence between the predictions of f' and f. The objective of the adversarial regressor is to maximize the target disparity, and we achieve this by minimizing the KL-divergence between the predictions of f' and the *ground false* predictions of f,

$$L_{\mathrm{F}}(\boldsymbol{p}', \boldsymbol{p}) \triangleq \frac{1}{K}\sum_{k}^{K} \mathrm{KL}(\mathcal{P}_{\mathrm{F}}(\mathcal{J}(\boldsymbol{p}))_k || \boldsymbol{p}'_k), \tag{11}$$

where $\boldsymbol{p}' = (\sigma \circ f' \circ \psi)(\boldsymbol{x}^t)$ is the prediction of f' and $\boldsymbol{p}$ is the prediction of f. Compared to directly maximizing the distance from the *ground truth* predictions of f, minimizing L_{F} can take advantage of the spatial sparsity and effectively change the mean of the output distribution.

Now we use Figure 3 to explain Equation (11). Assume we have K supports for each keypoint in the output space. The outputs on the labeled source domain (in yellow) will fall into the correct support. But for outputs on the target domain, the position of the left ankle might be confused as the right ankle. And these are the samples far from the supports. Through minimizing L_{F}, we mislead f' to predict other keypoints as right ankle, which encourages the adversarial regressor f' to detect target samples far from the support of the right ankle. Then we train the feature generator network ψ to fool the adversarial regressor f' by minimizing L_{T} on the target domain. This encourages the target features to be generated near the support of the right ankle. This adversarial learning steps are repeated and the target features will be aligned to the supports of the source finally.

4.4. Overall Objectives

The final training objectives are summarized as follows. Though described in different steps, these loss functions are optimized simultaneously in a unified framework.

Objective 1. First, we train the generator ψ and regressor f to detect the source samples correctly. Also, we train the adversarial regressor f' to minimize its disparity with f on the source domain. The objective is as follows:

$$\begin{aligned}\min_{\psi, f, f'} \mathbb{E}_{(\boldsymbol{x}_s, \boldsymbol{y}^s)\sim \widehat{P}} (L_{\mathrm{T}}((\sigma \circ f \circ \psi)(\boldsymbol{x}_s), \boldsymbol{y}^s) \\ + \eta L_{\mathrm{T}}((\sigma \circ f' \circ \psi)(\boldsymbol{x}_s), (\mathcal{J} \circ f \circ \psi)(\boldsymbol{x}_s))).\end{aligned} \tag{12}$$

Objective 2. Besides, we need the adversarial regressor f' to increase its disparity with f on the target domain by minimizing L_{F}. By maximizing the disparity on the target domain, f' can detect the target samples that deviate far from the support of the source. This corresponds to Objective 2 in Figure 5, which can be formalized as follows,

$$\min_{f'} \eta \mathbb{E}_{\boldsymbol{x}_t \sim \widehat{Q}} L_{\mathrm{F}}((\sigma \circ f' \circ \psi)(\boldsymbol{x}_t), (f \circ \psi)(\boldsymbol{x}_t)). \tag{13}$$

Objective 3. Finally, the generator ψ needs to minimize the disparity between the current regressors f and f' on the target domain. This corresponds to Objective 3 in Figure 5,

$$\min_{\psi} \eta \mathbb{E}_{\boldsymbol{x}_t \sim \widehat{Q}} \eta L_{\mathrm{T}}((\sigma \circ f' \circ \psi)(\boldsymbol{x}_t), (\mathcal{J} \circ f \circ \psi)(\boldsymbol{x}_t)). \tag{14}$$

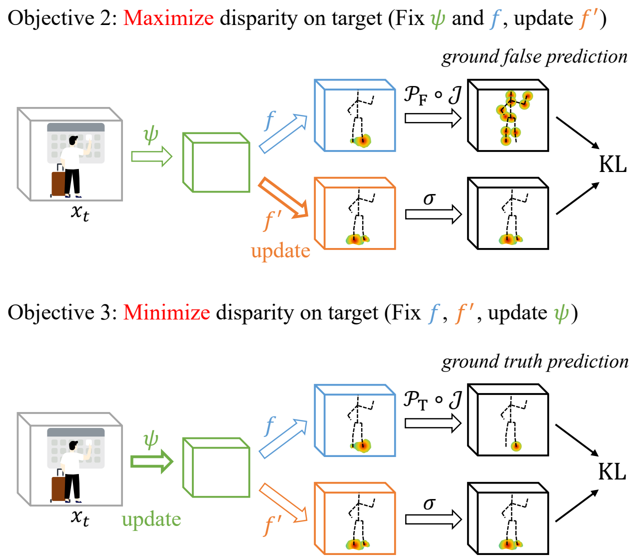


Figure 5. Adversarial training objectives. Our network has three parts: feature generator ψ, regressor f and adversarial regressor f'. **Objective 2**: f' learns to maximize the target disparity by minimizing its KL with *ground false* predictions of f. **Objective 3**: ψ learns to minimize the target disparity by minimizing the KL between the predictions of f' with *ground truth* predictions of f.

5. Experiments

First, we experiment on a toy dataset called *dSprites* to illustrate the effects of high dimension on the minimax game. Then we perform extensive experiments on real-world datasets, including hand datasets (*RHD→H3D*) and human datasets (*SURREAL→Human3.6M*, *SURREAL→LSP*), to verify the effectiveness of our RegDA method. We set $\eta = 1$ on all datasets. Code is available at https://github.com/thuml/Transfer-Learning-Library.

5.1. Experiment on Toy Datasets

Dataset *DSprites* is a 2D synthetic dataset (see Figure 6). It consists of three domains: *Color* (C), *Noisy* (N) and *Scream* (S), with $737,280$ images in each. There are four regression factors and we will focus on two of them: position X and Y. We generate a 64×64 heatmap for the keypoint. Experiments are performed on six transfer tasks: C→N, C→S, N→C, N→S, S→C, and S→N.

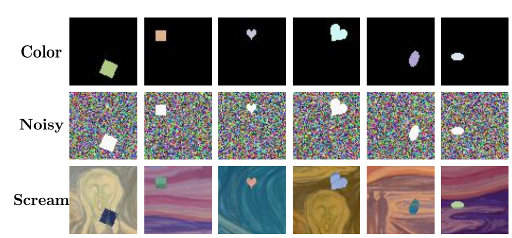


Figure 6. Some example images in the dSpirtes dataset.

Implementation Details We finetune ResNet18 [9] pre-trained on ImageNet. Simple Baseline [29] is used as our detector head and is trained from scratch with learning rate 10 times that of the lower layers. We adopt mini-batch SGD with momentum of 0.9 and batch size of 36. The learning rate is adjusted by $\eta_p = \eta_0(1+\alpha p)^{-\beta}$ where p is the training steps, $\eta_0 = 0.1, \alpha = 0.0001$ and $\beta = 0.75$. All models are trained for $20k$ iterations and we only report their *final* MAE on the target domain. We compare our method mainly with **DD** [30], which is designed for classification. We extend it to keypoint detection by replacing cross-entropy loss with L_{T}. The main regressor f and the adversarial regressor f' in **DD** and **our method** are both 2-layer convolutional neural networks with 256 channels.

Discussions Since each image has only one keypoint in *dSprites*, we cannot generate $\mathcal{P}_{\mathrm{F}}(\widehat{\boldsymbol{y}})$ according to Equation (10). Yet we find that for each image in the *dSprites*, keypoints only appear in the middle area $A = \{(h,w)|16 \le h \le 47, 16 \le w \le 47\}$. Therefore, we only assign positions inside A with positive probability,

$$\begin{aligned} \mathcal{H}_{\mathrm{F}}(\widehat{\boldsymbol{y}})_{h,w} &= \sum_{\boldsymbol{a}\in A, \boldsymbol{a}\neq\widehat{\boldsymbol{y}}} \mathcal{H}(\boldsymbol{a})_{h,w} \\ \mathcal{P}_{\mathrm{F}}(\widehat{\boldsymbol{y}})_{h,w} &= \frac{\mathcal{H}_{\mathrm{F}}(\widehat{\boldsymbol{y}})_{h,w}}{\sum_{h'=1}^{H'}\sum_{w'=1}^{W'}\mathcal{H}_{\mathrm{F}}(\widehat{\boldsymbol{y}})_{h',w'}}. \end{aligned} \tag{15}$$

We then minimize L_{F} to maximize the target disparity. Note that Equation (15) just narrows the original space from 64×64 to 32×32. However, this conversion from maximization to minimization has achieved significant performance gains on *dSprites*. Table 1 shows that this conversion reduces the error by **63**% in a relative sense.

Table 1. MAE on *dSprites* for different source and target domains (lower is better). The last row (oracle) corresponds to training on the target domain with supervised data (lower bound).

Method	C→N	C→S	N→C	N→S	S→C	S→N	Avg
ResNet18 [29]	0.495	0.256	0.371	0.639	0.030	0.090	0.314
DD [30]	0.037	0.078	0.054	0.239	0.020	0.044	0.079
RegDA	**0.020**	**0.028**	**0.019**	**0.069**	**0.014**	**0.022**	**0.029**
Oracle	0.016	0.022	0.014	0.022	0.014	0.016	0.017

We can conclude several things from this experiment:

1. The dimension of the output space has a huge impact on the minimax game of the adversarial regressor f'. As the output space enlarges, the maximization of f' would be increasingly difficult.

2. When the probability distribution of the output by f' is not uniform and our objective is to maximize the disparity on the target domain, minimizing the distance with this *ground false* distribution is more effective than maximizing the distance with the *ground truth*.

5.2. Experiment on Hand Keypoint Detection

5.2.1 Dataset

RHD Rendered Hand Pose Dataset (*RHD*) [34] is a synthetic dataset containing $41,258$ training images and $2,728$ testing images, which provides precise annotations for 21 hand keypoints. It covers a variety of viewpoints and difficult hand poses, yet hands in this dataset have very different appearances from those in reality (see Figure 7).

Figure 7. Some annotated images in the *RHD* dataset.

H3D Hand-3D-Studio (*H3D*) [31] is a real-world dataset of hand color images with 10 persons of different genders and skin colors, $22k$ frames in total. We randomly pick $3.2k$ frames as the testing set, and the remaining part is used as the training set. Since the images in *H3D* are sampled from videos, many images share high similarity in appearance. Thus models trained on the training set of *H3D* (**oracle**) achieve high accuracy on the test set. This sampling strategy is reasonable in the domain adaptation setup since we cannot access the labels on the target domain.

5.2.2 Training Details

We evaluate the performance of Simple Baseline [29] with ResNet101 [9] as the backbone.

Source only model is trained with L_2. All parameters are the optimal parameters under the supervised setup. The base learning rate is 1e-3. It drops to 1e-4 at 45 epochs and 1e-5 at 60 epochs. There are 70 epochs in total. Mini-batch size is 32. There are 500 steps for each epoch. Note that 70 epochs are completely enough for the models to converge both on the source and the target domains. Adam [13] optimizer is used (we find that SGD [1] optimizer will reach a very low accuracy when combined with L_2).

In our method, Simple Baseline is first trained with L_{T}, with the same learning rate scheduling as **source only**. Then the model is adopted as the feature generator ψ and trained with the proposed minimax game for another 30 epochs. The main regressor f and adversarial regressor f' are both 2-layer convolutional networks with width 256. The learning rate of the regressor is set 10 times to that of the feature generator, according to [6]. For optimization, we use the mini-batch SGD with the Nesterov momentum 0.9.

We compare our method to several feature-level domain adaptation methods, including **DAN** [17], **DANN** [6], **MCD** [22] and **DD** [30]. All methods are trained on the source domain for 70 epochs and then finetunes with the unlabeled data on the target domain for 30 epochs. We report the *final* PCK of all methods for a fair comparison.

5.2.3 Results

Percentage of Correct Keypoints (PCK) is used for evaluation. An estimation is considered correct if its distance from the ground truth is less than a fraction $\alpha = 0.05$ of the image size. We report the average PCK on all 21 keypoints. We also report PCK at different parts of the hand, such as metacarpophalangeal (MCP), proximal interphalangeal (PIP), distal interphalangeal (DIP), and fingertip.

The results are presented in Table 2. In our experiments, most existing domain adaptation methods do poorly on real keypoint detection tasks. They achieve a lower accuracy than **source only**, and their accuracy on the test set varies greatly during training. In comparison, our method has significantly improved the accuracy at all positions of hands, and the average accuracy has increased by **10.7**%.

Table 2. PCK on task *RHD*→*H3D*. The last row (oracle) corresponds to training on *H3D* with supervised data (upper bound on the domain adaptation performance). For all kinds of keypoints, our approach outperforms **source only** considerably.

Method	MCP	PIP	DIP	Fingertip	Avg
ResNet101 [29]	67.4	64.2	63.3	54.8	61.8
DAN [17]	59.0	57.0	56.3	48.4	55.1
DANN [6]	67.3	62.6	60.9	51.2	60.6
MCD [22]	59.1	56.1	54.7	46.9	54.6
DD [30]	72.7	69.6	66.2	54.4	65.2
RegDA	**79.6**	**74.4**	**71.2**	**62.9**	**72.5**
Oracle	97.7	97.2	95.7	92.5	95.8

We visualize the results before and after adaptation in Figure 8. As we mentioned in the introduction, the false predictions of **source only** are often located at the positions of other keypoints, resulting in the predicted skeleton not look like a human hand. To our surprise, although we did not impose a constraint (such as bone loss [32]) on the output of the model, the outputs of the adapted model (RegDA) look more like a human hand automatically.

Figure 8. Qualitative results of some images in the *H3D* dataset.

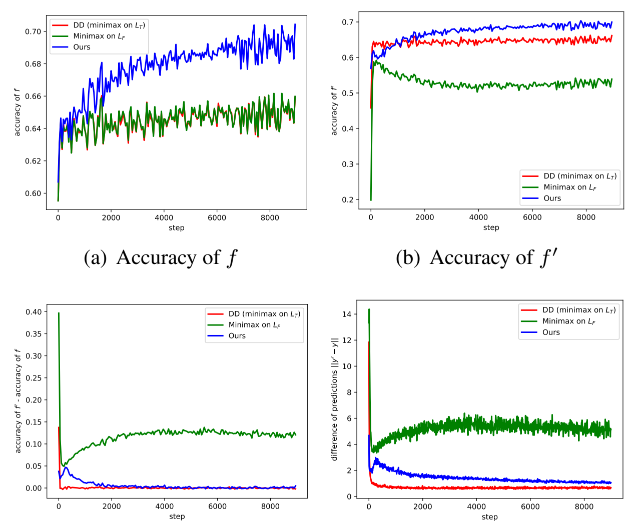
5.2.4 Ablation Study

We also conduct an ablation study to illustrate how minimization and maximization influences domain adaptation. Table 3 shows the results. The first row is **DD**, which plays the minimax game on L_{T}. The second row plays the minimax game on L_{F}. The last row is our method, which minimizes two opposite goals separately. Our proposed method outperforms the previous two methods by a large margin.

Table 3. Ablation study on the minimax of target disparity.

Method	f'	ψ	MCP	PIP	DIP	Fingertip	Avg
DD [30]	max L_{T}	min L_{T}	72.7	69.6	66.2	54.4	65.2
	min L_{F}	max L_{F}	74.4	71.1	66.9	56.4	66.5
RegDA	min L_{F}	min L_{T}	**79.6**	**74.4**	**71.2**	**62.9**	**72.5**

Figure 9 visualizes the training process. For **DD**, the difference in predictions $||\widehat{\boldsymbol{y}}' - \widehat{\boldsymbol{y}}||$ is small throughout training, which means that maximizing L_{T} will make the adversarial regressor f' weak. For methods that play minimax on L_{F}, the difference in predictions keeps enlarging and the performance of f' gradually drops. Thus, maximizing L_{F} will make the generator ψ too weak. In contrast, the prediction difference of our method increases at first and then gradually converges to zero during the adversarial training. As the training proceeds, the accuracy of both f and f' **steadily** increases on the target domain. Therefore, using two minimizations is the most effective way to realize adversarial training in a large discrete output space.



(a) Accuracy of f (b) Accuracy of f'

(c) Accuracy difference (d) Prediction difference

Figure 9. Empirical statistics during the training process.

5.3. Experiment on Human Keypoint Detection

We further evaluate our method on the human keypoint detection task. The training details are the same as 5.2.2.

5.3.1 Dataset

SURREAL *SURREAL* [26] is a synthetic dataset that consists of monocular videos of people in motion against indoor backgrounds (see Figure 10). There are more than 6 million frames in *SURREAL*.

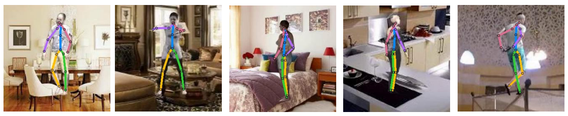


Figure 10. Some annotated images in the *SURREAL* dataset.

Human3.6M *Human3.6M* [11] is a large-scale real-world video dataset captured in indoor environments, with 3.6 million frames in total. It contains videos of human characters performing actions. We down-sampled the video from 50*fps* to 10*fps* to reduce redundancy. Following the standard in [15], we use 5 subjects (S1, S5, S6, S7, S8) for training and the rest 2 subjects (S9, S11) for testing.

LSP Leeds Sports Pose (*LSP*) [12] is a real-world dataset containing $2k$ images with annotated human body joint locations collected from sports activities. The images in *LSP* are captured in the wild, which look very different from those indoor synthetic images in *SURREAL*.

5.3.2 Results

For evaluation, we also use the PCK defined in 5.2.3. Since the keypoints defined by different datasets are different, we select the shared keypoints (such as shoulder, elbow, wrist, hip, knee) and report their PCK.

As shown in Tables 4 and 5, our RegDA method substantially outperforms **source only** at all positions of the body. The average accuracy has increased by **8.3%** and **10.7%** on *Human3.6M* and *LSP* respectively.

Figures 11 and 12 show the visualization results. The model before adaptation often fails to distinguish between left and right, and even hands and feet. Our RegDA method effectively help the model distinguish between different keypoints on the unlabeled domain.

6. Conclusion

In this paper, we proposed a novel method to enable regressive domain adaptation in keypoint detection, which utilizes the sparsity of the regression output space to help adversarial training in the high-dimensional space. We use a spatial probability distribution to guide the optimization of the adversarial regressor and perform the minimization of two opposite goals to solve the optimization difficulties. Extensive experiments are conducted on hand keypoint detection and human keypoint detection datasets. Our method surpasses the source only model by a large margin and outperforms state-of-the-art domain adaptation methods.

Table 4. PCK on task *SURREAL*→*Human3.6M*. Sld: shoulder, Elb: Elbow.

Method	Sld	Elb	Wrist	Hip	Knee	Ankle	Avg
ResNet101 [29]	69.4	75.4	66.4	37.9	77.3	77.7	67.3
DAN [17]	68.1	77.5	62.3	30.4	78.4	79.4	66.0
DANN [6]	66.2	73.1	61.8	35.4	75.0	73.8	64.2
MCD [22]	60.3	63.6	45.0	28.7	63.7	65.4	54.5
DD [30]	71.6	83.3	75.1	42.1	76.2	76.1	70.7
RegDA	**73.3**	**86.4**	**72.8**	**54.8**	**82.0**	**84.4**	**75.6**
Oracle	95.3	91.8	86.9	95.6	94.1	93.6	92.9

Table 5. PCK on task *SURREAL*→*LSP*. Sld: shoulder, Elb: Elbow.

Method	Sld	Elb	Wrist	Hip	Knee	Ankle	Avg
ResNet101 [29]	51.5	65.0	62.9	68.0	68.7	67.4	63.9
DAN [17]	52.2	62.9	58.9	71.0	68.1	65.1	63.0
DANN [6]	50.2	62.4	58.8	67.7	66.3	65.2	61.8
MCD [22]	46.2	53.4	46.1	57.7	53.9	52.1	51.6
DD [30]	28.4	65.9	56.8	75.0	74.3	73.9	62.4
RegDA	**62.7**	**76.7**	**71.1**	**81.0**	**80.3**	**75.3**	**74.6**

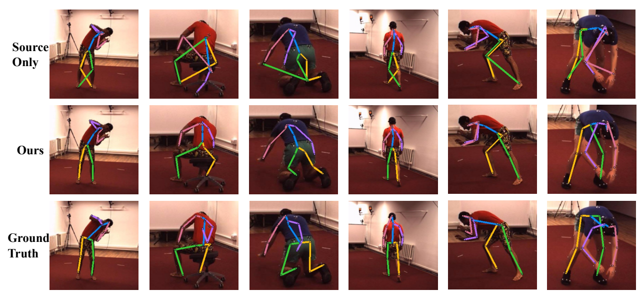


Figure 11. Qualitative results of some images in the *Human3.6M* dataset. Note that the keypoints on the blue lines are not shared between different datasets.

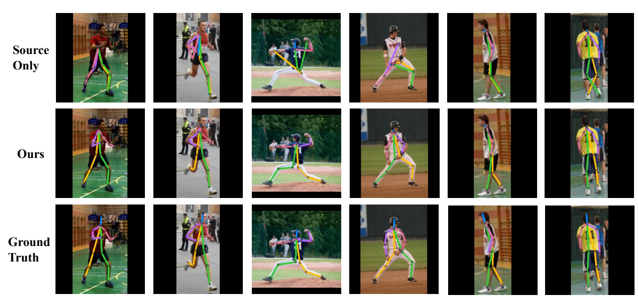


Figure 12. Qualitative results of some images in the *LSP* dataset.

Acknowledgments

This work was supported by National Key R&D Program of China (2020AAA0109201), NSFC grants (62022050, 62021002, 61772299, 71690231), Beijing Nova Program (Z201100006820041), MOE Innovation Plan of China, and Kuaishou Technology Fund.

References

[1] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, pages 185–196, 1993.

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[3] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018.

[4] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1212–1221, 2017.

[5] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2017.

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[7] Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in neural information processing systems*, pages 1205–1213, 2012.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.

[11] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[12] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *bmvc*, volume 2, page 5. Citeseer, 2010.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[15] Sijin Li and Antoni B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *ACCV*, 2014.

[16] Wenbo Li, Zhicheng Wang, Binyi Yin, Qixiang Peng, Yuming Du, Tianzi Xiao, Gang Yu, Hongtao Lu, Yichen Wei, and Jian Sun. Rethinking on multi-stage networks for human pose estimation. *arXiv preprint arXiv:1901.00148*, 2019.

[17] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[18] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[19] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017.

[20] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and N Lawrence. Covariate shift and local learning by distribution matching, 2008.

[21] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[22] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[23] Baochen Sun and Kate Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *BMVC*, volume 1, page 3, 2014.

[24] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5693–5703, 2019.

[25] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional

network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014.

[26] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017.

[27] David Vazquez, Antonio M Lopez, Javier Marin, Daniel Ponsa, and David Geronimo. Virtual and real world adaptation for pedestrian detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(4):797–809, 2013.

[28] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[29] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. *arXiv e-prints*, 2018.

[30] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging theory and algorithm for domain adaptation. In *ICML*, pages 7404–7413, 2019.

[31] Zhengyi Zhao, Tianyao Wang, Siyu Xia, and Yangang Wang. Hand-3d-studio: A new multi-view system for 3d hand reconstruction. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2478–2482. IEEE, 2020.

[32] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 398–407, 2017.

[33] Xingyi Zhou, Arjun Karpur, Chuang Gan, Linjie Luo, and Qixing Huang. Unsupervised domain adaptation for 3d keypoint estimation via view consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 137–153, 2018.

[34] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017.