

Unsupervised Domain Adaptation with Temporal-Consistent Self-Training for 3D Hand-Object Joint Reconstruction

Mengshi Qi, *Member, IEEE*, Edoardo Remelli, Mathieu Salzmann, and Pascal Fua, *Fellow, IEEE*

Abstract—Deep learning-solutions for hand-object 3D pose and shape estimation are now very effective when an annotated dataset is available to train them to handle the scenarios and lighting conditions they will encounter at test time. Unfortunately, this is not always the case, and one often has to resort to training them on synthetic data, which does not guarantee that they will work well in real situations. In this paper, we introduce an effective approach to addressing this challenge by exploiting 3D geometric constraints within a cycle generative adversarial network (CycleGAN) to perform domain adaptation. Furthermore, in contrast to most existing works, which fail to leverage the rich temporal information available in unlabeled real videos as a source of supervision, we propose to enforce short- and long-term temporal consistency to fine-tune the domain-adapted model in a self-supervised fashion.

We will demonstrate that our approach outperforms state-of-the-art 3D hand-object joint reconstruction methods on three widely-used benchmarks and will make our code publicly available.

Index Terms—unsupervised domain adaption, temporal consistency, self-training, cyclegan, 3D hand-object reconstruction

I. INTRODUCTION

HAND-object 3D joint reconstruction is one of many computer vision problems to which convolutional neural networks have brought increasingly effective solutions [1]–[5], including algorithms that can model both the hand and the object it is grasping [6], [7]. However, these methods remain difficult to deploy in practice due to the lack of large annotated datasets with ground-truth 3D hand-object pose and shape that cover a wide enough range of scenarios and lighting conditions. A common solution is therefore to train on synthetic data. Unfortunately, as shown in Fig. 1, a network trained in this fashion can easily fail when dealing with real-world images whose statistics are different from the synthetic ones.

This is known as the *domain shift* problem and the whole field of Domain Adaption is devoted to mitigating it, ideally without requiring any additional annotations in the target domain, which is referred to as Unsupervised Domain Adaptation (UDA) [8], [9]. In its usual formulation [8], [10]–[18], one starts with a large number of annotated *source images* and another set of *target images* whose statistics are those

M. Qi, E. Remelli, M. Salzmann, and P. Fua are with Computer Vision Laboratory, École polytechnique fédérale de Lausanne, Lausanne CH-1015, Switzerland. (Corresponding author: Mengshi Qi, E-mail: {firstname.lastname}@epfl.ch).

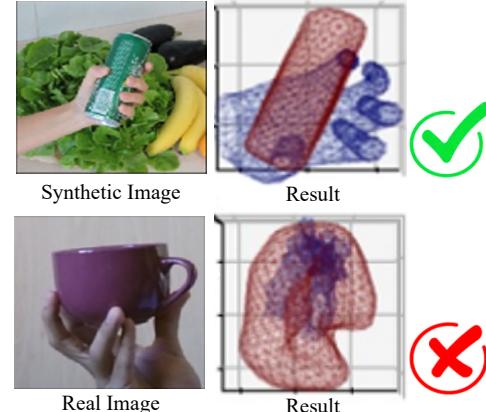


Fig. 1. **From synthetic to real data.** **Top.** It is common practice to train networks on synthetic data, in this case to estimate hand-object pose and shape. **Bottom.** Even though the network of [6] performs well on the synthetic data, domain shift can cause it to fail on real data.

of the images we intend to use in practice but for which we have no annotations. In our case, the source images are the synthetic ones, and the target ones are real images acquired in specific situations in which we want our software to operate. Furthermore, the complex background, various occlusion, and rapidly changing illumination in realistic environments always make it arduous to generalize the parameters of model to the different domain data. Therefore, it is necessary to make use of unlabeled target domain data to help fine-tune the model.

A promising direction to address UDA is to use a Cycle-consistent Generative Adversarial Network (CycleGAN) [19] to turn the annotated source domain images into images that are statistically indistinguishable from the target ones and can be used to train the network. This was demonstrated for semantic segmentation [20]. In this work, we extend this approach for 3D hand-object reconstruction as depicted by Fig. 2. At training time, we use a CycleGAN to translate labeled source images into target-style images. We minimize an adversarial loss to ensure the statistical similarity of the translated images and the real target images while jointly training a 3D hand-object reconstruction network to predict the correct 3D poses and shapes. We further increase performance by introducing unlabeled target video data to add an element of self-supervision by enforcing short- and long-term consistency between the predictions.

We will demonstrate that our approach outperforms state-of-the-art 3D joint hand-object reconstruction methods on

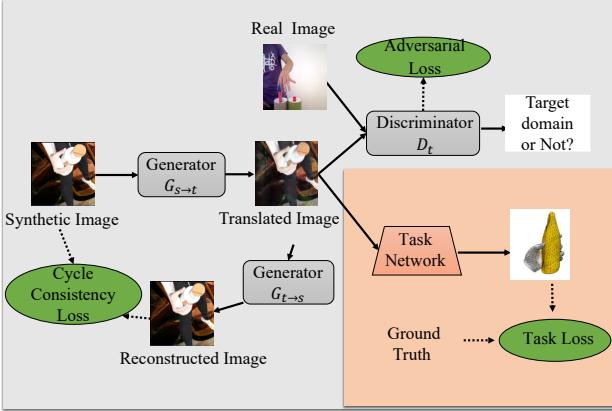


Fig. 2. **Approach.** In this work, we take synthetic images to be the source domain and real one to be the target one. In our approach, we use the CycleGAN component $G_{s \rightarrow t}$ to translate source domain images into ones that cannot be discriminated from target domain images and we use the translated images to retrain the task network.

three widely-used benchmarks and that our approach to UDA for hand-object reconstruction is more effective than other standard ones.

Our main contributions are summarized as follows:

- We propose a 3D geometric constraints based CycleGAN model to bridge the domain gap between synthetic and realistic image domain.
- We design a short-term and a long-term temporal consistency loss for self-supervised fine-tuning to make full use of unlabeled video-based real data.
- Extensive experiments on three public benchmarks demonstrate that our proposed approach outperforms the state-of-the-art UDA methods and baselines.

In the remainder of the paper, we first review related works on unsupervised domain adaptation and 3D hand-object reconstruction. We then present the version of our method that relies only the CycleGAN followed by the more sophisticated version that also uses the unannotated data for self-supervision. Finally, we specialize it for 3D hand-object reconstruction and present our results.

II. RELATED WORK

In this section we briefly review existing deep learning-based approaches to jointly modeling hands and the objects they can hold in 3D. We then turn to existing domain adaptation techniques and unsupervised approaches that could be used to train networks on synthetic data, refine them using unannotated real-images, and finally infer reliable poses from real images.

Hand and Object Joint 3D Reconstruction. The last ten years have seen an explosion in the number of papers that use deep networks and conventional methods to model the interaction between hands and objects, which have been applied in scene understanding [21]–[23] and video analysis [24]–[26]. Many are devoted to reconstruction from RGB-D or multi-view RGB [27]–[31] input. They can be roughly classified into those that use generative methods [29]–[38] and those that rely on discriminative ones [39]–[42].

There has been somewhat less interest for deep learning-based hand/object reconstruction from single RGB images. Unified frameworks for 3D hand-object poses estimation, interaction activity recognition, and synthetic datasets related to these tasks were introduced in [2], [6], [7], [43]–[47]. The resulting methods perform well on the kind of synthetic data for which they have been trained but not so well on real-images, as shown in the example of Fig. 1 because of the domain shift. Our work differs from an approach such as that of [2] in that we directly use hand-object task loss to supervise the CycleGAN and train it jointly with the hand-modeling network.

Domain Adaptation. In recent years, the dominant approach to tackling the domain shift issue has been to learn a domain-invariant representation [9]. One way is to map source data and target data into a latent space by minimizing the Maximum Mean Discrepancy [48] or matching second and higher order statistics [49]–[51] between the source and target domains [10], [12]–[16], [52], [53]. Class labels [52] and anchor points [54], [55] can also be used. Another way consists of leveraging adversarial training to learn source and target features that are indistinguishable [8], [20], [56]–[65]. Beyond image recognition, this has been applied to semantic segmentation [20], [57], [66] and active learning [67], and has been the focus of our own recent work, in which the source and target data pass through networks with the same architecture but different weights [17], [18].

An interesting alternative to these two dominant approaches consists of translating the images from one domain to the other using a CycleGAN [19], thus making it possible to adapt the images directly at the pixel level. To this end, Cycada [20] uses a CycleGAN to translate source images into target-like images that can be used along with the annotations to train a network that performs the desired task to operate in the target domain. Specifically, the CycleGAN and the task network can be trained jointly, which has inspired our own approach. Here, however, we translate this idea to the geometric task of reconstructing 3D hand and object from a single image.

Self-Supervised Learning. Another approach to reducing the required amount of annotated data is to rely on self- or semi-supervised learning [68]–[73]. To provide a useful self-supervisory signal, current methods rely on image color [68]–[70], relative position of image patches [71], [72], random image rotations [73], [74], missing part of image [75], multi-view consistency and depth information [76], depth hints from left and right cameras [77], [78], temporal consistency [70], [79]–[81], multi-domain invariant representation [82], and image clustering [83]. In this work, we show that temporal consistency effectively complements domain adaptation and makes it possible to leverage unannotated video sequences for hand and object 3D reconstruction.

III. APPROACH

In this work, we propose a new UDA framework that exploits a generative adversarial network (CycleGAN) to translate source synthetic images into realistic target-like images for training purposes. We use the translated source images

along with the corresponding annotations to train the target network to perform the task of interest in the target domain. Hence, at inference time, we do not need the CycleGAN anymore and the network can operate on unmodified target images. The key to making this scheme work properly is to impose a task loss, *i.e.*, 3D geometric preservation constraint while training the CycleGAN model so that the translated images retain enough key information for the target network to work well when trained using them. Furthermore, we propose to self-supervised fine-tune the task network by adding a ConvLSTM [84] layer using unlabeled target video data by enforcing long-term and short-term temporal consistency of its predictions.

In this section, we first formalize our UDA problem. We then describe our proposed CycleGAN-based model, and our self-supervised training strategy with the temporal consistency losses.

A. Formalization

We use the encoder-decoder architecture of [6], which we will refer to as the task network F . It takes images as input and returns 3D models for the hand and the object it interacts with. These models include coordinate vectors for vertices of the surface meshes representing the hand and the object surface along with hand pose and shape parameters. We will refer to them collectively as *hand-object vectors*, which will be described in more detail in Section III-C.

Let us consider a source domain $\mathcal{D}_s = \{\mathbf{x}_s, \mathbf{y}_s\}_{i=1}^{N_s}$ with N_s images and corresponding ground-truth hand-object vectors, and a target domain $\mathcal{D}_t = \{\mathbf{x}_t\}_{i=1}^{N_t}$ in which we only have N_t images. We can use \mathcal{D}_s to train a source network F_s and our goal is to learn new network weights for the target network F_t using only \mathcal{D}_s and \mathcal{D}_t so that the network operates effectively in the target domain.

To perform domain adaptation, we introduce a CycleGAN that comprises two generator networks $G_{s \rightarrow t}$ and $G_{t \rightarrow s}$ that translate source domain images into target domain ones and target domain images into source ones, respectively, along with a discriminator network D_t that attempts to discriminate translated source images from true target images. Fig. 2 depicts the role of these networks.

B. Joint Training the CycleGAN and Task Networks

We iteratively train the generator network $G_{s \rightarrow t}$ and task network F_t so as to maximize the performance of the latter on target images. Let $F_t^{(0)} = F_s$ be the task network pre-trained on the source data and let $F_t^{(k-1)}$ be the task network after iteration $k-1$. At iteration k , we feed a source image \mathbf{x}_s to the generator network $G_{s \rightarrow t}$, which outputs a translated source image $G_{s \rightarrow t}(\mathbf{x}_s)$. In turn, it serves as input to the task network $F_t^{(k-1)}$ to add a task-related constraint during training CycleGAN. To update the weights of F_t , D_t , and $G_{s \rightarrow t}(\mathbf{x}_s)$, we minimize a loss that comprises the three loss functions depicted by green ellipses in Fig. 2. They are defined as follows. The first two are standard while the third one is specific to our approach.

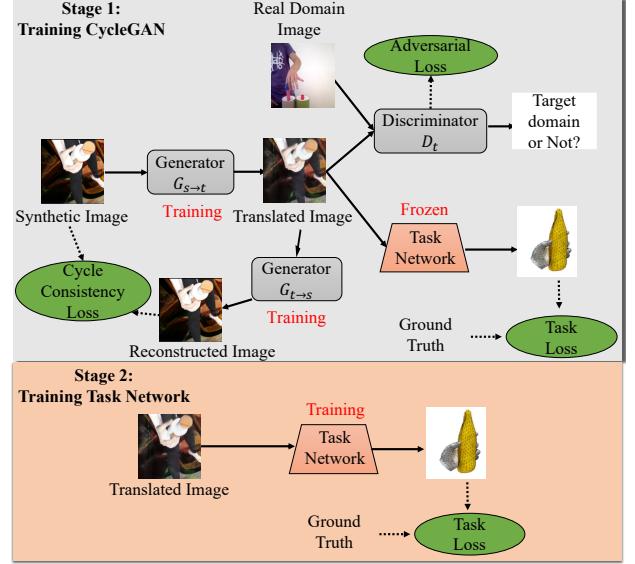


Fig. 3. **Two-Stage Training.** We alternatively train the generators with the task network frozen and then the task network with the generators frozen.

Adversarial Loss: \mathcal{L}_{adv} . It favors translated source images that are statistically indistinguishable from target ones. We take it to be

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_{\mathbf{x}_t \in \mathcal{D}_t} [\log D_t(\mathbf{x}_t)] + \mathbb{E}_{\mathbf{x}_s \in \mathcal{D}_s} [\log (1 - D_t(G_{s \rightarrow t}(\mathbf{x}_s)))] . \quad (1)$$

Cycle-consistency Loss: \mathcal{L}_{cyc} . It ensures that the translated images can be translated back into the original ones. We write it as

$$\mathcal{L}_{\text{cyc}} = \mathbb{E}_{\mathbf{x}_s \in \mathcal{D}_s} [\|G_{t \rightarrow s}(G_{s \rightarrow t}(\mathbf{x}_s)) - \mathbf{x}_s\|_1] . \quad (2)$$

Task Loss: $\mathcal{L}_{\text{task}}$. It is the usual supervised loss used to train the task network given samples and corresponding labels. It is application specific and we describe it Section III-C.

Optimization Strategy: We start with task network $F_t^{(0)}$ and generator networks $G_{s \rightarrow t}$ and $G_{t \rightarrow s}$ trained in the usual manner. In theory, we could then simultaneously train them further by minimizing the joint objective function

$$\mathcal{L}_{\text{UDA}} = \mathcal{L}_{\text{adv}} + \lambda_{\text{cyc}} \mathcal{L}_{\text{cyc}} + \lambda_{\text{task}} \mathcal{L}_{\text{task}} \quad (3)$$

with respect to their weights, where λ_{cyc} and λ_{task} are scalar coefficients. In practice, however, the full architecture depicted by Fig. 2 is too heavy for this. Instead, we alternatively optimize the weights of $F^{(k-1)}$ into those of $F^{(k)}$ with those of the generators fixed and then the weights of the generators with those of the task network frozen. Fig. 3 depicts this process. We have found empirically that the process converges better when we give more weight to the task loss than the cycle loss. We therefore set $\lambda_{\text{cyc}} = 0.1$ and $\lambda_{\text{task}} = 1$.

C. Implementation

We take the task network to be the encoder-decoder architecture of [6] depicted by Fig. 4 and use its publicly available implementation. A ResNet-18 [85] pre-trained on ImageNet [86] serves as the image encoder. The *hand decoder* predicts the shape and pose parameters, θ_H and β_H , of MANO [87], which are used to compute the vertex coordinates

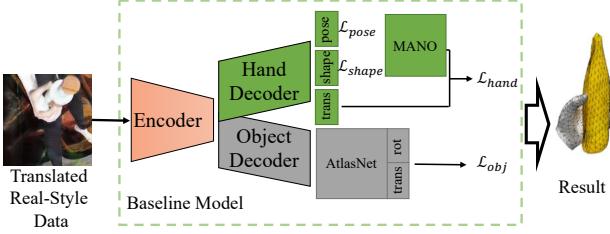


Fig. 4. **Task network.** We use the encoder-decoder architecture of [6]. It takes as input an image and returns 3D mesh models for both the hand and the object.

v_{hand} of the mesh describing the hand. The *object decoder* outputs 3D pose parameters that are combined with the output of an AtlasNet [88] network to yield the vertex coordinates of a 3D mesh describing the object v_{obj} . Principal Component Analysis (PCA) is used to keep the dimensions of θ_H to 15 and that of β_H to 10. Hence, the task loss function $\mathcal{L}_{\text{task}}$ introduced in Section III-B can be written as

$$\begin{aligned} \mathcal{L}_{\text{task}} &= \mathcal{L}_{\text{obj}} + \lambda_h \mathcal{L}_{\text{hand}} + \lambda_s \mathcal{L}_{\text{shape}} + \lambda_p \mathcal{L}_{\text{pose}} + \lambda_c \mathcal{L}_{\text{cont}}, \\ \mathcal{L}_{\text{obj}} &= \sum \| \hat{v}_{\text{obj}} - v_{\text{obj}}^{gt} \|_2^2, \\ \mathcal{L}_{\text{hand}} &= \sum \| \hat{v}_{\text{hand}} - v_{\text{hand}}^{gt} \|_2^2, \\ \mathcal{L}_{\text{shape}} &= \| \beta_H \|_2^2, \\ \mathcal{L}_{\text{pose}} &= \| \theta_H \|_2^2, \\ \mathcal{L}_{\text{cont}} &= \mathcal{L}_{\text{repulsion}} + \mathcal{L}_{\text{attraction}}, \end{aligned} \quad (4)$$

where v_{hand}^{gt} , v_{obj}^{gt} and $\hat{v}_{\text{hand}}^{gt}$, $\hat{v}_{\text{obj}}^{gt}$ are ground-truth and estimated values, respectively. $\mathcal{L}_{\text{repulsion}}$ and $\mathcal{L}_{\text{attraction}}$ are additional loss terms designed to guarantee that the hand does not penetrate into the object but actually touches it when appropriate. For a more complete description we refer the interested reader to the original publication [6]. The model is implemented within the PyTorch framework and all experiments were run on one NVIDIA Tesla V100 GPU. We use the Adam optimizer [89] to train our model for 200 epochs with an initial learning rate of 5×10^{-5} and a batch size of 16.

D. Adding Temporal Consistency

The method of Section III-B treats all unannotated images as individual images as opposed to elements of a video sequence in which motions are continuous. This fails to exploit the fact that, even though the 3D poses in consecutive images are initially unknown, the network predictions should be consistent across consecutive images and without brutal jumps. In this section, we use this fact to define a new loss term that enforces temporal consistency both over the short and the long term.

Formally, let $\mathcal{D}_t = \{\mathbf{x}_t^T\}_{T=1}^{N_t}$ be a sequence of N_t consecutive unlabeled target domain images from a realistic video. For each one, the task network F_t outputs a description of the hand and object at time T in the form of an output vector v^T that encodes the vertex coordinates for the hand and object meshes, along with hand pose and shape vectors θ_H^T and β_H^T , as described in Section III-A.

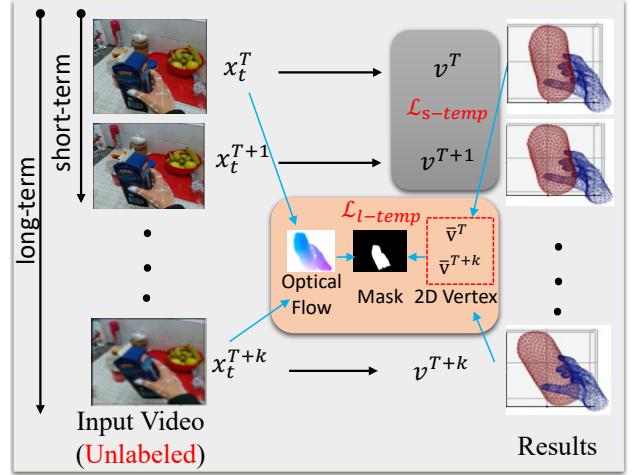


Fig. 5. **Temporal consistency loss.** Given an unannotated video sequence, we compute a smoothness-based short-term consistency loss in neighboring frames x_t^T and x_t^{T+1} and a long-term one between frames x_t^T and x_t^{T+k} with $k > 1$. The latter depends on the consistency between the predicted projected displacement between mesh vertices and the optical flow between the two images.

Short-Term Temporal Consistency. We require the predictions at consecutive time instants to be similar by introducing the short-term loss

$$\begin{aligned} \mathcal{L}_{\text{s-temp}} &= \sum_T (\mathcal{L}_{\text{mano}} + \lambda_{\text{s-temp}} \| v^{T+1} - v^T \|_2^2); \\ \mathcal{L}_{\text{mano}} &= \| \beta_H^{T+1} - \beta_H^T \|_2^2 + \lambda_{\text{s-temp}} \| \theta_H^{T+1} - \theta_H^T \|_2^2, \end{aligned} \quad (5)$$

where the summation occurs over a video sequence. As the hand shape parameters remain constant over such a sequence whereas the pose parameter change, we impose a stronger constraint on the former and set $\lambda_{\text{s-temp}} = 0.01$. Note that we penalize the change in vertex coordinates before applying the roto-translation that maps object-centered vertex coordinates into camera-centered ones with AtlasNet.

Long-Term Temporal Consistency. Because the images at time T and $T + 1$ are different but still similar, $\mathcal{L}_{\text{s-temp}}$ provides a supervisory signal but one that is not as informative than the one that could be obtained from images that are further in time. To this end and as in [7], we use an independent estimate [90] of the optical flow \mathcal{O}_T^{T+k} between images acquired as time T and $T + k$ to enforce temporal consistency on the vertices predicted at time T and $T + k$, i.e., v^T and v^{T+k} , respectively. Then the displacement between the 2D projections of these vertices, \bar{v}^T and \bar{v}^{T+k} , should be consistent with the corresponding optical flow values. We therefore take the long-term temporal consistency loss to be

$$\begin{aligned} \mathcal{L}_{\text{l-temp}} &= \sum_n \mathcal{L}_{\text{l-temp}}^{T,nk} \\ \mathcal{L}_{\text{l-temp}}^{T,nk} &= \frac{1}{N_v} \sum_{i=1}^{N_v} M^T(\bar{v}_i^T) \| (\mathcal{O}_T^{T+k}(\bar{v}_i^T) - (\bar{v}_i^{T+k} - \bar{v}_i^T)) \|_2^2, \end{aligned} \quad (6)$$

where N_v is the total number of vertices and M^T is a binary visibility mask at time T . It is computed by performing an optical flow forward-backward consistency check as in [7], [91]. In practice, we take $T = 1$ and $k = 5$.

Optimization Strategy: We first train the task network F_t on individual images as described in Section III-B. We then fine-tune the network in a self-supervised fashion by minimizing

$$\mathcal{L}_{\text{self}} = \mathcal{L}_{\text{s-temp}} + \lambda_l \mathcal{L}_{1-\text{temp}}, \quad (7)$$

where $\lambda_l = 0.1$. During this minimization, the generator networks are frozen and only the task network weights are refined. In fact, we have found it advantageous to also freeze the first layers of the task network and to only refine the final layer, that is, a ConvLSTM [84] layer with the hidden size of 512 added between the encoder and the last fully connected layer.

IV. EXPERIMENTS

In this section, we first introduce the baselines against which we compare several variants of our approach. We then present the experimental results and analysis in terms of 3D hand-object joint reconstruction.

A. Variants and Baselines

To compare against other approaches to UDA, we use the following baselines:

- **No DA:** Running the task network [6] trained on synthetic data and test on real data without any domain adaptation.
- **UDA w/ CycleGAN [19]:** Using a standard CycleGAN to translate synthetic images and use the translated images to train the task network.
- **UDA w/ FDA [92]:** Using a simple Fourier transform to translate the image.
- **UDA w/ ADAA [61]:** Performing adversarial discriminative domain adaptation on the encoder of the task network to learn domain-invariant representations.
- **UDA w/ DANN [8]:** Adding the gradient reversal layers to the task network to learn domain-invariant representations.

Moreover, we compare them to the following variants of our approach:

- **TASK-CYCLE:** Iterative training of the generators and task network as described in Section III-B.
- **TASK-CYCLE-TEMPO(L+S):** Adding long-term and short-term temporal consistency as described in Section III-D.
- **TASK-CYCLE-TEMPO(S):** Adding only short-term temporal consistency as described in Section III-D.
- **TASK-CYCLE-TEMPO(L):** Adding only long-term temporal consistency as described in Section III-D.
- **TASK-CYCLE-TEMPO(L+S)-LSTM:** Replacing the ConvLSTM layer in the task network by an LSTM layer during self-supervised fine-tuning.
- **TASK-CYCLE-TEMPO(L+S)-Render:** Computing the long-term consistency using an approach similar to [7], that is, first using Neural Render [93] to render the hand and object meshes and then estimating the consistency of the rendered images and the optical flow.
- **TASK-CYCLE-CYCADA:** Instead of using the task loss to compare the output of the network to the ground truth, using it to compare to the images *before* translation, as in [20] and as shown in Fig. 6.

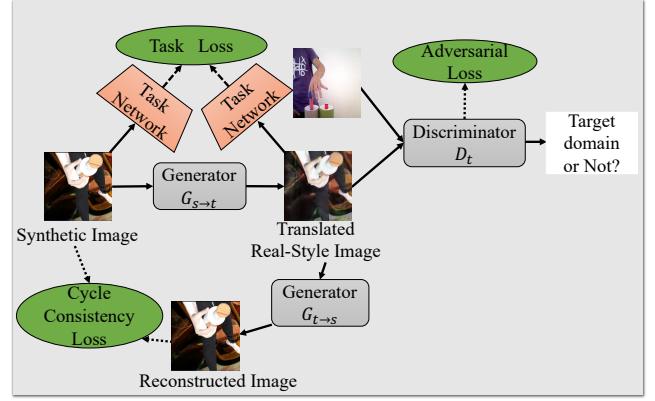


Fig. 6. **Cycada like variant of our approach.** The task loss is now used to compare the output of the network on the input synthetic data and the translated images.

B. Datasets

We evaluate our framework on one large-scale synthetic image dataset *ObMan* [6], and two widely-used real image datasets—*First-person hand benchmark* [94] and *Hands in action dataset* [42]—that we describe in more details below.

ObMan Dataset [6]: ObMan is the largest fully labeled synthetic image dataset describing hands grasping objects. It contains 2.7k objects models of eight categories from the ShapeNet dataset [95] (i.e., bottles, bowls, cans, jars, knifes, cellphones, cameras and remote controls), and grasping actions generated by the GraspIt software [96]. Furthermore, all hand models are fitted with MANO [87] to grasp a provided object mesh, and the hands are rendered by SMPL [87]. The body and object textures were taken from SURREAL [97] and ShapeNet [95]. Each scene was rendered using Blender [98] with a random lighting and in front of background images randomly sampled from LSUN [99] and ImageNet [86]. Following [6], we select 141k frames and 6K frames as training set and test set, respectively.

First-person hand benchmark (FHB) [94]: FHB is an RGB-D video-based dataset of first-person hand-object interactions. The data is fully annotated with hand's and object's 6D pose, 3D location and mesh model. Following the setting in [6], to focus on hand-object interaction, we only select the frames in which the distance between the manipulating hand and the manipulated object is less than 1cm, and choose three categories of objects, i.e., *salt*, *juice carton* and *liquid soap*. This corresponds to a subset of FHB with 8420 training frames and 9103 testing ones.

Hands in action dataset (HIC) [42]: Following the experimental setting in [6], we use four video sequences from HIC, and employ two sequences (251 frames) as training set and two others (307 frames) as test. All frames feature interactions between one hand and a sphere or a cube. In our experiments, we choose the frames in which the minimal distance between the hand and the vertices of the manipulated object is below 5mm. Furthermore, the HIC dataset provides hand and object meshes, to which we fit the MANO [87] model for supervision with dense 3D points.

TABLE I
Comparative results on FHB and HIC datasets. WE PROVIDE BOTH MEAN VALUES AND VARIANCES OVER SEVERAL TRIALS. THE BEST RESULTS ARE SHOWN IN BOLD.

Methods	FHB Dataset						HIC Dataset				
	Hand Error	Object Error	Maximum Penetration	Simulation Displacement	Intersection volume	Hand Error	Object Error	Maximum Penetration	Simulation Displacement	Intersection volume	
No DA-only source	32.7 ± 0.7	2079.2 ± 58.7	13.7 ± 0.6	49.2 ± 2.2	19.7 ± 0.3	33.1 ± 0.7	2150.0 ± 66.5	23.7 ± 0.9	62.5 ± 2.0	36.7 ± 0.3	
UDA w/ CycleGAN [19]	26.9 ± 0.6	2057.3 ± 61.5	10.9 ± 0.6	45.1 ± 2.0	16.1 ± 0.3	28.0 ± 0.6	2140.7 ± 63.5	21.5 ± 0.8	59.1 ± 1.6	34.9 ± 0.3	
UDA w/ FDA [92]	27.2 ± 0.7	2062.3 ± 62.2	11.7 ± 0.7	46.6 ± 1.7	16.7 ± 0.3	28.9 ± 0.7	2147.5 ± 65.0	22.9 ± 0.6	61.2 ± 1.9	35.7 ± 0.5	
UDA w/ ADDA [61]	28.1 ± 0.6	2064.5 ± 57.5	11.9 ± 0.6	47.5 ± 1.8	17.3 ± 0.2	29.8 ± 1.1	2149.3 ± 61.2	23.5 ± 0.7	62.2 ± 2.1	36.3 ± 0.4	
UDA w/ DANN [8]	27.8 ± 0.8	2064.0 ± 59.0	11.5 ± 0.5	47.1 ± 1.9	16.9 ± 0.3	29.3 ± 1.0	2149.0 ± 62.5	23.2 ± 0.8	61.7 ± 2.2	35.9 ± 0.5	
TASK-CYCLE	25.2 ± 0.6	2054.0 ± 58.0	10.1 ± 0.5	44.1 ± 1.8	15.5 ± 0.2	26.5 ± 0.6	2137.3 ± 61.0	20.1 ± 0.6	57.7 ± 1.6	33.7 ± 0.2	
TASK-CYCLE-TEMPO(L+S)	24.3 ± 0.5	2052.6 ± 60.5	9.5 ± 0.6	43.1 ± 1.6	15.0 ± 0.2	25.3 ± 0.5	2135.2 ± 62.2	18.8 ± 0.7	56.6 ± 1.7	33.0 ± 0.3	
TASK-CYCLE-CYCADA	25.5 ± 0.5	2054.3 ± 61.9	10.3 ± 0.6	44.3 ± 2.1	15.6 ± 0.3	26.7 ± 0.6	2137.7 ± 63.9	20.4 ± 0.8	58.0 ± 2.0	33.9 ± 0.4	
TASK-CYCLE-TEMPO(S)	24.9 ± 0.2	2053.3 ± 56.3	9.9 ± 0.3	43.9 ± 1.5	15.5 ± 0.1	26.3 ± 0.3	2136.7 ± 60.5	19.8 ± 0.5	57.5 ± 1.5	33.6 ± 0.2	
TASK-CYCLE-TEMPO(L)	24.7 ± 0.5	2053.0 ± 57.6	10.0 ± 0.5	43.7 ± 1.6	15.3 ± 0.3	26.1 ± 0.5	2136.2 ± 63.2	19.6 ± 0.6	57.3 ± 1.6	33.5 ± 0.3	
TASK-CYCLE-TEMPO(L+S)-LSTM	24.5 ± 0.5	2052.7 ± 59.0	9.7 ± 0.5	43.5 ± 1.6	15.1 ± 0.2	25.8 ± 0.6	2135.5 ± 63.0	19.5 ± 0.6	57.0 ± 1.8	33.2 ± 0.3	
TASK-CYCLE-TEMPO(L+S)-Render	24.6 ± 0.6	2052.5 ± 60.5	9.7 ± 0.6	43.4 ± 2.0	15.2 ± 0.3	25.7 ± 0.6	2135.3 ± 64.7	19.3 ± 0.7	56.9 ± 1.8	33.2 ± 0.5	

C. Metrics

For comparison purposes, we use the same metrics as in [4], [6], [42]:

Hand Reconstruction Error. We compute the mean end-point error (mm) across 21 joints as in [4].

Object Reconstruction Error. We measure the symmetric Chamfer distance (mm) between sampled vertices on the ground-truth mesh and points on the predicted mesh as in [6], [88]. The mean per vertex error is therefore the number we report over 642, the total number of vertices.

Contact Quality. We compute the *penetration depth* and *intersection volume* between reconstructed hands and objects. To this end, we use a voxel size of 0.5cm to measure the intersection volume, and the penetration depth is the maximum distance from the sampled points on the hand mesh to the surface of the object mesh.

Simulation displacement. We run the simulation environment introduced in [42] that returns a measure of the physical plausibility of our estimates. We refer the interested reader to its full description in the original paper.

D. Results

Fig. 8 depicts four results on individual frames of **FHB**. In the first three, the recovered hand-pose is very realistic and much better than the baseline **No DA**. The fourth one showcases a failure. It is attributable to the absence of any such grasping action in the synthetic training set and illustrates the fact that our approach can correct for changing imaging conditions but not for missing poses. Fig. 9 features results over a sequence. Even though each frame is fed to the network individually, the result is very consistent over time. Fig. 11 depicts a few of the translated synthetic images created while training the network. Fig. 10 shows the results on **HIC** dataset.

We report our quantitative results on **FHB** and **HIC** in Table I. Both **TASK-CYCLE**, our joint-training approach of Section III-B, and **TASK-CYCLE-TEMPO(L+S)**, our full approach with the temporal consistency terms turned on, consistently outperform the baselines, with **TASK-CYCLE-TEMPO(L+S)** doing better than **TASK-CYCLE**. To obtain these results, we performed six iterations of the iterative algorithm of Section III-B for each dataset. As can be seen in Fig. 7, the performance tends to plateau after three iterations.

TABLE II
Success vs Failure ON THE FHB DATASET. (**Top rows**) HAND RECONSTRUCTION ERROR FOR THE 50 WORST AND BEST IMAGES, WITHOUT DA AND USING **TASK-CYCLE**. (**Bottom row**) DISTANCE TO THE CLOSEST POSE IN THE TRAINING SET. UNSURPRISINGLY, THIS NUMBER CORRELATES CLOSELY WITH THE CHANCES OF SUCCESS OR FAILURE.

Methods	50 Worst Images	50 Best Images
No DA hand error	59.5	17.2
TASK-CYCLE hand error	56.2	9.5
Dist. to closest training sample	12.5	3.1

To pinpoint the reasons for success or failure, we selected from the FHB test set the 50 images that gave the worst results in terms of hand reconstruction error and the 50 images that gave the best. As can be seen in Table II, in the 50 best cases, our approach divides the error by almost a factor 2, whereas in the 50 worst cases the improvement is much smaller in percentage terms. This closely correlates with the fact that the distance between the observed pose and its closest training sample is an excellent predictor of success or failure. In other words, our approach can compensate for changes of imaging conditions but not for the training set being relatively small and failing to cover the whole range of hand motions that can be seen in the real world.

E. Ablation Study

The bottom rows of Table I document the performance of the variants of our approach introduced in Section IV-A and show that the various losses we introduced in Sections III-B and III-D all contribute to the final result. In particular, **TASK-CYCLE** does better than **TASK-CYCLE-CYCADA**, which highlights the importance of imposing the task loss at the right place. Also, **TASK-CYCLE-TEMPO(L+S)** does slightly better than **TASK-CYCLE-TEMPO(L+S)-Render**, even though we use a much simpler formulation of the long-term temporal loss function. We attribute this to the fact our version is closer to being convex and, unlike the NeuralRenderer, does not introduce any artifacts.

V. CONCLUSION

In this paper, we have presented an unsupervised domain adaptation strategy for 3D hand-object joint reconstruction.

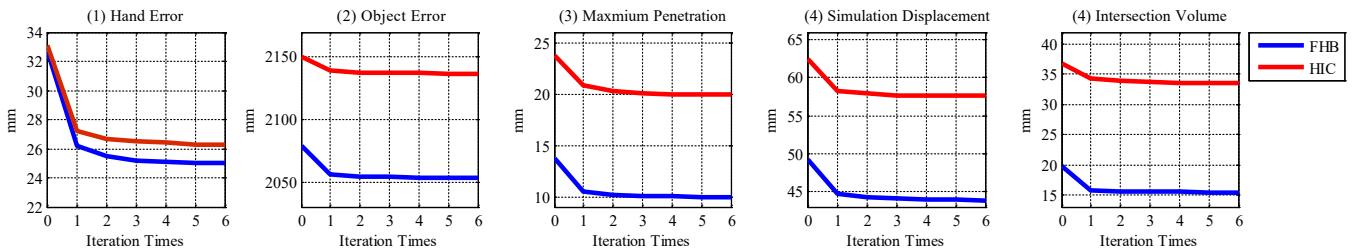


Fig. 7. **Convergence of our joint training scheme.** Accuracy as a function of the number of iterations for the FHB and HIC datasets.

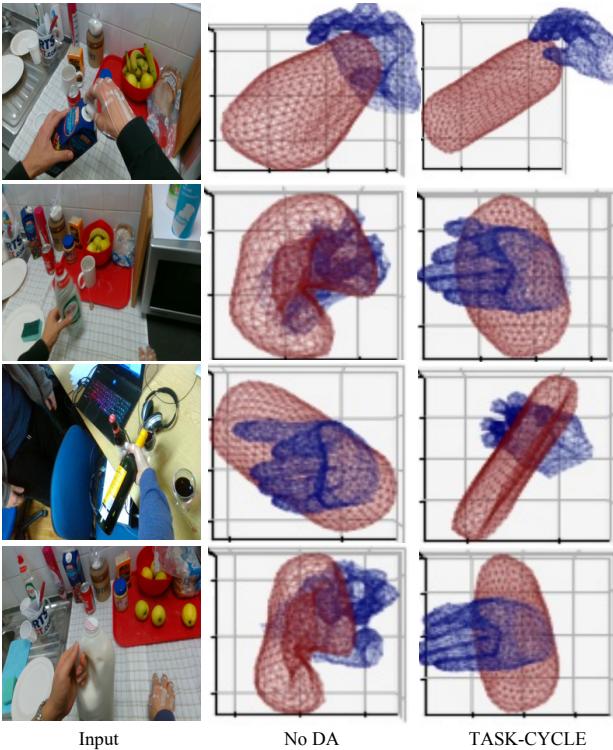


Fig. 8. **Qualitative results on single images** from the FHB dataset. The hand pose recovered by our system is correct in the first three rows but wrong in the fourth, which can be attributed to the fact that no such hand pose exists in the training set.

It involves introducing 3D geometric constraints and self-supervised temporal consistency in a CycleGAN-based framework. Our geometric constraints allow our approach to effectively transfer the annotations from the synthetic source data to the unlabeled, real target domain. Furthermore, our short-term and long-term temporal consistency loss functions let us leverage unlabeled video data to fine-tune the task model. Our experiments on three widely-used datasets have demonstrated the effectiveness of our method and its superiority over the state of the art. In future work, we will explore the use of physics-based constraints to model the contact between hand and object and to supply further supervisory signals.

ACKNOWLEDGMENT

This work was supported by the Microsoft JRC Project.

REFERENCES

- [1] U. Iqbal, P. Molchanov, T. B. J. Gall, and J. Kautz, "Hand Pose Estimation via Latent 2.5 D Heatmap Regression," in *European Conference on Computer Vision*, 2018, pp. 118–134.
- [2] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "Generated Hands for Real-Time 3D Hand Tracking from Monocular RGB," in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 49–59.
- [3] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand Keypoint Detection in Single Images Using Multiview Bootstrapping," in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1145–1153.
- [4] C. Zimmermann and T. Brox, "Learning to Estimate 3D Hand Pose from Single RGB Images," in *International Conference on Computer Vision*, 2017, pp. 4903–4911.
- [5] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-Modal Deep Variational Hand Pose Estimation," in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 89–98.
- [6] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. Black, I. Laptev, and C. Schmid, "Learning Joint Reconstruction of Hands and Manipulated Objects," in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11807–11816.
- [7] Y. Hasson, B. Tekin, F. Bogo, I. Laptev, M. Pollefeys, and C. Schmid, "Leveraging photometric consistency over time for sparsely supervised hand-object reconstruction," in *Conference on Computer Vision and Pattern Recognition*, 2020.
- [8] Y. Ganin and V. Lempitsky, "Unsupervised Domain Adaptation by Backpropagation," in *International Conference on Machine Learning*, 2015, pp. 1180–1189.
- [9] G. Csurka, "A Comprehensive Survey on Domain Adaptation for Visual Applications," in *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 1–35.
- [10] M. Long, H. Zhu, J. Wang, and M. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 136–144.
- [11] K. K. Saito, S. S. Yamamoto, Y. Y. Ushiku, and T. T. Harada, "Open Set Domain Adaptation by Backpropagation," in *European Conference on Computer Vision*, 2018, pp. 153–168.
- [12] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep Domain Confusion: Maximizing for Domain Invariance," in *arXiv Preprint*, 2014.
- [13] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning Transferable Features with Deep Adaptation Networks," in *International Conference on Machine Learning*, 2015, pp. 97–105.
- [14] M. Long, J. Wang, and M. Jordan, "Deep Transfer Learning with Joint Adaptation Networks," in *International Conference on Machine Learning*, 2017, pp. 2208–2217.
- [15] W. Zellinger, T. Grubinger, E. Lugofer, T. Natschläger, and S. Saminger-Platz, "Central Moment Discrepancy (cmd) for Domain-Invariant Representation Learning," in *International Conference on Learning Representations*, 2016.
- [16] P. Koniusz, Y. Tas, and F. Porikli, "Domain Adaptation by Mixture of Alignments of Second- or Higher-Order Scatter Tensors," in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4478–4487.
- [17] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond Sharing Weights for Deep Domain Adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 4, pp. 801–814, 2019.
- [18] R. Bermúdez-Chacón, M. Salzmann, and P. Fua, "Domain Adaptive Multibranch Networks," in *International Conference on Learning Representations*, 2020.

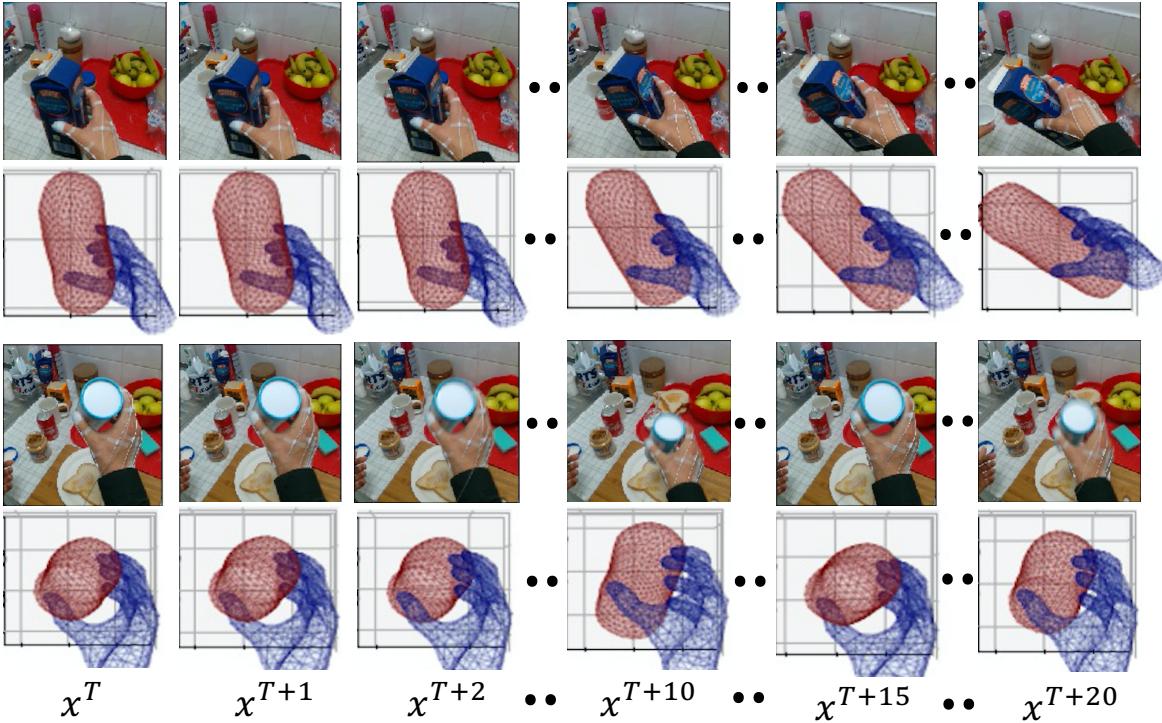


Fig. 9. **Qualitative results on two sequences** from the FHB dataset. The temporal loss was used to train the network but, at inference time, each image is processed individually.

- [19] J.-Y. Zhu, T. Park, P. Isola, and A. Efros, “Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks,” in *International Conference on Computer Vision*, 2017, pp. 2223–2232.
- [20] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, “CyCADA: Cycle Consistent Adversarial Domain Adaptation,” in *International Conference on Machine Learning*, 2018, pp. 1989–1998.
- [21] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo, “Attentive relational networks for mapping images to scene graphs,” in *Conference on Computer Vision and Pattern Recognition*, 2019.
- [22] M. Qi, Y. Wang, J. Qin, and A. Li, “KE-GAN: Knowledge embedded generative adversarial networks for semi-supervised scene parsing,” in *Conference on Computer Vision and Pattern Recognition*, 2019.
- [23] M. Qi, Y. Wang, A. Li, and J. Luo, “STC-GAN: Spatio-Temporally Coupled Generative Adversarial Networks for Predictive Scene Parsing,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5420–5430, 2020.
- [24] M. Qi, J. Qin, A. Li, Y. Wang, J. Luo, and L. V. Gool, “stagnet: An attentive semantic RNN for group activity recognition,” in *European Conference on Computer Vision*, 2018.
- [25] M. Qi, J. Qin, X. Zhen, D. Huang, Y. Yang, and J. Luo, “Few-Shot Ensemble Learning for Video Classification with SlowFast Memory Networks,” in *ACM International Conference on Multimedia*, 2020.
- [26] M. Qi, J. Qin, Y. Wu, and Y. Yang, “Imitative Non-Autoregressive Modeling for Trajectory Forecasting and Imputation,” in *Conference on Computer Vision and Pattern Recognition*, 2020.
- [27] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys, “Motion Capture of Hands in Action Using Discriminative Salient Points,” in *European Conference on Computer Vision*, 2012, pp. 640–653.
- [28] I. Oikonomidis, N. Kyriazis, and A. Argyros, “Full Dof Tracking of a Hand Interacting with an Object by Modeling Occlusions and Physical Constraints,” in *International Conference on Computer Vision*, 2011, pp. 2088–2095.
- [29] C. Wan, T. Probst, L. V. Gool, and A. Yao, “Self-Supervised 3D Hand Pose Estimation Through Training by Fitting,” in *Conference on Computer Vision and Pattern Recognition*, 2019.
- [30] C. Wan, T. Probst, L. V. Gool, and A. Yao, “Dense 3D Regression for Hand Pose Estimation,” in *Conference on Computer Vision and Pattern Recognition*, 2018.
- [31] C. Wan, T. Probst, L. V. Gool, and A. Yao, “Dual Grid Net: Hand Mesh Vertex Regression from Single Depth Maps,” in *European Conference on Computer Vision*, 2020.
- [32] H. Hamer, J. Gall, T. Weise, and L. V. Gool, “An Object-Dependent Hand Pose Prior from Sparse Training Data,” in *Conference on Computer Vision and Pattern Recognition*, 2010, pp. 671–678.
- [33] H. Hamer, K. Schindler, E. Koller-Meier, and L. V. Gool, “Tracking a Hand Manipulating an Object,” in *International Conference on Computer Vision*, 2009, pp. 1475–1482.
- [34] I. Oikonomidis, N. Kyriazis, and A. Argyros, “Tracking the Articulated Motion of Two Strongly Interacting Hands,” in *Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1862–1869.
- [35] T. Pham, N. Kyriazis, A. Argyros, and A. Kheddar, “Hand-Object Contact Force Estimation from Markerless Visual Tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2883–2896, 2017.
- [36] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt, “Real-Time Joint Tracking of a Hand Manipulating an Object from RGB-D Input,” in *European Conference on Computer Vision*, 2016, pp. 294–310.
- [37] A. Tsoli and A. Argyros, “Joint 3D Tracking of a Deformable Object in Interaction with a Hand,” in *European Conference on Computer Vision*, 2018, pp. 484–500.
- [38] D. Tzionas and J. Gall, “3D Object Reconstruction from Hand-Object Interactions,” in *International Conference on Computer Vision*, 2015, pp. 729–737.
- [39] G. Rogez, J. Supancic, and D. Ramanan, “Understanding Everyday Hands in Action from RGB-D Images,” in *International Conference on Computer Vision*, 2015, pp. 3889–3897.
- [40] J. Romero, H. Kjellström, and D. Kragic, “Hands in Action: Real-Time 3D Reconstruction of Hands in Interaction with Objects,” in *International Conference on Robotics and Automation*, 2010, pp. 458–463.
- [41] G. Rogez, J. Supancic, and D. Ramanan, “First-Person Pose Recognition Using Egocentric Workspaces,” in *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4325–4333.
- [42] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall, “Capturing Hands in Action Using Discriminative Salient Points and Physics Simulation,” *International Journal of Computer Vision*, vol. 118, no. 2, pp. 172–193, 2016.
- [43] B. Tekin, F. Bogo, and M. Pollefeys, “H+o: Unified Egocentric

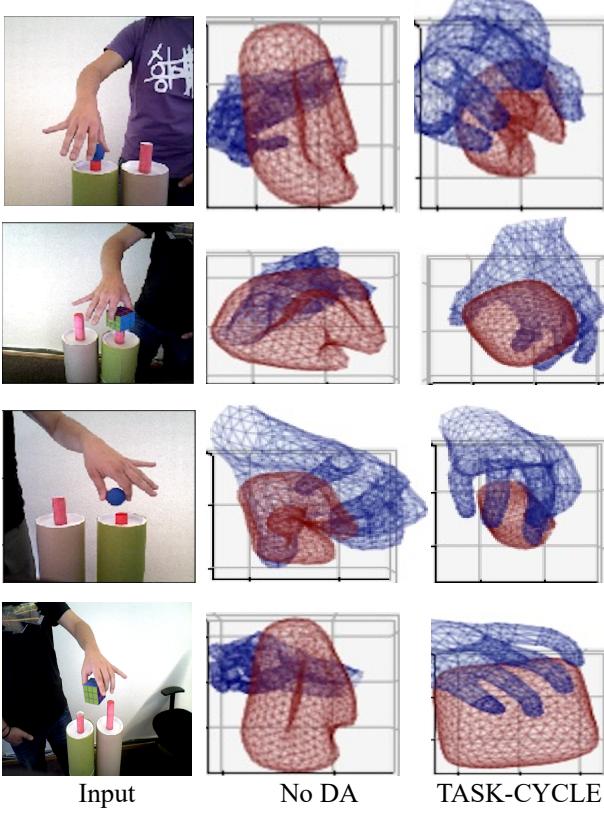


Fig. 10. Qualitative results on single images from the HIC dataset.

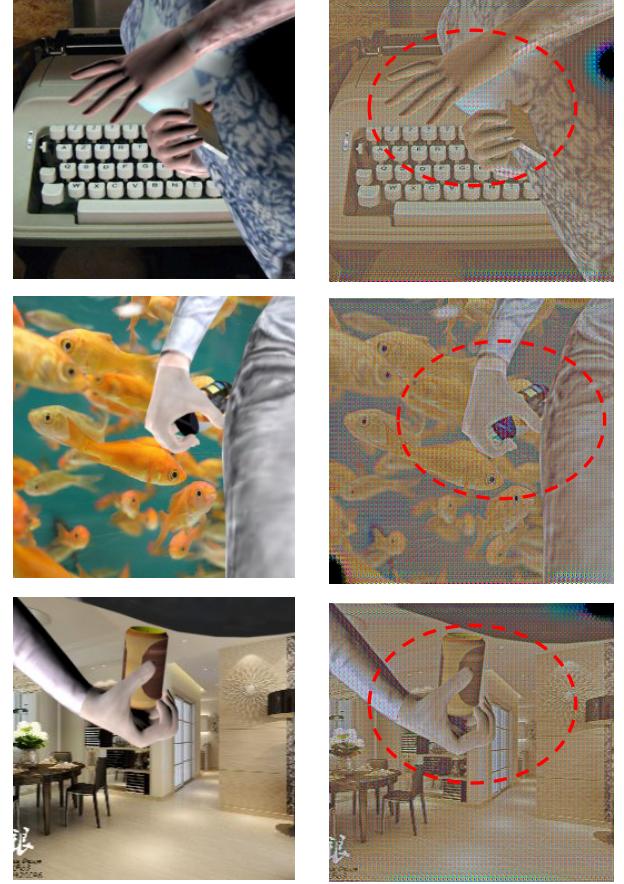
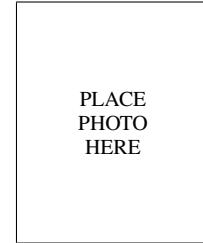


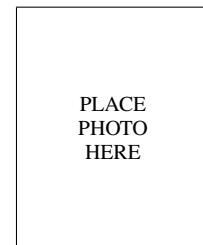
Fig. 11. Original and translated ObMan images. As observed in [100], the translated images do not need to appear particularly realistic to the human eye to be useful for training purposes, as long as their statistics are appropriate for this purpose.

- Recognition of 3D Hand-Object Poses and Interactions,” in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4511–4520.
- [44] A. Armagan, G. Garcia-Hernando, S. Back, and et al., “Measuring Generalisation to Unseen Viewpoints, Articulations, Shapes and Objects for 3D Hand Pose Estimation under Hand-Object Interaction,” in *European Conference on Computer Vision*, 2020.
- [45] S. Brahmbhatt, C. Tang, C. Twigg, C. Kemp, and J. Hays, “Contact-Pose: A Dataset of Grasps with Object Contact and Hand Pose,” in *European Conference on Computer Vision*, 2020.
- [46] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, “Honnote: A Method for 3D Annotation of Hand and Object Poses,” in *Conference on Computer Vision and Pattern Recognition*, 2020.
- [47] B. Doosti, S. Naha, M. Mirbagheri, and D. Crandall, “HOPE-Net: A Graph-based Model for Hand-Object Pose Estimation,” in *Conference on Computer Vision and Pattern Recognition*, 2020.
- [48] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, “A Kernel Method for the Two-Sample Problem,” in *Advances in Neural Information Processing Systems*, 2007, pp. 513–520.
- [49] P. Koniusz, F. Yan, P.-H. Gosselin, and A. K. Mikolajczyk, “Higher-Order Occurrence Pooling for Bags-Of-Words: Visual Concept Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 313–326, 2017.
- [50] B. Sun, J. Feng, and K. Saenko, “Correlation Alignment for Unsupervised Domain Adaptation,” in *Domain Adaptation in Computer Vision Applications*, 2017, pp. 153–171.
- [51] B. Sun and K. Saenko, “Deep CORAL: Correlation Alignment for Deep Domain Adaptation,” in *European Conference on Computer Vision*, 2016, pp. 443–450.
- [52] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, “Maximum Classifier Discrepancy for Unsupervised Domain Adaptation,” in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3723–3732.
- [53] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, “Mind the Class Weight Bias: Weighted Maximum Mean Discrepancy for Unsupervised Domain Adaptation,” in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2272–2281.
- [54] P. Häusser, T. Frerix, A. Mordvintsev, and D. Cremers, “Associative Domain Adaptation,” in *International Conference on Computer Vision*, 2017, pp. 2784–2792.
- [55] S. Shkodrani, M. Hofmann, and E. Gavves, “Dynamic Adaptation on Non-Stationary Visual Domains,” in *European Conference on Computer Vision*, 2018.
- [56] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, “Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks,” in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3722–3731.
- [57] W. Hong, Z. Wang, M. Yang, and J. Yuan, “Conditional Generative Adversarial Network for Structured Domain Adaptation,” in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1335–1344.
- [58] T. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, “Advent: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation,” in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2517–2526.
- [59] Y. Li, L. Yuan, and N. Vasconcelos, “Bidirectional Learning for Domain Adaptation of Semantic Segmentation,” in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6936–6945.
- [60] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, “Simultaneous Deep Transfer Across Domains and Tasks,” in *International Conference on Computer Vision*, 2015, pp. 4068–4076.
- [61] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial Discriminative Domain Adaptation,” in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7167–7176.
- [62] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.
- [63] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, “Domain-Adversarial Training of Neural Networks,” *Journal of Machine Learning Research*, vol. 17, pp. 591–5935, 2016.
- [64] L. Hu, M. Kan, S. Shan, and X. Chen, “Duplex Generative Adversarial

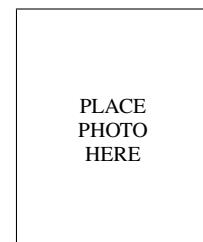
- Network for Unsupervised Domain Adaptation,” in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1498–1507.
- [65] R. Zhang, P. Isola, A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *Conference on Computer Vision and Pattern Recognition*, 2018.
- [66] Y. Chen, W. Li, and L. Van Gool, “ROAD: Reality Oriented Adaptation for Semantic Segmentation of Urban Scenes,” in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7892–7901.
- [67] J. Su, Y. Tsai, K. Sohn, B. Liu, S. Maji, and M. Chandraker, “Active Adversarial Domain Adaptation,” in *Conference on Computer Vision and Pattern Recognition*, 2019.
- [68] R. R. Zhang, P. P. Isola, and A. A. A. Efros, “Colorful Image Colorization,” in *European Conference on Computer Vision*, 2016, pp. 649–666.
- [69] G. Larsson, M. Maire, and G. Shakhnarovich, “Learning Representations for Automatic Colorization,” in *European Conference on Computer Vision*, 2016, pp. 577–593.
- [70] C. C. Vondrick, A. A. Shrivastava, A. A. Fathi, S. S. Guadarrama, and K. K. Murphy, “Tracking Emerges by Colorizing Videos,” in *European Conference on Computer Vision*, 2018, pp. 391–408.
- [71] M. Noroozi and P. Favaro, “Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles,” in *European Conference on Computer Vision*, 2016, pp. 69–84.
- [72] C. Doersch, A. Gupta, and A. Efros, “Unsupervised Visual Representation Learning by Context Prediction,” in *International Conference on Computer Vision*, 2015, pp. 1422–1430.
- [73] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised Representation Learning by Predicting Image Rotations,” in *arXiv Preprint*, 2018.
- [74] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4I: Self-Supervised Semi-Supervised Learning,” in *International Conference on Computer Vision*, 2019, pp. 1476–1485.
- [75] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, “Context Encoders: Feature Learning by Inpainting,” in *Conference on Computer Vision and Pattern Recognition*, 2016.
- [76] C. Godard, O. M. Aodha, and G. Brostow, “Unsupervised Monocular Depth Estimation with Left-Right Consistency,” in *Conference on Computer Vision and Pattern Recognition*, 2017, p. 7.
- [77] V. Sterzentsenko, L. Saroglou, A. Chatzitofis, S. Thermos, N. Zioulis, A. Doumanoglou, D. Zarpalas, and P. Daras, “Self-Supervised Deep Depth Denoising,” in *International Conference on Computer Vision*, 2019, pp. 1242–1251.
- [78] J. Watson, M. Firman, G. Brostow, and D. Turmukhambetov, “Self-Supervised Monocular Depth Hints,” in *International Conference on Computer Vision*, 2019, pp. 2162–2171.
- [79] H. Lee, J. Huang, M. Singh, and M. Yang, “Unsupervised Representation Learning by Sorting Sequences,” in *International Conference on Computer Vision*, 2017, pp. 667–676.
- [80] I. Misra, C. Zitnick, and M. Hebert, “Shuffle and learn: unsupervised learning using temporal order verification,” in *European Conference on Computer Vision*, 2016, pp. 527–544.
- [81] X. Zhou, A. Karpur, C. Gan, L. Luo, and Q. Huang, “Unsupervised Domain Adaptation for 3D Keypoint Prediction from a Single Depth Scan,” in *arXiv Preprint*, 2017.
- [82] Z. Feng, C. Xu, and D. Tao, “Self-Supervised Representation Learning from Multi-Domain Data,” in *International Conference on Computer Vision*, 2019, pp. 3245–3255.
- [83] M. Larsson, E. Stenborg, C. Toft, L. Hammarstrand, T. Sattler, and F. Kahl, “Fine-Grained Segmentation Networks: Self-Supervised Segmentation for Improved Long-Term Visual Localization,” in *International Conference on Computer Vision*, 2019, pp. 31–41.
- [84] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.
- [85] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [86] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A Large-Scale Hierarchical Image Database,” in *Conference on Computer Vision and Pattern Recognition*, 2009.
- [87] J. Romero, D. Tzionas, and M. Black, “Embodyed Hands: Modeling and Capturing Hands and Bodies Together,” *ACM Transactions on Graphics*, vol. 36, no. 6, p. 245, 2017.
- [88] T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry, “Atlasnet: A Papier-Mâché Approach to Learning 3D Surface Generation,” in *Conference on Computer Vision and Pattern Recognition*, 2018.
- [89] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimisation,” in *International Conference on Learning Representations*, 2015.
- [90] Y. Hu, R. Song, and Y. Li, “Efficient Coarse-To-Fine Patch Match for Large Displacement Optical Flow,” in *Conference on Computer Vision and Pattern Recognition*, 2016.
- [91] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu, “Occlusion aware unsupervised learning of optical flow,” in *Conference on Computer Vision and Pattern Recognition*, 2018.
- [92] Y. Yang and S. Soatto, “Fda: Fourier domain adaptation for semantic segmentation,” in *Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4085–4095.
- [93] H. Kato, Y. Ushiku, and T. Harada, “Neural 3D Mesh Renderer,” in *Conference on Computer Vision and Pattern Recognition*, 2018.
- [94] G. Garcia-Hernando, S. Yuan, S. Baek, and T. Kim, “First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations,” in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 409–419.
- [95] A. Chang, T. Funkhouser, L. G., P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “Shapenet: An Information-Rich 3D Model Repository,” in *arXiv Preprint*, 2015.
- [96] A. Miller and P. Allen, “Graspit! a Versatile Simulator for Robotic Grasping,” *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
- [97] G. Varol, J. Romero, X. Martin, N. Mahmood, M. Black, I. Laptev, and C. Schmid, “Learning from Synthetic Humans,” in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 109–117.
- [98] B. O. Community, “Blender-A 3D Modelling and Rendering Package,” 2017.
- [99] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, “Lsun: Construction of a Large-Scale Image Dataset Using Deep Learning with Humans in the Loop,” in *arXiv Preprint*, 2015.
- [100] A. Rozantsev, V. Lepetit, and P. Fua, “On Rendering Synthetic Images for Training an Object Detector,” *Computer Vision and Image Understanding*, vol. 137, pp. 24–37, 2015.



Mengshi Qi (S’16-M’19) received the B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2012, and M.S. and Ph.D. degrees in computer science from Beihang University, Beijing, China, in 2014 and 2019, respectively. He is currently a post-doc researcher in the CVLAB at EPFL. His research interests include machine learning and computer vision, especially scene understanding, 3D reconstruction and multi-media analysis.



Edoardo Remelli Currently he is a Ph.D. student and pursuing his Ph.D. degree in Computer Vision Laboratory, EPFL. His research interests lie in the field of computer vision and machine learning with applications in 3D reconstruction.



Mathieu Salamann is a Senior Researcher at EPFL. Previously, he was a Senior Researcher and Research Leader in NICTAs computer vision research group, a Research Assistant Professor at TTI-Chicago, and a postdoctoral fellow at ICSI and EECS at UC Berkeley. He obtained his PhD in Jan. 2009 from EPFL. His research interests lie at the intersection of machine learning and geometry for computer vision.

PLACE
PHOTO
HERE

Pascal Fua is a Professor of Computer Science at EPFL, Switzerland. His research interests include shape and motion reconstruction from images, analysis of microscopy images, and Augmented Reality. He is an IEEE Fellow and has been an Associate Editor of the IEEE journal Transactions for Pattern Analysis and Machine Intelligence.