Structured Domain Adaptation for 3D Keypoint Estimation

Levi O. Vasconcelos[1,2] Massimiliano Mancini[2,3,4] Davide Boscaini[4] Barbara Caputo[2,5] Elisa Ricci[1,4]
[1]University of Trento [2]Italian Institute of Technology [3]Sapienza University of Rome
[4]Fondazione Bruno Kessler [5]Politecnico di Torino
l.osternovasconcelos@unitn.it, {mancini, dboscaini, eliricci}@fbk.eu, barbara.caputo@polito.it

Abstract

Motivated by recent advances in deep domain adaptation, this paper introduces a deep architecture for estimating 3D keypoints when the training (source) and the test (target) images greatly differ in terms of visual appearance (domain shift). Our approach operates by promoting domain distribution alignment in the feature space adopting batch normalization-based techniques. Furthermore, we propose to collect statistics about 3D keypoints positions of the source training data and to use this prior information to constrain predictions on the target domain introducing a loss derived from Multidimensional Scaling. We conduct an extensive experimental evaluation considering three publicly available benchmarks and show that our approach outperforms state-of-the-art domain adaptation methods for 3D keypoints predictions.

1. Introduction

The ability to accurately estimate 3D keypoints is fundamental for many tasks, such as pose estimation [43], object recognition [26] and 3D reconstruction [16]. The widespread diffusion of consumer devices equipped with depth sensors has brought up the need for methods able to detect 3D keypoints from depth scans. Within this context, a practical issue is the lack of labelled data. Indeed, 3D keypoints annotations (*e.g.* joints of a skeleton) are not only very costly but also hard to obtain, as depth scans typically correspond to partial views of the underlying objects [10].

A solution would be to employ synthetic data, for which annotations are inherently available [51]. Ideally, we would like to train a model on synthetic data and readily apply it to RGB images or depth scans of the real world. Unfortunately, this is not possible due to the *domain shift* [42], *i.e.* the discrepancy among the train (*source*) and test (*target*) data distributions, which usually implies a degradation of the recognition models performances when applied to the target data. This issue is observed not only when going from synthetic to real data, but also when train and test data correspond to different modalities and/or different environmental conditions (*e.g.* images collected at different times of the day).

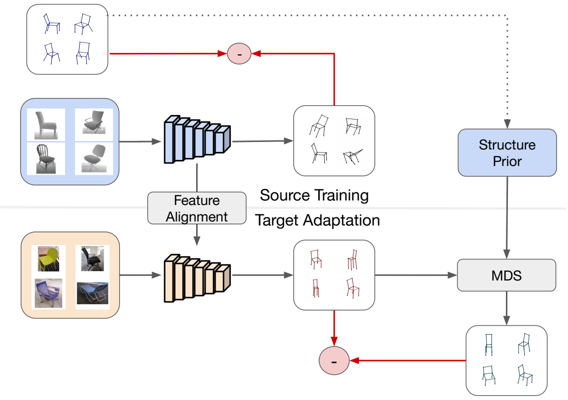


Figure 1. Overview of our approach. We train a model on the source domain in a supervised manner, using the available ground truth information. We use the same ground truth to compute statistics regarding the structure of the output space (structure prior). We then adapt the model to the target domain by (i) performing feature alignment and (ii) exploiting the structure prior to impose a consistency loss on the predictions using Multidimensional Scaling. Red circles and arrows represent loss computations.

To cope with the domain shift, various Domain Adaptation (DA) approaches [46] have been proposed, with a particular focus on object recognition tasks. Typical strategies tries to align the distributions of the features learned at different levels of the deep architecture by using moment matching techniques [5, 37], domain adversarial training [21, 45, 15], or generative models [3, 19, 39].

To our knowledge, in the context of 3D keypoints estimation, the only work addressing the domain shift problem is [50]. There, Zhou *et al.* proposed a deep architecture that minimizes a loss made of three terms: (i) a supervised loss term operating on source data and promoting predictions to be close to the ground-truth labels, (ii) a multi-view consistency loss term encouraging predictions from different views of the same objects to be consistent, and (iii) a geometric alignment term enforcing the alignments of the source and target 3D keypoints distributions. While worthy, this work has two main limitations. First, while the consistency loss is highly beneficial in terms of performance, it assumes multiple views of the same object. Unfortunately, this information may not always be available in practical applications. Second, the geometric alignment loss requires the presence of source data when adapting the model to the target in order to provide prior information on the pose of the estimated object. This has a negative impact both in terms of memory requirements and computational cost.

The DA approach we propose tackles both issues. Our main idea is to extract meaningful statistics from the structured output space of 3D keypoints configurations which are invariant to the domain shift. Specifically, we collect statistics about distances between the 3D keypoints predicted on the source domain (and their ratios) through a supervised learning procedure; we then enforce consistency to such statistics when predicting 3D keypoints on the target samples adopting an approach derived from Multidimensional Scaling (MDS) [25]. Additionally, we propose to complement the proposed MDS-based method and further cope with the domain shift by introducing a feature alignment procedure derived from batch normalization [30]. An overview of our method is depicted in Fig. 1.

We evaluate our approach for predicting 3D keypoints of rigid objects (chairs), adapting from rendered depth scans in ModelNet [47] to real data in Redwood Object Scans [10], as well as non-rigid shapes (humans), using the Human3.6M dataset [23]. Experiments show that we outperform [50] as well as general-purpose DA approaches [45].

To summarize, the contributions of this work are: (i) We introduce a DA approach for 3D keypoints estimation which does not require source samples during the adaptation phase or specific assumptions on the given images (*e.g.* multiple views available); (ii) We show how prior information about the 3D object structure of source data can be exploited for the purpose of DA for 3D keypoints estimation; (iii) We test our algorithm both on non-deformable and deformable objects, showing that it is more accurate than [50] while being also more computationally efficient.

2. Related Work

3D Keypoints Estimation. Several approaches addressed 3D keypoint detection from 2D images [8, 32, 33, 41]. These works employ CNNs to compute 3D keypoints from monocular RGB images with varying network architectures, loss functions and strategies for extracting 3D structural prior information. For instance, [41] proposed to find 3D keypoints by enforcing a projection consistency term between an image and its rigidly-transformed counterpart. Chen *et al.* [8], instead, introduced a fully convolutional architecture trained within an adversarial framework.

Other works adopt the so-called strategy of "2D-to-3D-lifting", *i.e.* leverage from 2D keypoints detectors and learned 3D prior information in order to infer the 3D keypoints coordinates [44, 7, 36]. For instance, Chen *et al.* [7] proposed to reason through intermediate 2D pose predictions. Specifically, they lift 2D keypoints predicted by a deep model by using a nearest neighbour search on a prior collection of 3D poses. The 3D poses are projected into the camera frame and the $L2$ norm of the difference between projected and predicted keypoints is used as a distance metric. Tung *et al.* [44], instead, introduced Adversarial Inverse Graphics Networks (AIGNs) to combine information from 2D predictions and 3D priors and apply their model to 3D human pose estimation.

Domain Adaptation. DA tackles the problem of rectifying the discrepancy between source and target distributions. As argued by Ben *et al.* [1], the feature representation space plays an important role in DA approaches. A common practice is to seek for a feature mapping that aligns source and target marginal distributions. To that end, two main strategies are commonly considered: Moment Matching [5, 37, 6] and Adversarial Training [28, 21, 45, 15].

Recent developments in DA research proposed different strategies to align domain distributions, but the vast majority of them focus on classification tasks only [46]. Surprisingly, despite their relevance for computer vision tasks, DA methods for structured output prediction problems have received little attention. Notable exceptions include DA methods for semantic segmentation [21, 48, 20, 9], depth prediction [49], and 3D keypoints estimation [50]. The latter being the more relevant to the proposed approach.

In particular, Zhou *et al.* [50] addressed the DA problem in the context of 3D keypoints prediction by optimizing a loss made of two terms: (i) a consistency term and a Chamfer distance term. The consistency term enforces that the predicted 3D keypoints of the same scene from different views must differ only up to a rotation. The Chamfer term is used to align the posterior distributions of source and target datasets. However, this method needs multiple views of the same object, which can be very expensive to obtain especially in case of non-rigid subjects such as humans. In this work, we also focus on the problem of structured DA for 3D keypoints estimation from 2D images. However, differently from [50], our approach does not require multiple views. Furthermore, source samples are not needed during the adaptation phase, thus enabling knowledge transfer in a lightweight manner.

3. Preliminaries: Multidimensional Scaling

Before delving into the actual contribution, we briefly review Multidimensional Scaling (MDS) first. MDS [25, 11, 2] is a family of methods addressing the problem of finding the optimal embedding of a set of (potentially high-dimensional) data points into a (low-dimensional) Euclidean space, having only information about their dissimilarity. This information is typically provided in terms of pairwise distances. Among the possible embeddings, MDS techniques finds the one which distorts the distances the least. More specifically, given a set of k points, the ith D-dimensional feature $\boldsymbol{f}_i$ is associated to the d-dimensional embedding coordinates $\boldsymbol{y}_i \in \mathbb{R}^d$, $d \leq D$, by minimizing the *stress* function:

$$\sigma(\boldsymbol{Y}) = \sum_{i>j} \left(d_{ij}(\boldsymbol{Y}) - \delta_{ij}\right)^2, \tag{1}$$

where $\boldsymbol{Y} = (\boldsymbol{y}_i) \in \mathbb{R}^{k \times d}$ is a matrix containing the d-dimensional coordinates of the k embeddings, $d_{ij}(\boldsymbol{Y}) = d_{\mathbb{R}^d}(\boldsymbol{y}_i, \boldsymbol{y}_j)$ is the Euclidean distance between them, and $\delta_{ij} = d_{\mathbb{R}^D}(\boldsymbol{f}_i, \boldsymbol{f}_j)$ is the given distance between the ith and jth features. Conveniently, Eq. (1) can be rewritten in matrix notation as:

$$\sigma(\boldsymbol{Y}) = \mathrm{Tr}\left(\boldsymbol{Y}\boldsymbol{V}\boldsymbol{Y}^\top\right) - 2\ \mathrm{Tr}\left(\boldsymbol{Y}^\top \boldsymbol{B}(\boldsymbol{Y})\boldsymbol{Y}\right) + \sum_{i>j} \delta_{ij}^2,$$

where Tr denotes the matrix trace, and $\boldsymbol{V}, \boldsymbol{B}(\boldsymbol{Y})$ are $k \times k$ matrices whose entries are defined as:

$$v_{ij} = \begin{cases} -1 & i \neq j, \\ k-1 & i = j, \end{cases}$$

and:

$$b_{ij}(\boldsymbol{Y}) = \begin{cases} -\delta_{ij}/d_{ij}(\boldsymbol{Y}) & i \neq j \text{ and } d_{ij}(\boldsymbol{Y}) \neq 0, \\ 0 & i \neq j \text{ and } d_{ij}(\boldsymbol{Y}) = 0, \\ -\sum_{i=1, i\neq j}^{k} b_{ij}(\boldsymbol{Y}) & i = j. \end{cases}$$

The optimization of Eq. (1) is rather difficult because of the nonlinearity of the second term. To tackle this issue, de Leeuw [12] proposed to optimize the quadratic majorization function:

$$\tau(\boldsymbol{Y}, \boldsymbol{Z}) = \mathrm{Tr}\left(\boldsymbol{Y}\boldsymbol{V}\boldsymbol{Y}^\top\right) - 2\ \mathrm{Tr}\left(\boldsymbol{Y}^\top \boldsymbol{B}(\boldsymbol{Z})\boldsymbol{Z}\right) + \sum_{i>j} \delta_{ij}^2$$

instead, where $\tau(\boldsymbol{Y}, \boldsymbol{Z}) \geq \sigma(\boldsymbol{Y})$ by virtue of the Cauchy-Schwarz inequality. More specifically, de Leeuw [12] proposed an iterative majorization method, called SMACOF, which is guaranteed to decrease the stress $\sigma(\boldsymbol{Y})$ monotonically. At each step of the SMACOF, $\boldsymbol{Z} = \boldsymbol{Y}^{(t-1)}$ is initialized and the quadratic loss $\tau(\boldsymbol{Y}, \boldsymbol{Z})$ is optimized, *i.e.*:

$$\boldsymbol{Y}^{(t)} = \arg\min_{\boldsymbol{Y}} \tau(\boldsymbol{Y}, \boldsymbol{Z}). \tag{2}$$

The iterations continue until $\|\boldsymbol{Y}^{(t)} - \boldsymbol{Y}^{(t-1)}\| < \epsilon$, for a given threshold ϵ. Eq. (2) admits a closed form solution:

$$\boldsymbol{Y}^{(t)} = \frac{1}{k} \boldsymbol{B}(\boldsymbol{Y}^{(t-1)})\boldsymbol{Y}^{(t)}. \tag{3}$$

4. Proposed Approach

4.1. Problem and Notation

We deal with the problem of unsupervised domain adaptation, which can be formalized as follows. Let us denote as $\mathcal{X}$ the input space (*e.g.* RGB images, depth scans) and as $\mathcal{Y}$ the output space (*e.g.* semantic categories, 3D coordinates). We have two domains $\mathcal{S}$ and $\mathcal{T}$, namely the *source* and the *target* respectively. The source domain is composed by labelled samples, *i.e.* $\mathcal{S} = \{(x_1^s, \boldsymbol{Y}_1^s), \ldots, (x_n^s, \boldsymbol{Y}_n^s)\}$, while for the target domain only unlabelled samples are available, *i.e.* $\mathcal{T} = \{x_1^t, \ldots, x_m^t\}$. The two domains are characterized by two different distributions over $\mathcal{X} \times \mathcal{Y}$, namely: $\mathrm{p}_{xy}^{\mathcal{S}}$ and $\mathrm{p}_{xy}^{\mathcal{T}}$ with $\mathrm{p}_{xy}^{\mathcal{S}} \neq \mathrm{p}_{xy}^{\mathcal{T}}$. The latter inequality represents the core of the domain shift problem and is the reason why, a model trained on $\mathcal{S}$, cannot perform well on $\mathcal{T}$ without explicitly taking into account the discrepancy among the two domains. Our goal is to overcome the domain shift, building a prediction model $f_\Theta : \mathcal{X} \to \mathcal{Y}$, for the target domain, where Θ denote the model parameters.

A discrepancy among the domains may arise at the feature level, due to the different marginals on the input space $\mathrm{p}_x^{\mathcal{S}}$ and $\mathrm{p}_x^{\mathcal{T}}$. This difference can be caused by *e.g.* different acquisition sensors or illumination conditions. While we can align $\mathrm{p}_x^{\mathcal{S}}$ and $\mathrm{p}_x^{\mathcal{T}}$ in multiple ways (see *e.g.* [14, 29, 4, 5]), even an optimal alignment among the two may not guarantee to solve the domain shift issue. In fact, the optimal alignment is achieved only if we are able to match the conditional distributions $\mathrm{p}_{y|x}^{\mathcal{S}}$ and $\mathrm{p}_{y|x}^{\mathcal{T}}$. The problem is that in unsupervised domain adaptation we have no access to $\mathrm{p}_{y|x}^{\mathcal{T}}$, as the samples in $\mathcal{T}$ have no labels. However, it is fundamental to try to model $\mathrm{p}_{y|x}^{\mathcal{T}}$ and design appropriate loss functions for the target samples.

In the context of classification, a semantic loss on target data can be imposed by considering the confidence of the classifier, *i.e.* by applying an entropy loss [5, 34], or by employing consistency terms [13, 37]. However, these approaches cannot be used for *structured output* prediction tasks, as in this paper. Here, in fact, we deal with the more complex problem of predicting the keypoints of a 3D shape from 2D images. Specifically, given a sample, our goal is to predict a set of 3D coordinates $\boldsymbol{Y} \in \mathbb{R}^{k \times 3}$, where k is the number of keypoints. For this task is hard to define a semantic loss on the samples in $\mathcal{T}$, since the problem requires to predict the positions of a set of points in a multidimensional and continuous space. However, our output space is structured, because the position of a keypoint is not independent from the position of the others. Following this intuition and reasoning about the structure of the output space, we propose a loss function to optimize over target samples for the purpose of adaptation.

4.2. Structured DA for 3D Keypoints Prediction

Overview. Our approach operates in two main phases (see Figure1). In a preliminary phase a keypoints predictor f_Θ is trained on labelled source data. Furthermore, statistics are extracted from the labels in the source domain $\mathcal{S}$ and used as prior knowledge of the structure of the label space.

Then, in the alignment training phase a two-step procedure is performed. The first alignment is performed at the feature level, where we apply DA techniques [30] derived from Batch Normalization (BN) [22]. The second alignment considers the output space $\mathcal{Y}$. We propose a novel consistency loss between the network predictions and a synthesized set of keypoints. The latter set is obtained by exploiting the pre-computed structure prior and MDS. In the following we describe the details of our method.

Source Pre-training and Prior Information Extraction. The first phase of our approach aims to learn a model f_Θ (*e.g.* a deep neural network) trained on the source domain $\mathcal{S}$ which, given an image outputs a set of k 3D keypoints, *i.e.* $\boldsymbol{Y} \in \mathbb{R}^{k\times 3}$. Since $\mathcal{S}$ contains labeled data, the parameters Θ can be obtained with standard regression. Given a sample x_i^s we can minimize:

$$\ell_\Theta(x_i^s, \boldsymbol{Y}_i^s) = \|f_\Theta(x_i^s) - \boldsymbol{Y}_i^s\|_2. \tag{4}$$

The learned model f_Θ is used to initialize the predictor on the target domain.

Besides for learning the source predictor, our approach also exploits source data for modelling the structure of the output space $\mathcal{Y}$. Let us assume that $\mathcal{S}$ and $\mathcal{T}$ contain images (*e.g.* RGB or depth) of 3D shapes of the same category (*e.g.* chairs). We assume that the overall shape of an object does not change significantly, regardless of the specific instance considered. For example, we do not expect a rail of a chair to be longer than its back post and, similarly, we do not expect the leg of a human to be shorter than his/her arms. Under this assumption, it is reasonable to consider the relative distances among keypoints as a suitable model to describe the structure of an object. Given a sample x, let us define the distance among two keypoints i, j with coordinates y_i and y_j as $d_{ij}^x = \|y_i - y_j\|_2$. Obviously, if we take two random chairs, the probability that they share exactly the same dimensions is very low (*i.e.* usually $d_{ij}^x \neq d_{ij}^z$ if $x \neq z$). However, it is very likely that, regardless of the chair, the right and left arm have the same dimensions (*i.e.* if (i, j) are the keypoints of the left arm and (l, m) the keypoints of the right arm, we usually have $d_{ij}^x = d_{lm}^x$ and $d_{ij}^z = d_{lm}^z$). Similar considerations apply *e.g.* to the legs. This prior information (which we can find both in deformable and non-deformable objects), is captured by the ratio among the distances. Following this intuition we propose to take the ratios among keypoints distances as structure prior modelling the output space.

Let us denote as $r_{ij,lm}^x = d_{ij}^x / d_{lm}^x$ the ratio among the distances d_{ij} and d_{lm}. In the following the superscript x is removed for the sake of clarity. We define as $\boldsymbol{R}(\boldsymbol{Y}) = (r_{ij,lm}) \in \mathbb{R}^{k^2\times k^2}$ the ratio matrix relative to a set of keypoints $\boldsymbol{Y}$. We propose to exploit the labels in $\mathcal{S}$ in order to extract statistics about the ratios. While higher order statistics can be used, in this work we consider the average ratio matrix on the labels of $\mathcal{S}$, defined as $\mu_{\boldsymbol{R}}^s = \mathbb{E}_{\boldsymbol{Y}\sim\mathcal{S}}[\boldsymbol{R}(\boldsymbol{Y})]$.

We underline that the source data are needed to (i) initialize the model and (ii) extract the prior information about the structure of $\mathcal{Y}$. We *do not need* source samples while adapting the model to the target domain. This is greatly beneficial in practical applications as multiple target predictors for different scenarios can be learned starting from the same source model. Furthermore, we do not require to store source data and adaptation is computationally efficient.

Aligning Features with Batch Normalization. The source model f_Θ is biased towards its underlying data distribution, namely $\mathrm{p}_x^{\mathcal{S}}$. However, when processing the target data, a new marginal distribution $\mathrm{p}_x^{\mathcal{T}}$ arises. In order to reduce the discrepancy among $\mathrm{p}_x^{\mathcal{S}}$ and $\mathrm{p}_x^{\mathcal{T}}$ we introduce a first alignment procedure, acting at the feature level. To this extent, we adopt a simple yet effective strategy based on maintaining domain-specific BN statistics. BN operates to normalize deep features to have zero mean and standard deviation equal to one. By using domain-specific statistics, we ensure that the feature distribution of different domains are matched to the same reference distribution, thus reducing the domain shift. This strategy has been successfully employed by various DA algorithms [27, 5, 31]. We use this approach to promote alignment at the feature level and to facilitate adaptation at the output space level, as we describe in the next paragraph.

Structured DA through MDS. As highlighted in Sec. 4.1, performing domain adaptation only at the feature level is suboptimal. In fact, an optimal alignment of the two domains can be achieved only considering conditional distributions $\mathrm{p}_{y|x}^{\mathcal{S}}$ and $\mathrm{p}_{y|x}^{\mathcal{T}}$. To this extent, the second phase of the proposed approach operates at prediction level, *i.e.* in the output space. Given the matrix $\mu_{\boldsymbol{R}}^s$ modelling the structure of the source output space, a possible approach to force predictions consistent with the structure prior is to formulate a regression problem on target samples. Specifically, given a target sample x^t, the following loss function can be defined:

$$\ell_\Theta(\hat{\boldsymbol{Y}}) = \|\boldsymbol{R}(\hat{\boldsymbol{Y}}) - \mu_{\boldsymbol{R}}^s\|_2, \tag{5}$$

where $\hat{\boldsymbol{Y}} = f_\Theta(x^t)$. However, this objective is hard to optimize because it is defined over a very high dimensional space (on the order of k^4). To address this issue, we devise a strategy to scale down from the ratio matrix space to a lower dimensional space, the output space $\boldsymbol{Y} \in \mathbb{R}^{k\times 3}$. In order to do this, we resort to MDS.

The input required by MDS are an initial set of points, *i.e.* $\boldsymbol{Y}$ in Eq. (1) and the reference distances δ_{ij}. Given a target image x^t the initial set of points can be obtained through the network, *i.e.* $\hat{\boldsymbol{Y}} = f_\Theta(x^t)$. However, since we have no supervision on $\mathcal{T}$, we lack any information about the reference distances δ_{ij}. To this extent, we must define a way to extract δ_{ij} given the prior $\mu^s_{\boldsymbol{R}}$ and the predictions $\hat{\boldsymbol{Y}}$. Obviously, due to the lack of supervision, we cannot compute how close our estimate of δ_{ij} is to the true distance among the keypoints i and j, no matter the function we use. For this reason, the MDS-based objective we propose and we describe below can be regarded as a function acting as a *consistency* term.

To define δ_{ij} we follow a simple intuition. Suppose we have a chair and we know the length of all its components but the left front leg. For us, as humans, it is quite easy to infer the length of this component, since we know *e.g.* that the left front leg has usually the same length of the right front leg. To reach this conclusion we use our prior knowledge about the chair structure. Obliviously, similar reasoning can be applied also to other objects as well. Our approach is derived from this intuition. Formally, let us assume to have the ground truth distances d^*_{ij} and d^*_{lm} corresponding to two edges of an object. These distances are related, *i.e.* they are not independently defined in a single object. It is easy to observe that this relation holds:

$$d^*_{ij} = \frac{d^*_{ij}}{d^*_{lm}} \cdot d^*_{lm} \approx \mu^s_{\boldsymbol{R}}(ij, lm) \cdot d^*_{lm}, \tag{6}$$

where $\mu^s_{\boldsymbol{R}}(ij, lm) = \mathbb{E}_{\mathcal{S}}[r_{ij,lm}]$. Under this perspective, we define a loss forcing the distance among the edges i and j obtained from our model to be consistent with the same distance as estimated by using the source prior information and all the other distances among keypoints. We achieve this by computing δ_{ij} as:

$$\delta_{ij} = \frac{1}{G_{ij}} \sum_{\substack{l,m=1 \\ l \neq m \\ (i,j) \neq (l,m)}}^{k} g(ij, lm) \cdot \mu^s_{\boldsymbol{R}}(ij, lm) \cdot d_{lm} \tag{7}$$

where $g(ij, lm)$ is a scalar denoting the reliability of $\mu^s_{\boldsymbol{R}}(ij, lm)$ in order to estimate d_{ij} from d_{lm}, and $G_{ij} = \sum^k_{l,m=1} g(ij, lm)$ is a normalization factor. The rationale behind Eq. (7) is that we should be able to obtain an estimate δ_{ij} of d_{ij}, given the expected ratios and all the other distances of the object shape (as for Eq.(6)) and that $\delta_{ij} \approx d_{ij}(\hat{\boldsymbol{Y}})$. Obviously, the choice of how to compute $g(ij, lm)$ has an impact on the performances, because it allows to filter out non reliable ratios/relations (*e.g.* height of the back post and width of the chair apron). We investigated various choices for $g(ij, lm)$ in the experimental section.

Once defined δ_{ij}, we can derive the loss for target samples using the MDS formulation. Given an initial estimate of the keypoints coordinates $\hat{\boldsymbol{Y}}$ and the input distances δ_{ij}, we can minimize Eq. (1) directly. However, we choose to define a simple regression loss over the output of the SMACOF algorithm (Eq. (3)):

$$\ell_\Theta(\boldsymbol{Y}^{(T)}) = \|\hat{\boldsymbol{Y}} - \boldsymbol{Y}^{(T)}\|_2. \tag{8}$$

Algorithm 1 DA through MDS

```
Input: Source model $f_\Theta$, Target data $\mathcal{T}$, Average ratio matrix
    $\mu^s_{\boldsymbol{R}}$, Reliability function $g$, number of SMACOF iterations $T$,
    number of keypoints $k$
Output: Adapted model parameters $\Theta$
for each image $x^t \in \mathcal{T}$
    Compute predicted keypoints: $\hat{\mathbf{Y}} = f_\Theta(x^t)$
    Compute distances among predicted keypoints: $d_{ij}(\hat{\mathbf{Y}})$
    Compute $\delta_{ij}$ through $\mu^s_{\boldsymbol{R}}$ and $d_{ij}(\hat{\mathbf{Y}})$
    Initialize keypoints for MDS: $\mathbf{Y}^{(0)} \leftarrow \hat{\mathbf{Y}}$
    for $t = 1$ to $t = T$
        Apply MDS iteration: $\boldsymbol{Y}^{(t)} = \frac{1}{k}\boldsymbol{B}(\boldsymbol{Y}^{(t-1)})\boldsymbol{Y}^{(t)}$
    Compute loss: $\ell_\Theta(\boldsymbol{Y}^{(T)}) = \|\hat{\boldsymbol{Y}} - \boldsymbol{Y}^{(T)}\|_2$
    Backpropagate $\ell_\Theta$ through $\hat{\boldsymbol{Y}}$and $\boldsymbol{Y}^{(T)}$
    Update $\Theta$
return $\Theta$
```

where T is the number of SMACOF iterations and is set as hyper-parameter. We found the latter to be more stable and yielding to better performances in our experiments. The whole adaptation procedure is summarized in algorithm 1.

5. Experiments

5.1. Experimental setup

Datasets. We evaluate our method on the datasets: ModelNet [47], Redwood [10] and Human3.6M [23].

ModelNet [47] consists of a collection of 3D CAD models from several object categories. Following [50], we use rendered depth scans of chairs, taking the pre-processed images and labels provided by the authors of [50].

Redwood [10] dataset is composed of RGB-D scans of real objects. Following [50], we use RGB and depth images of chair models taking their pre-processed data and labels. We use Redwood and ModelNet for predicting the coordinates of 10 keypoints.

Human3.6M [23] contains 3.6 millions human poses, collected by recording different action sequences of 11 actors, with the human joint locations obtained through motion capture devices. We use this dataset to compare our method with [50] in a human pose estimation task, using RGB and depth images as input data. We consider the protocol 2 described in [40], considering the 7 labelled subjects for our experiments. In particular, independently on the specific setting, we use subjects (S1,S5,S6) as training set of the source domain, subjects (S7,S8) as target training set and subjects (S9,S11) as target test set. We use subjects (S1,S5,S6), but considering the same modality of the target domain, as validation set. We further subsample the dataset by picking, at random, for subjects (S1, S5, S6) 2000 images for each subject; and 375 images for each of the remaining subjects. For the output space we consider 15 joint positions (14, as in [24] plus the pelvis), applying the transformation to the label space described in [40].

Training protocols. As in [50], we use a pre-trained ResNet50 [17] on ImageNet [38] as base architecture, replacing its final layer with a fully-connected layer having the output dimension equal to $3 \times k$, where 3 are the coordinates of the keypoints and k the number of keypoints (*i.e.* 10 for ModelNet and Redwood and 15 for Human3.6M). We ran our experiments on a GeForce RTX 2080 Ti, implementation was done with the PyTorch[1] framework [35] and results reported as average across 5 runs. Following [50], we evaluate the results using two metrics: the average distance error (AE) between the predicted keypoint positions and the ground truth, and the pose-invariant average error (PAE) which measures the same quantity of AE but after aligning the predicted keypoints and the ground truth through a rotation.

For the experiments on chairs, we use ModelNet as source domain and Redwood (either RGB or depth images) as target. For ensuring a fair comparison we use the pretrained model provided by [50]. For training our model on the target domain, we employ similar hyperparameters with respect to [50], training our model for 30 epochs with a batch size of 32 and a learning rate of $2 \cdot 10^{-4}$, decaying the learning rate of a 0.1 factor after 20 epochs.

For the experiments on Human3.6M, we train our source model (either on RGB or depth images) for 40 epochs, with a batch size of 64, a weight decay of $1e^{-4}$ and a learning rate of $3 \cdot 10^{-2}$, dropping its value by 0.1 every 40 epochs. During the adaptation phase, we use the same hyperparameters of the previous experiments, except for the batch-size, set to 64 and for the initial learning rate, set to $1 \cdot 10^{-5}$.

5.2. Results

Analysis of our approach. We first perform an analysis to investigate the impact of various design choices in our algorithm. We analyze different choices for (i) loss function and (ii) the function $g(ij, lm)$. We perform our analysis on the chairs datasets, considering two domain adaptation setting: synthetic depth (ModelNet) to real depth (Redwood depth) and synthetic depth to real RGB (Redwood RGB).

We first examine the performance with different loss functions. As baseline, we consider (i) the source model, (ii) a model performing just the feature alignment step through BN, as in [30]. For what concerns the loss itself we consider 3 possible choices. The first is a simple L2 loss among the distances of the keypoints, as predicted by the network, and the distances δ_{ij} we obtain through Eq. (7). The second is optimizing the MDS objective directly, as defined in Eq. (2), using the predicted keypoints and the distances δ_{ij}. The third option is to have the loss defined as an L2 loss between the predicted keypoints and the keypoints obtained after T iterations of the SMACOF algorithm (Eq. (8)). We use the same training hyperparameters for all the alternatives, setting $T = 10$ and $g(ij, lm) = 1$ (T is set to 10 in all experiments on chairs and to 100 for experiments on human poses, see supplementary material for the corresponding ablation).

Table 1. ModelNet to Redwood: ablation on the semantic loss.

$\mathcal{S}$	ModelNet (depth)			
$\mathcal{T}$	Redwood (depth)		Redwood (RGB)	
Metric	AE	PAE	AE	PAE
Baseline	16.01	10.73	27.59	13.44
Batch Norm [30]	14.74	9.87	25.33	12.43
L2 norm	15.02	11.87	23.23	12.89
MDS direct	13.44	9.45	23.24	10.92
MDS SMACOF	**12.33**	**7.76**	**22.86**	**10.20**

Table 2. ModelNet to Redwood: ablation on $g(ij, lm)$.

$\mathcal{S}$	ModelNet (depth)			
$\mathcal{T}$	Redwood (depth)		Redwood (RGB)	
Metric	AE	PAE	AE	PAE
Baseline	16.01	10.73	27.59	13.44
All joints	12.33	7.76	22.86	10.20
Bones	12.27	7.68	23.02	10.02
Bones + std	**12.18**	**7.62**	**22.68**	10.11
Bones + corr.	12.41	7.82	22.92	**9.99**

The results are reported in Table 1. As shown in the Table, only applying the BN-based adaptation strategy allows to improve the performances with respect to the baseline model, with a clear gain in the depth to RGB setting, where the shift is larger. Among the various choices of the losses applying MDS-based losses allows to obtain a boost in the performances in both depth and RGB scenarios, with the SMACOF based L2 loss (*MDS SMACOF*) outperforming the direct minimization of the MDS objective (*MDS direct*). We found instead the L2 loss to be unstable: while bringing to an improvement comparable to MDS based objective on the first epochs, it easily tends to diverge later in the training. The best and most stable choice is the one exploiting the closed form solution of SMACOF. For this reason, we used this approach in all the other experiments.

The second ablation study regards the impact of the function $g(ij, lm)$, which outputs the relatedness of the edges connecting keypoints i to j and l to m. Let us denote with $\mathcal{E}$ the set of keypoint pairs (i, j) whose connection constitutes a bone of the shape. We consider the following choices:

- *All*: relations are accounted equally: $g(ij, **) = 1$.
- *Bones*: we consider only the relations of (i, j) actual bones of the chair skeleton and we treat all relations equally, *i.e.* $g(ij, lm) = 1$ if $(l, m) \in \mathcal{E}$.
- *Bones + std.*: we consider again the bones but filtered through the standard deviation on the source

[1] The code is available at https://github.com/LeviVasconcelos/3DKeypoints-DA

Table 3. Quantitative analysis in terms of the Average Error (AE) and Pose-invariant Average Error (PAE), in percentage. Lower is better.

$\mathcal{S}$	$\mathcal{T}$	Baseline		ADDA [45]		Zhou et al. [50]		Our	
		AE	PAE	AE	PAE	AE	PAE	AE	PAE
ModelNet (depth)	Redwood (depth)	16.01	10.73	15.44	10.13	12.76	8.27	**12.27**	**7.68**
	Redwood (RGB)	27.59	13.44	26.16	11.38	25.24	11.38	**23.02**	**10.02**

ground truth ratio as prior information *i.e.* $\sigma^s_{\boldsymbol{R}} = \mathbb{E}_{\boldsymbol{Y}\sim\mathcal{S}}[(\boldsymbol{R}(\boldsymbol{Y}) - \sigma^s_{\boldsymbol{R}})]^{\frac{1}{2}}$. Taking into account the relative uncertainty on the average, we define $g(ij, lm) = \exp(-\sigma^s_{\boldsymbol{R}}(ij, lm)/\mu^s_{\boldsymbol{R}}(ij, lm))$, with $(l, m) \in \mathcal{E}$.

- *Bones + corr.*: similar to the previous variation, but considering the absolute value of the correlation ($C^s_{\boldsymbol{R}}$) among the source ground truth distances: *i.e.* $g(ij, lm) = |C^s_{\boldsymbol{R}}(ij, lm)|$, with $(l, m) \in \mathcal{E}$.

We compare the different models in Table 2. As shown in the table, considering the bones within $g(\cdot,\cdot)$, brings usually a slight gain in performances with respect to not filtering any pair. A further gain can be obtained by weighting the relatedness through the standard deviation, while the correlation does not seem to bring particular benefits. We highlight that what we analyzed here are some possible choices: we believe that our model can benefit from further explorations on how to design $g(\cdot,\cdot)$. In the following, for simplicity, we will use the *Bones* only version of $g(\cdot,\cdot)$.

Comparison with state of the art. In this section we compare our model with state of the art DA methods. We first conduct experiments on rigid objects, using the chairs from ModelNet and Redwood datasets, as we did for the ablation studies. The results are shown in Table 3. We compare our model with [50] and with the general purpose adversarial DA-method ADDA [45], as reported in [50]. It is evident that our model significantly outperforms the baselines in all settings and metrics. Particularly interesting are the results for the challenging setting where Redwood RGB is used as target domain, where the shift is not only from the domain of synthetic to real images, but also from depth to RGB. In this scenario, our model brings a gain of more than 2% on the average error of [50] and more than 4.5% over the source only baselineWe would like to remark that we consider the same source model provided by [50], without using source samples during adaptation and without assuming the presence of multiple views of the same object.

Finally, we compare our model and [50] on non-rigid shapes, using the Human3.6M dataset, in a human pose estimation task. We consider 3 settings: RGB to RGB of [50], RGB to depth and depth to RGB. For this task, we re-train both the source only baseline and the algorithm of [50] on the subsampled dataset, obtaining similar results to those reported in the original paper (137.5 vs 135.6 reported). We highlight that only one view is available for the humans in the depth domain, thus the results of [50] do not make use of the multi-view consistency term.

The results are shown in Table 4. Our model outperforms [50] in all the settings and in both metrics, using less information (*i.e.* single view, no source samples) and with a lower computational cost - practically, our method took 5 hours to perform adaptation from depth to RGB in the Human3.6m dataset, while the author's implementation of [50] took more than 1 day under the same hardware settings. Particularly remarkable are the results for the RGB to depth setting, where multiple views are available: without the multi-view constraint, the performance of [50] have a huge drop and are worse than those of the baseline, while our model consistently improves the source only model.

Qualitative analysis. We report some qualitative results of our method on both Redwood and Human3.6M, comparing its output with the baseline source model without adaptation and [50]. The results are shown in Fig. 2 and 3, both in 2D[2] and 3D. As shown in the Fig.s, our model is accurate even in challenging scenarios where the initial model was not able to provide meaningful predictions (*e.g.* second row of Fig. 2 and 3). By exploiting the prior information about the structure, our consistency loss can provide a better objective for keypoints estimation than the multiple views term in [50]. Particularly significant in this sense are the second row of Fig. 2, where the adaptation of [50] leads to a wrong orientation of the model while ours does not, and the first row of Fig. 3, where the estimated pose of the arms is wrong for [50] but correctly computed by our algorithm.

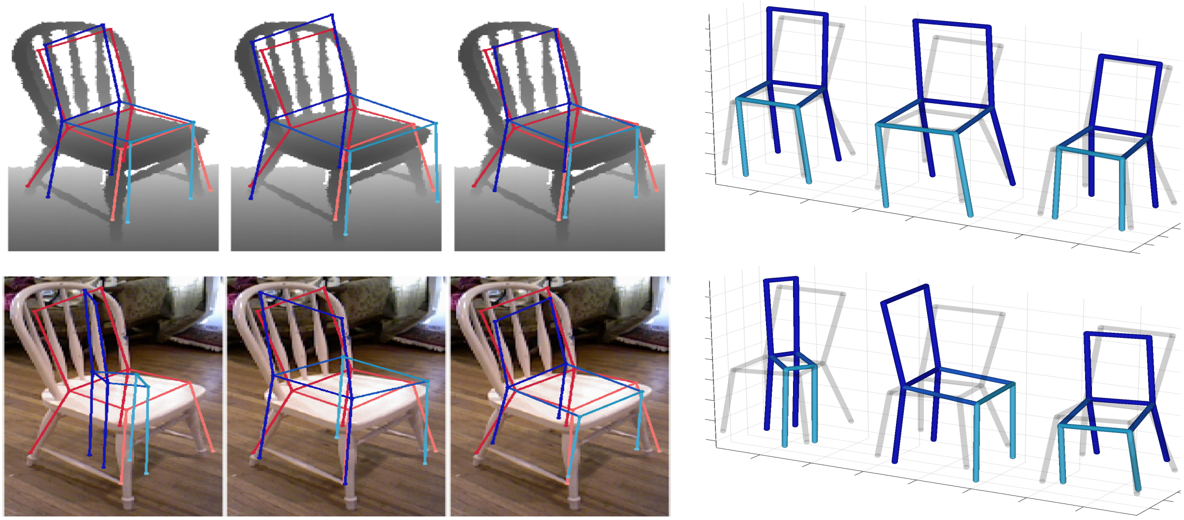
6. Conclusions

We presented an algorithm for performing domain adaptation of a deep model trained for the task of 3D keypoints estimation. Our approach addresses the domain shift by aligning features with BN-based techniques and by operating on the output space and constraining target predictions to be consistent with a structure prior derived from annotated source data. Quantitative and qualitative experiments show the effectiveness of our approach w.r.t. state of the art models. Future works will be devoted to improve our approach by considering multiple views and by exploiting continuous refinement DA strategies [18, 30]. Moreover we plan to extend the idea of our approach to other applications

[2] For the 2D visualization on chairs, we use the code of [50].

Table 4. Quantitative analysis on Human3.6M in terms of the Average Error (AE) and Pose-invariant Average Error (PAE), in mm. Lower is better.

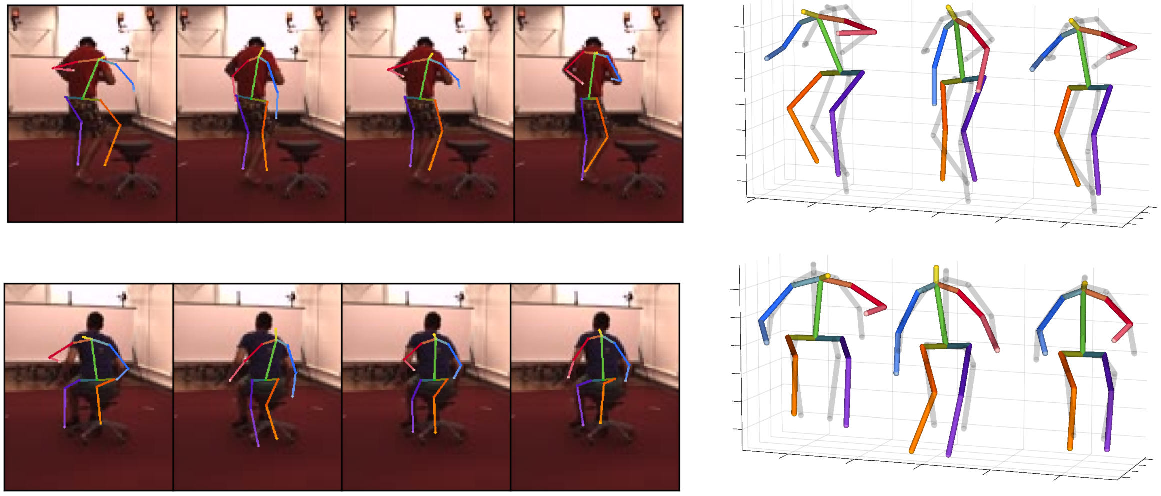
$\mathcal{S}$	$\mathcal{T}$	Baseline		Zhou et al. [50]		Our	
		PA	PAE	AE	PAE	AE	PAE
H3.6M (RGB)	H3.6M (RGB)	180.0	148.5	192.6	159.1	**154.4**	**124.4**
	H3.6M (depth)	304.5	202.1	333.6	234.8	**244.9**	**193.5**
H3.6M (depth)	H3.6M (RGB)	369.6	174.6	369.7	190.8	**340.4**	**160.6**



Baseline Zhou et al. [50] Our

Baseline Zhou et al. [50] Our

Figure 2. Qualitative comparison of our method and the baselines on ModelNet to Redwood of [50], for both depth (first row) and RGB (second row) modalities. The red chairs on the 2D images and the grey chair on the 3D projections denote the ground truth.



Baseline Zhou et al. [50] Our Ground-truth

Baseline Zhou et al. [50] Our

Figure 3. Qualitative comparison of our method and the baselines on Human3.6M [23] in the depth to RGB setting. Different colors denote different bones of the skeleton. The grey skeletons on the 3D projections denote the ground truth.

(*e.g.* semantic segmentation) which requires different priors about the structure of the output space.

Acknowledgements. We acknowledge financial support from ERC grant 637076 - RoboExNovo (L.O.V., M.M., B.C.). This work was carried out under the "Vision and Learning joint Laboratory" between FBK and UniTn.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of Representations for Domain Adaptation. In *Proc. NeurIPS*, 2007. 2

[2] I. Borg and P. Groenen. *Modern Multidimensional Scaling. Theory and Applications*. Springer, 1997. 2

[3] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised Pixel-level Domain Adaptation with GANs. In *Proc. CVPR*, 2017. 1

[4] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain Separation Networks. In *Proc. NeurIPS*, 2016. 3

[5] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulò. AutoDIAL: Automatic Domain Alignment Layers. In *Proc. ICCV*, 2017. 1, 2, 3, 4

[6] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. Rota Bulò. Just DIAL: DomaIn Alignment Layers for Unsupervised Domain Adaptation. In *Proc. ICIAP*, 2017. 2

[7] C.-H. Chen and D. Ramanan. 3D Human Pose Estimation = 2D Pose Estimation + Matching. In *Proc. CVPR*, 2017. 2

[8] Y. Chen, C. Shen, H. Chen, X.-S. Wei, L. Liu, and J. Yang. Adversarial Learning of Structure-Aware Fully Convolutional Networks for Landmark Localization. *IEEE Trans. PAMI*, 2019. 2

[9] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun. No More Discrimination: Cross City Adaptation of Road Scene Segmenters. In *Proc. ICCV*, 2017. 2

[10] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun. A Large Dataset of Object Scans. *arXiv:1602.02481*, 2016. 1, 2, 5

[11] M. Cox and T. Cox. *Multidimensional Scaling*. Chapman and Hall, 1994. 2

[12] J. de Leeuw. Applications of Convex Analysis to Multidimensional Scaling. *Recent Developments in Statistics*, pages 133–145, 1997. 3

[13] G. French, M. Mackiewicz, and M. Fisher. Self-Ensembling for Visual Domain Adaptation. *Proc. ICLR*, 2018. 3

[14] Y. Ganin and V. Lempitsky. Unsupervised Domain Adaptation by Backpropagation. *Proc. ICML*, 2015. 3

[15] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-Adversarial Training of Neural Networks. *Trans. JMLR*, 17(59):1–35, 2016. 1, 2

[16] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Aligning 3D Models to RGB-D Images of Cluttered Scenes. In *Proc. CVPR*, 2015. 1

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. CVPR*, 2016. 6

[18] J. Hoffman, T. Darrell, and K. Saenko. Continuous Manifold Based Adaptation for Evolving Visual Domains. In *Proc. CVPR*, 2014. 7

[19] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In *Proc. ICML*, 2018. 1

[20] J. Hoffman, D. Wang, F. Yu, and T. Darrell. FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation. *arXiv:1612.02649*, 2016. 2

[21] W. Hong, Z. Wang, M. Yang, and J. Yuan. Conditional Generative Adversarial Network for Structured Domain Adaptation. In *Proc. CVPR*, 2018. 1, 2

[22] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. ICML*, 2015. 4

[23] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Trans. PAMI*, 36(7):1325–1339, 2014. 2, 5, 8

[24] I. Kostrikov and J. Gall. Depth Sweep Regression Forests for Estimating 3D Human Pose from Images. In *Proc. BMVC*, 2014. 5

[25] J. Kruskal and M. Wish. *Multidimensional Scaling*. Sage, 1978. 2

[26] Y. Li, A. Dai, L. Guibas, and M. Niessner. Database-Assisted Object Retrieval for Real-Time 3D Reconstruction. *Computer Graphics Forum*, 34(2):435–446, 2015. 1

[27] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu. Adaptive Batch Normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018. 4

[28] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional Adversarial Domain Adaptation. In *Proc. NeurIPS*, 2018. 2

[29] M. Long and J. Wang. Learning Transferable Features with Deep Adaptation Networks. In *Proc. ICML*, 2015. 3

[30] M. Mancini, H. Karaoguz, E. Ricci, P. Jensfelt, and B. Caputo. Kitting in the Wild through Online Domain Adaptation. *IROS*, 2018. 2, 4, 6, 7

[31] M. Mancini, L. Porzi, S. Rota Bulò, B. Caputo, and E. Ricci. Boosting Domain Adaptation by Discovering Latent Domains. In *Proc. CVPR*, 2018. 4

[32] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3D Human Pose Estimation in the Wild Using Improved CNN Supervision. In *Proc. 3DV*, 2017. 2

[33] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. *ACM Trans. Graph.*, 36(4), 2017. 2

[34] P. Morerio, J. Cavazza, and V. Murino. Minimal-Entropy Correlation Alignment for Unsupervised Deep Domain Adaptation. In *Proc. ICLR*, 2018. 3

[35] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic Differentiation in PyTorch. In *Proc. NeurIPS-WS*, 2017. 6

[36] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3D Human Pose from 2D Image Landmarks. In *Proc. ECCV*, 2012. 2

[37] S. Roy, A. Siarohin, E. Sangineto, S. Rota Bulò, N. Sebe, and E. Ricci. Unsupervised Domain Adaptation using Feature-Whitening and Consensus Loss. In *Proc. CVPR*, 2019. 1, 2, 3

[38] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 6

[39] P. Russo, F. M. Carlucci, T. Tommasi, and B. Caputo. From Source to Target and Back: Symmetric Bi-Directional Adaptive GAN. In *Proc. CVPR*, 2018. 1

[40] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional Human Pose Regression. In *Proc. ICCV*, 2017. 5

[41] S. Suwajanakorn, N. Snavely, J. J. Tompson, and M. Norouzi. Discovery of Latent 3D Keypoints via End-to-End Geometric Reasoning. In *Proc. NeurIPS*, 2018. 2

[42] A. Torralba and A. A. Efros. Unbiased Look at Dataset Bias. In *Proc. CVPR*, 2011. 1

[43] S. Tulsiani and J. Malik. Viewpoints and Keypoints. In *Proc. CVPR*, 2015. 1

[44] H.-Y. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki. Adversarial Inverse Graphics Networks: Learning 2D-to-3D Lifting and Image-to-Image Translation from Unpaired Supervision. In *Proc. ICCV*, 2017. 2

[45] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial Discriminative Domain Adaptation. In *Proc. CVPR*, 2017. 1, 2, 7

[46] M. Wang and W. Deng. Deep Visual Domain Adaptation: A Survey. *Neurocomputing*, 312:135–153, 2018. 1, 2

[47] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proc. CVPR*, 2015. 2, 5

[48] Y. Zhang, P. David, and B. Gong. Curriculum Domain Adaptation for Semantic Segmentation of Urban Scenes. In *Proc. ICCV*, 2017. 2

[49] Z. Zhang, S. Lathuilière, A. Pilzer, N. Sebe, E. Ricci, and J. Yang. Online Adaptation through Meta-Learning for Stereo Depth Estimation. *arXiv:1904.08462*, 2019. 2

[50] X. Zhou, A. Karpur, C. Gan, L. Luo, and Q. Huang. Unsupervised Domain Adaptation for 3D Keypoint Estimation via View Consistency. In *Proc. ECCV*, 2018. 1, 2, 5, 6, 7, 8

[51] C. Zimmermann and T. Brox. Learning to Estimate 3D Hand Pose from Single RGB Images. In *Proc. ICCV*, 2017. 1