Partially-Shared Variational Auto-encoders for Unsupervised Domain Adaptation with Target Shift – Supplementary Material –

Ryuhei Takahashi[1], Atsushi Hashimoto[2], Motoharu Sonogashira[1], and Masaaki Iiyama[1]

[1] Kyoto University, Japan {sonogashira,iiyama}@mm.media.kyoto-u.ac.jp
[2] OMRON SINIC X Corp., Tokyo, Japan atsushi.hashimoto@sinicx.com

A Details of the experimental settings

A.1 Various levels of target shifts with digit datasets

Table A lists the numbers of samples for each dataset at each level of the target-shift, which was used in the experiment in 4.2 in the main paper. Because USPS had only small numbers of training samples (500 to 1000 for each category), the samples from the dataset were over-sampled to adjust the levels. For the other datasets, we randomly discarded the samples. Note that SVHN had no imbalanced situation because it was used only as a source domain but not as a target domain, but to balance the total number against the MNIST dataset, the samples were also randomly discarded.

Table A. Number of samples under each condition. SVHN was only used as a source domain and had no imbalanced setting.

	10%	20%	30%	40%	50%
USPS (1)	500	1125	1922	3000	4500
USPS (other)	500	500	500	500	500
MNIST (1)	4000	4500	5400	6000	6300
MNIST (other)	4000	2000	1400	1000	700
SVHN (1)	4000	-	-	-	-
SVHN (other)	4000	-	-	-	-

A.2 Hyper-parameters

As defined in Eq. (6) in the main paper, the proposed method has five hyper-parameters of $\alpha, \beta, \gamma, \delta, \epsilon$. In fact, the method has many hyper-parameters but they can be hand-tuned without accessing the labels of target domain samples.

First, L_{pred} is little competitive to the other losses due to the trainable parameters in M. Hence, we can simply set any value to ϵ. Second, L_{fc} is the main objective of the method and should become small enough to predict the target domain samples. Hence, to prioritize L_{fc} against the other competing losses, we should set a large value for δ. Third, the hyper-parameters other than the above two can be tuned by a visual check of domain conversion quality while maintaining the small L_{fc}. After these steps, we can intuitively tune the all hyper-parameters without accessing labels of the target domain samples. In this sense, the sensitivity against hyper-parameter tuning with this method is less important than general UDA methods.

Based on the above hand-tuning process, in each experiment, we set these hyper-parameters as listed in Table B, which are carefully adjusted by hand without accesing target domain labels.

Table B. hyper-parameters used in the experiments.

	α	β	γ	δ	ϵ
digit classification	3	1	3	10	3
human pose estimation	5	10	3	20	3

A.3 Network structure and the algorithm detail

We show the entire network architecture and the exact training algorithm in Fig. A and Alg. 1, respectively. Note that the important sub-parts are identical with those shown in Figures 2, 3, 4 in the main paper, where all the encoders and decoders are partially sharing the parameters as shown in Figure 5.

Each module was implemented as following. In the human pose estimation task, we used a common network architecture for all the baseline methods and the proposed method (including ablations). It was designed for this specific task but its CycleGAN part is basically based on the ComboGAN, which was also used by CyCADA, with four ResNet modules. The detail of the network architecture is shown in Tables C to G, where ResBlock is defined in Table H. The number of channels for z and ζ_* are set to 64 and 96, respectively. Note that this setting is adopted in the digit classification task, too.

In the digit classification task, we used the same base architecture with CyCADA for the proposed method (other than the shared weights). For the other methods, we basically used the architecture from their original papers but some methods are not tested with the digit datasets and borrowed their architecture from the other methods, which is summarized in Table I.

A.4 Data augmentation in the SVHN $\rightarrow$ MNIST task

In the SVHN $\rightarrow$ MNIST task, some conventional methods (UFDN, CyCADA) augmented training images in the MNIST dataset by inverting all pixel values

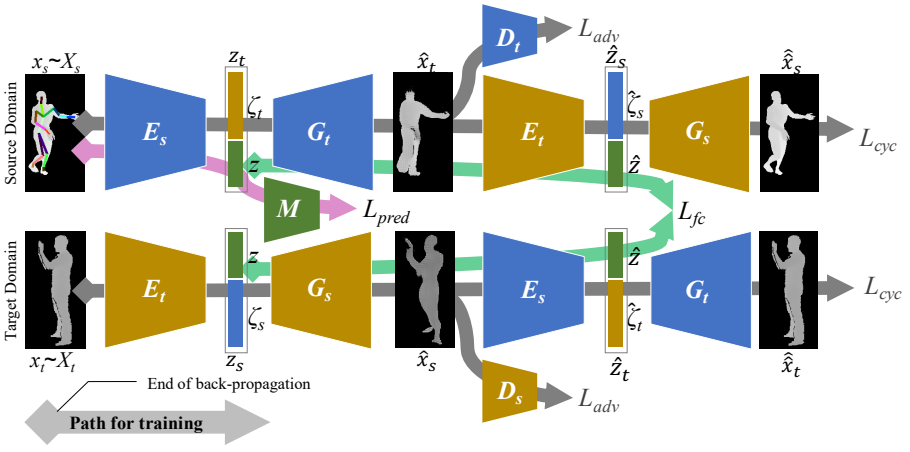

Fig. A. The entire network architecture with some loss calculations.

Algorithm 1 Training of PS-VAEs.

Require: $\{X_s, Y_s\}$ and X_t, which are the source and target domain datasets.
Ensure: E_*, G_*, D_* ($* \in \{s, t\}$), which are encoders, decoders, and discriminators for the domains, and the predictor M.

repeat
  freeze the parameters of Es, Gs, defreeze Ds.
  sample batches x_s, x_t from X_s, X_t.
  update Ds to maximize L_{adv} (Fig. 2).
  freeze Ds and defreeze the others.
  sample batches x_s, y_s from X_s, Y_s and x_t from X_t.
  calculate L_{cyc} and L_{adv} for x_s and x_t (Fig. A, the gray paths) while L_pred for x_s and y_s (Fig. 4 or Fig. A, the pink path).
  calculate L_{id} and L_{KL} for x_s and x_t (Fig. 5).
  extract z_s and z_t, detach them from derivation graph, then calculate L_{fc} (Fig. 3 or Fig. A, the green paths).
  update the model to minimize Eq. (6) in the main paper.
until reaching to a decided number of iterations,

by chance. The reported results of these methods and the proposed method are obtained with this augmentation.

A.5 Background subtraction for the target domain data in human pose estimation

The depth images in the CMU panoptic dataset contains cluttered background regions, which are diverse and essentially not related to human pose. These background regions are detected automatically in a pre-process. Since depth

values in a background are always bigger than the foreground in nature and videos in the dataset are captured with fixed cameras, background images can be generated by obtaining the maximum depth value at each pixel in a video. We obtained background images of each target domain samples in this way and applied background subtraction in advance. After that, we selected the maximum foreground connected component as the human region. We set a uniform value to the detected background pixels. Finally, the detected foreground regions (i.e., human regions) were cropped with the aspect ratio of 2 : 1 and resized to 256 channels × 128 pixels.

Note that the uniform value of the background region is equal to that of CG images. In addition, the same cropping and resizing process were performed to CG images.

Table C. Input x_* for the human pose estimation task.

	height	width	channel(s)
MNIST↔USPS	32	32	1
SVHN→MNIST	32	32	3
pose estimation	256	128	1

Table D. Encoder E_* for the human pose estimation task.

Input	$h \times w \times c$
Conv	$7 \times 7 \times 64$, refrectionpad 3
Parametrized ReLU	
Conv	$3 \times 3 \times 128$, pad 1, stride 2
Parametrized ReLU	
Conv	$3 \times 3 \times 256$, pad 1, stride 2
Parametrized ReLU	
ResBlock	256 channels
ResBlock	256 channels
ResBlock	256 channels
ResBlock	256 channels
(mu) Conv	$3 \times 3 \times 256$, pad 1
(logvar) Conv	$3 \times 3 \times 256$, pad 1
Output	$h/4 \times w/4 \times 256$

Table E. Decoder G_* for the human pose estimation task.

Input	$h/4 \times w/4 \times 256$
ResBlock	256 channels
ResBlock	256 channels
ResBlock	256 channels
ResBlock	256 channels
Upsampling	$2 \times 2 \times 256$
Conv	$3 \times 3 \times 128$, pad 1
Parametrized ReLU	
Upsampling	$2 \times 2 \times 128$
Conv	$3 \times 3 \times 64$, pad 1
Parametrized ReLU	
Conv	$7 \times 7 \times c$, reflectionpad 3
Output	$h \times w \times c$

Table F. Regresser M for the human pose estimation task.

Regressor M	
Input	8×8×64 channels
ResBlock	64 channels
ResBlock	64 channels
ResBlock	64 channels
ResBlock	64 channels
Upsampling	$2 \times 2 \times 64$
Conv	$3 \times 3 \times 32$, pad 1
Parametrized ReLU	
Upsampling	$2 \times 2 \times 32$
Conv	$3 \times 3 \times 16$, pad 1
Parametrized ReLU	
Conv	$7 \times 7 \times 18$, reflectionpad 3
Output	$h \times w \times 18$

Table G. Discriminator D_* for the human pose estimation task.

Input	$h \times w \times c$
Conv	$4 \times 4 \times 64$, pad 2, stride 2
Spectral norm	
Parametrized ReLU	
Conv	$4 \times 4 \times 128$, pad 2, stride 2
Spectral norm	
Parametrized ReLU	
Conv	$4 \times 4 \times 256$, pad 2, stride 2
Spectral norm	
Parametrized ReLU	
Conv	$4 \times 4 \times 512$, pad 2, stride 2
Spectral norm	
Parametrized ReLU	
Conv	$4 \times 4 \times 512\times$, pad 2
Spectral norm	
Parametrized ReLU	
Conv	$4 \times 4 \times 1$, pad 2
Output	$(h/16 + 4) \times (w/16 + 4)$

Table H. Definition of the ResBlock.

Input	$h \times w \times c_{in}$
Conv	$3 \times 3 \times c_{in}$, pad 1
Parametrized ReLU	
Conv	$3 \times 3 \times c_{in}$, pad 1
Parametrized ReLU	
Output	$h \times w \times c_{out}$

Table I. Networks used in each method for digit classification. "original" indicated that the network is the same with that used in the original paper.

	digit classification
Source only	same as ADDA
ADDA	original
UFDN	original
PADA	same as ADDA
SimGAN	same as CyCADA
CyCADA	original
MCD	original
PS-VAEs	same as CyCADA

B Joint distributions

We show all the distributions of joint positions in Figures B. Note that a right hand, for example, can appear in the both sides of the image by chance because the direction of the human is not fixed.

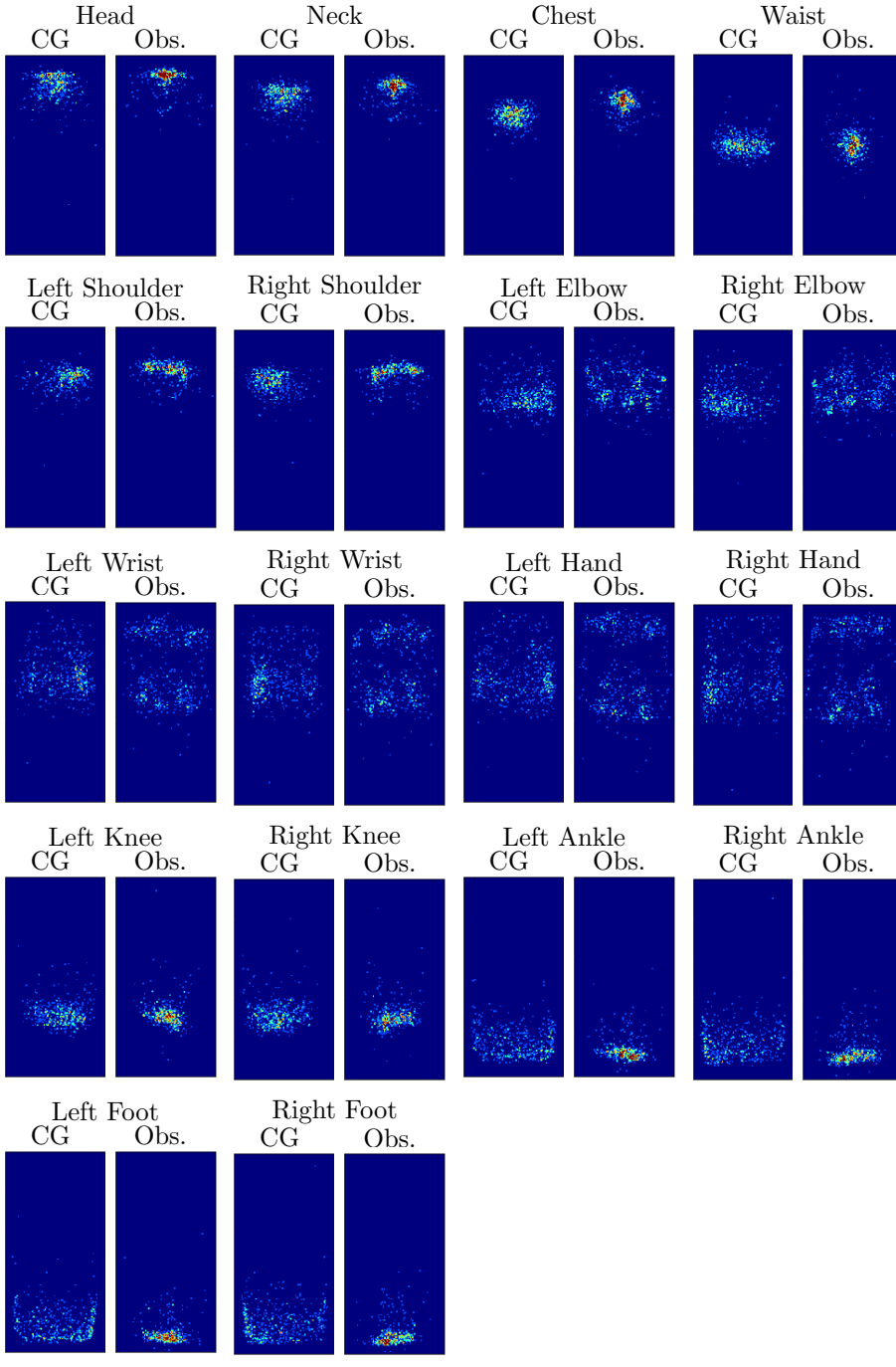

Fig. B. Difference in pose distribution between Observed images and CG images.