Unsupervised Domain Adaptation by Uncertain Feature Alignment Supplementary Material

Tobias Ringwald
tobias.ringwald@kit.edu
Rainer Stiefelhagen
rainer.stiefelhagen@kit.edu

Institute for Anthropomatics and Robotics (CV:HCI Lab)
Karlsruhe Institute of Technology
Karlsruhe, Germany

1 Training Setup and Hyperparameters

Our basic training setup is conducted as follows: For our Office-Home and Office-Caltech experiments, we use ResNet-50 as feature extractor; for VisDA 2017 ResNet-101 is the common setup. In both cases, networks are pretrained on ImageNet. We append one linear layer for classification and jointly optimize all parameters using SGD with Nesterov momentum of 0.95. All of our experiments share the same hyperparameters which are shown in Table 2: We first train on the source domain until convergence with learning rate 5×10^{-4}, batch size 240 and weight decay 5×10^{-5}. For the adaptation step, we use learning rate 2.5×10^{-4}, which is lowered at 66% training progress by factor 0.1.

We then train multiple adaptation cycles which all contain the same amount of cycle steps. For every step, we sample a mini-batch according to our proposed BIS setup, shuffle it according to SBL and then train on it with our proposed UFL loss function. UFL is only applied to target instances; source instances are trained with normal cross-entropy loss with label smoothing. Before a new cycle starts, we recalculate the features, p and $\tilde{p}$ (using the average over 20 MC dropout iterations). $\overline{p}$ (and thus UFM) is resampled every five steps. Parameter k of UBF's top-k calculation is defined as $\frac{|\mathcal{C}|}{4}$. Our code is implemented in PyTorch and trained on 4 NVIDIA GTX 1080 Ti GPUs. Thus, SBL considers 4 replicas when distributing source and target samples.

In all cases, we only use a single ResNet model (for all predictions, feature and uncertainty extraction) which is progressively adapted to the new target domain. We only vary two hyperparameters between datasets: a) The number of source epochs, which is owed to the fact that VisDA 2017 is much bigger than Office-Caltech and Office-Home. b) Adaptation cycles. BIS samples a fixed amount of classes per mini-batch. The number of adaptation cycles is thus 5 times higher for Office-Home as it has roughly 5 times the amount of classes.

2 Feature Visualization

We also provide visualizations for the feature embeddings of our proposed setup in Figure 1 for the VisDA 2017 validation set. Colors indicate the ground truth label of the image associ-

ated with the feature. The 2048-dimensional ResNet-101 embeddings were first trimmed by PCA and then projected into 2D space by t-SNE. Features of the source-only trained model are shown in Figure 1a. Evidently, the model has not been adapted to the target domain which is also reflected in the feature quality: With the exception of the aeroplane class, all features are clumped together in one large blob with no clear separation. After the first cycle, instances with low prediction uncertainty are separated from the remainder. This was the case for *e.g.* the classes *aeroplane*, *horse* and *plant*, as they are visually distinct from other classes. In Table 1, we show the mean $\tilde{p}$ distribution before the first adaptation step given the current maximum prediction of p. This mean distribution also reflects the visualization after the first cycle (see Figure 1b): Classes without multiple distinct peaks in the distribution are already separated (*e.g.* aeroplane and horse) while classes with multiple peaks are kept together (*e.g.* bicycle/motorcycle and knife/skateboard). Note that shown values represent an average over the whole class; for a single instance, $\tilde{p}$ shows even more distinct peaks.

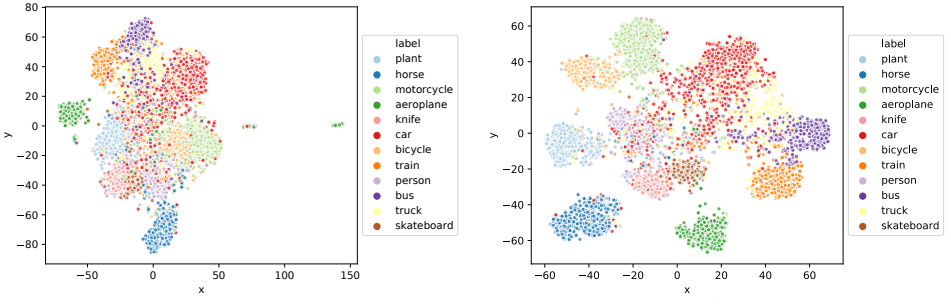
The overall uncertainty decreases as training progresses, leading to more distinct clusters. Eventually, even classes that started with high uncertainty levels (*e.g. person*) are separated from other class clusters. In the final visualization, we can see that our model kept classes with multiple peaks in $\tilde{p}$ – and therefore a high confusion level – together in terms of spatial proximity until later training stages (deferred disentanglement). Easier examples such as *e.g.* aeroplane are already separated from other classes in earlier training stages.

The final feature embeddings in Figure 1d also visualize some of VisDA's problems: For example, images of the *car* and *truck* class often include *persons* in the foreground although only a single label is given for the classification task. This is reflected by the seemingly random features in between the person and car clusters. The same holds true for the *car* class which was often found in the background of images labeled as either *bicycle* or *motorcycle*. This should thus not be regarded as a shortcoming of our proposed method but rather as a lack of annotation quality.

3 Uncertainty Based Filtering Visualization

In Figure 2, we also provide a visualization of our proposed Uncertainty Based Filtering (UBF) approach for the VisDA 2017 validation dataset and for 3 out of 12 transfer tasks from Office-Home. The figure depicts the relative amount of training samples that are filtered with regard to the current training progress.

Evidently, the amount of filtered samples decreases as training progresses and the model becomes more certain of its predictions. All depicted transfer tasks result in a plateau towards the end of the training. For VisDA, very few samples (414 out of 55388) remain filtered at the end of the training process. This is mostly caused by the dataset's small amount of classes (only 12) and their visual distinctiveness (*e.g.* aeroplane vs. person). For the Office-Home transfer tasks, we also notice a plateau towards the end but given the dataset's 65 classes and larger confusion potential (*e.g.* notebook vs. keyboard), the plateau is located at a higher relative percentage. Lastly, we also observe different plateaus between the three transfer tasks within the Office-Home dataset. Easier transfer tasks such as Cl-Rw and Ar-Pr converge towards a lower plateau when compared to the harder Pr-Cl and Rw-Ar tasks. This is also reflected in the accuracy of the mentioned transfer tasks (see main paper), with Ar-Pr being about 17% higher in accuracy than Rw-Ar. Overall, UBF prevents training on highly confusing (or sometimes even mislabeled) data, therefore not only improving the training process itself but also auxiliary tasks such as the proposed UFM computation.



(a) source only (b) 1st cycle

(c) 2nd cycle (d) last cycle

Figure 1: Visualization of VisDA 2017 validation set features after PCA+t-SNE dimension reduction for different phases of the training progress. Best viewed in color.

Figure 2: Relative amount of training samples filtered by UBF with regard to the training progress. Best viewed in color.

Maximum prediction in p	aeroplane	bicycle	bus	car	horse	knife	motorcycle	person	plant	skateboard	train	truck
aeroplane	**35.5**	4.2	5.9	7.4	6.0	4.4	7.1	4.4	6.4	5.9	9.0	3.7
bicycle	6.7	**29.1**	6.2	5.9	5.4	4.7	11.9	5.6	8.1	4.5	7.1	4.8
bus	6.0	4.5	**32.6**	10.5	5.7	3.8	6.6	4.6	6.4	3.6	11.5	4.1
car	5.5	4.6	8.7	**32.8**	5.8	4.6	8.4	5.1	7.2	3.9	6.9	6.7
horse	6.4	4.1	4.2	6.3	**38.4**	3.7	8.0	5.4	7.3	3.8	7.8	4.6
knife	7.5	5.4	5.0	7.7	6.6	**23.9**	6.5	5.5	8.8	11.0	7.9	4.3
motorcycle	5.8	5.5	5.1	9.6	5.8	3.6	**37.0**	5.0	7.9	4.3	6.5	3.9
person	7.8	5.8	5.0	8.8	9.4	4.5	9.8	**23.7**	7.8	5.6	6.3	5.5
plant	7.3	5.9	5.6	8.1	7.3	5.5	7.2	5.7	**27.6**	7.0	8.0	4.8
skateboard	6.8	5.6	4.3	8.1	5.5	8.7	7.9	5.2	10.7	**27.8**	5.9	3.4
train	7.1	4.4	9.2	7.5	5.7	3.6	6.7	4.4	6.7	3.4	**36.8**	4.5
truck	5.5	4.8	8.1	12.9	6.1	4.5	6.3	5.1	7.3	3.5	10.7	**25.2**

Table 1: Mean scores from $\tilde{p}$ given the maximum prediction from p as label (leftmost column). Values were extracted before the first adaptation step. Maximum values are depicted in bold face, second larges values are underlined. Note that these are mean scores over all $\tilde{p}$ distributions for a given class prediction, thus showing less distinct but still representative peaks.

Hyperparameter	Office-Caltech	Office-Home	VisDA 2017
Base LR	5×10^{-4}	5×10^{-4}	5×10^{-4}
Adaptation LR	2.5×10^{-4}	2.5×10^{-4}	2.5×10^{-4}
Weight Decay	5×10^{-5}	5×10^{-5}	5×10^{-5}
Batch Size	240	240	240
Num. GPUs	4	4	4
λ	$\{5,4,3\}$	$\{5,4,3\}$	$\{5,4,3\}$
MC dropout rate	0.85	0.85	0.85
φ	0.50	0.50	0.50
Dropout rate	0.75	0.75	0.75
Cycle steps	50	50	50
Regeneration steps	5	5	5
Source Epochs	50	50	1
Adaptation Cycles	15	75	15

Table 2: Hyperparameters used for training. Note that the only difference is the number of training epochs to accommodate the vastly different dataset sizes and class counts.