

UBR²S: Uncertainty-Based Resampling and Reweighting Strategy for Unsupervised Domain Adaptation – 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 Introduction

Due to space limitations in the main paper, we provide further experimental results in this supplementary material. Section 2 describes the employed hyperparameters, network architectures and training setup. Section 3 provides ablation studies for the ϵ hyperparameter and training stability over multiple runs. Section 4 reports results on the Office-31 [19] and Office-Home [23] datasets in a multi-source UDA setup. Section 5 studies UBR²S’ performance w.r.t. different backbone architectures.

Furthermore, we provide additional experimental results on Office-Home [23] and the VisDA 2017 [15] validation set in Tables 4 and 5. An expanded version of the ablation study in the main paper can be found in Table 6.

2 Implementation Details

Hyperparameters. All of our experiments are based on the same hyperparameter set and follow the setup proposed in [17]: We first optimize f and g on the source domain for 1000 iterations using SGD with batch size 240 and learning rate 5×10^{-4} . For the adaptation phase, the learning rate is 2.5×10^{-4} over 100 cycles with 50 forward passes each cycle and $\beta=12$ (as per [17]). The Monte-Carlo dropout rate is 75% with $|\mathcal{M}|=50$ for all setups. Experiments involving DSS use $\epsilon=0.25$ (see Section 3). Our method is implemented in PyTorch [14] and trained on four NVIDIA 1080 Ti GPUs.

Network architectures. For Office-Caltech, Office-31 and Office-Home, we utilize ResNet-50 [9] for comparison to SOTA. For our VisDA 2017 experiments, we use the default ResNet-101 [9] architecture unless otherwise noted. During the backbone ablation study, we also employ MobileNetV2 [20] and DenseNet-121 [6]. In any case, all networks are pretrained on ImageNet [8], use a two layer classifier g (similar to [18]) and the loss described in the main paper.

Multi-source bins. In multi-source UDA setups with D domains, source bins $\mathcal{B}_c^{S_{1..D}}$ (see

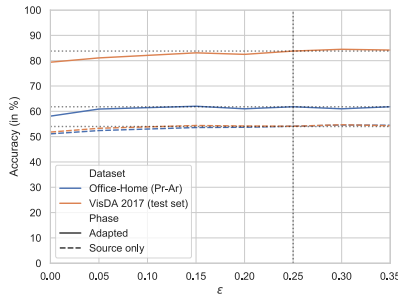


Figure 1: Ablation study for the ε parameter of our proposed domain specific smoothing (DSS). Results are shown for the VisDA 2017 train→test task and Office-Home’s Pr→Ar task before (source only) and after the adaptation step.

main paper) are created per-domain. At the start of every cycle, a domain d is chosen from the available source domains, whose bin $\mathcal{B}_c^{S_d}$ is then used for the construction of mini-batches.

3 Ablation Studies

3.1 Hyperparameter ε

In Figure 1, we provide an ablation study for the ε parameter of our proposed domain specific smoothing (DSS) setup. Results are generated on the VisDA 2017 [15] test set and Office-Home’s [23] Pr→Ar task after the source only pretraining phase and after the adaptation step (using the $\text{DSS}_{\text{Pre}}^S$, $\text{DSS}_{\text{Ada}}^S$ setup from the main paper). Setting $\varepsilon = 0$ is equivalent to not using label smoothing at all ($\text{DSS}_{\text{Pre}}^\times$, $\text{DSS}_{\text{Ada}}^\times$) and thus yields the worst results. In line with our hypothesis in the main paper, using label smoothing increases the domain adaptation capabilities. We notice that the results are robust for $\varepsilon \geq 0.15$ considering both source only pretraining and the adapted results. Overall, setting $\varepsilon = 0.25$ yields the most consistent performance for the examined transfer tasks and was hence chosen as basis for our further experiments.

3.2 Training Stability

In Table 3, we additionally analyze UBR²S’ stability over multiple runs with consecutive random seeds. Results are reported for the VisDA 2017 [15] train→val and train→test transfer tasks and averaged over 3 runs. Expectedly, the source only pretraining exhibits larger fluctuations as the model has not been trained on any target domain data at this point. However, after the adaptation step with UBR²S, results consistently converge towards similar values with very minor deviations of $\pm 0.3\%$ and $\pm 0.1\%$ accuracy for the validation and test set respectively. Similar results can also be noted for the VisDA challenge evaluation protocol metric (mean class accuracy). We thus conclude that our proposed UBR²S method is also stable over multiple runs with different random seeds.

4 Multi-source DA

We also evaluate UBR²S in a multi-source domain adaptation setting and report results in Table 1. For this, we employ the setup of [11, 12] and report results for Office-31 (D,W→A) and Office-Home (Ar,Cl,Pr→Rw) in two settings: *Source combine* merges all available source domains into a single, larger source domain while the *multi-source* setup keeps information about the source domain affiliation of all source samples available during training. For Office-31, UBR²S is able to surpass DSBN [11] by 4.4% in the source combine setting and 1.7% in the multi-source setting. For Office-Home, UBR²S is able to outperform MFSAN [12] by 0.5% and DSBN [11] by 0.9% in the source combine setup. UBR²S does also exceed MFSAN [12] by 1.7% and the very recent WAMDA [12] method by 1.2% in the multi-source setup. These results imply that our proposed UBR²S method also has an advantage in multi-source domain adaptation settings and is able to learn from multiple different source domains at once.

5 Backbone Architectures

One of the key advantages of UBR²S is that it can be applied to any off-the-shelf network. No additional layers or auxiliary networks (*e.g.* generators or discriminators) are required. We therefore also provide results for different network architectures to show UBR²S’ performance w.r.t. parameter count. We choose ResNet-50 [13], ResNet-101 [13], DenseNet-121 [14] and MobileNetV2 [15] for comparison and evaluate on the VisDA 2017 validation set. Parameter counts and results are depicted in Table 2 as both accuracy and mean class accuracy as per challenge evaluation protocol.

Most commonly, results are reported using ResNet-50 and ResNet-101. In these categories, UBR²S is able to outperform very recently proposed methods such as STAR [16], UFAL [16] and the approach proposed by Li *et al.* [17]. When comparing ResNet-50 to ResNet-101 results, we note that UBR²S’ accuracy only drops by 3.2%, even though ResNet-50 has approximately 19 million fewer parameters. Additionally, even at this reduced parameter count, UBR²S is able to outperform the larger ResNet-152 results of SimNet [18] and GTA [18] by a large margin.

For MobileNetV2 and DenseNet-121, no comparative values exist in literature. However, it is noteworthy that UBR²S is still able to perform on par with ResNet-50 results at one third (DenseNet-121) and one tenth (MobileNetV2) of their parameters. We thus show that UBR²S is indeed applicable to arbitrary off-the-shelf network architectures. This opens up opportunities for faster research prototyping, lower resource requirements while training and the option for an accuracy-speed trade-off in deployment scenarios.

Method	Source Combine		Multi-Source	
	D,W→A	Ar,Cl,Pr→Rw	D,W→A	Ar,Cl,Pr→Rw
BN [10]	71.3	81.2	69.9	81.4
WAMDA [14]	—	—	72.0	82.3
MFSAN [14]	67.6	82.7	72.7	81.8
DSBN [10]	73.2	82.3	75.6	83.0
UBR²S (ours)	77.6	83.2	77.3	83.5

Table 1: Classification accuracy (in %) for different methods when leveraging multiple source domains of the Office-31 (D, W) and Office-Home (Ar, Cl, Pr) datasets.

Backbone	Parameters	Method	Acc.	Mean Acc.
MobileNetV2	2.2×10^6	UBR²S (ours)	70.1	69.5
DenseNet-121	7.0×10^6	UBR²S (ours)	77.6	79.5
ResNet-50	23.5×10^6	GTA [14]	69.5	—
		SimNet [14]	—	69.6
		CDAN+E [14]	70.0	—
		TAT [14]	71.9	—
		DTA [8]	—	76.2
		UBR²S (ours)	79.0	79.8
ResNet-101	42.5×10^6	DSBN [10]	—	80.2
		DTA [8]	—	81.5
		STAR [14]	—	82.7
		Li <i>et al.</i> [14]	—	83.3
		UFAL [14]	81.8	84.7
		UBR²S (ours)	82.2	85.2
ResNet-152	58.1×10^6	SimNet [14]	—	72.9
		GTA [14]	77.1	—

Table 2: Classification accuracy and mean class accuracy (in %) for different network architectures and methods on the VisDA 2017 validation set.

Subset	Method	aero	bicyc	bus	car	horse	knife	motor	person	plant	skate	train	truck	Mean Acc.	Accuracy
Train→Val	Source only	59.8±10.9	19.8±7.7	63.4±3.8	72.9±3.3	73.8±2.9	15.6±2.3	81.5±5.1	39.8±8.4	71.7±3.8	31.2±2.5	87.7±0.8	7.8±0.6	52.1±2.4	57.1±1.9
Train→Val	UBR ² S	97.6±0.2	83.3±1.0	81.7±1.9	70.9±2.7	95.7±0.4	93.3±3.1	89.1±0.6	84.3±1.5	94.8±1.1	91.4±1.5	89.0±0.8	52.8±2.5	85.3±0.2	82.5±0.3
Train→Test	Source only	54.4±10.3	10.4±4.4	70.6±3.3	95.3±1.0	61.0±2.5	15.7±3.3	71.3±7.0	27.4±7.8	86.0±2.3	29.5±4.4	84.7±2.3	18.3±0.4	52.1±1.7	54.7±1.6
Train→Test	UBR ² S	96.6±0.1	88.5±2.1	88.3±0.8	95.2±0.6	92.4±1.2	95.1±2.3	77.5±0.3	78.9±0.4	96.4±1.1	85.4±3.9	93.0±1.2	87.0±2.0	89.5±0.3	89.0±0.1

Table 3: Stability analysis for our final UBR²S method on the VisDA 2017 validation and test set averaged over 3 runs with consecutive random seeds. Results are obtained with a ResNet-101 backbone.

Method	Ar			Cl			Pr			Rw			Avg.
	Cl	Pr	Rw	Ar	Pr	Rw	Ar	Cl	Rw	Ar	Cl	Pr	
TAT [14]	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
ETD [8]	51.3	71.9	85.7	57.6	69.2	73.7	57.8	51.2	79.3	70.2	57.5	82.1	67.3
MDDA [14]	54.9	75.9	77.2	58.1	73.3	71.5	59.0	52.6	77.8	67.9	57.6	81.8	67.3
CDAN+TransNorm [14]	50.2	71.4	77.4	59.3	72.7	73.1	61.0	53.1	79.5	71.9	59.0	82.9	67.6
CADA-P [8]	56.9	76.4	80.7	61.3	75.2	75.2	63.2	54.5	80.7	73.9	61.5	84.1	70.2
GSDA [8]	61.3	76.1	79.4	65.4	73.3	74.3	65.0	53.2	80.0	72.2	60.6	83.1	70.3
UFAL [14]	58.5	75.4	77.8	65.2	74.7	75.0	64.9	58.0	79.9	71.6	62.3	81.0	70.4
CAPLS [14]	56.2	78.3	80.2	66.0	75.4	78.4	66.4	53.2	81.1	71.6	56.1	84.3	70.6
UBR²S (ours)	58.3	75.5	78.7	67.8	72.6	71.4	67.0	58.7	79.0	73.7	61.8	82.2	70.6

Table 4: Classification accuracy (in %) for different methods on the Office-Home dataset with domains Art, Clipart, Product and Real-world.

Method	aero	bicyc	bus	car	horse	knife	motor	person	plant	skate	train	truck	Avg.
Source only	59.1	23.3	59.1	76.1	73.8	15.7	75.7	47.0	70.5	31.6	88.5	7.3	52.3
SimNet-152 [10]	94.3	82.3	73.5	47.2	87.9	49.2	75.1	79.7	85.3	68.5	81.1	50.3	72.9
DSBN [11]	94.7	86.7	76.0	72.0	95.2	75.1	87.9	81.3	91.1	68.9	88.3	45.5	80.2
DTA [8]	93.7	82.8	85.6	83.8	93.0	81.0	90.7	82.1	95.1	78.1	86.4	32.1	81.5
STAR [12]	95.0	84.0	84.6	73.0	91.6	91.8	85.9	78.4	94.4	84.7	87.0	42.2	82.7
Li <i>et al.</i> [13]	95.7	78.0	69.0	74.2	94.6	93.0	88.0	87.2	92.2	88.8	85.1	54.3	83.3
SE-152 [8]	95.9	87.4	85.2	58.6	96.2	95.7	90.6	80.0	94.8	90.8	88.4	47.9	84.3
UFAL [14]	97.6	82.4	86.6	67.3	95.4	90.5	89.5	82.0	95.1	88.5	86.9	54.0	84.7
UBR²S (ours)	97.5	84.4	81.5	67.9	95.5	89.8	88.7	85.8	93.6	93.0	89.4	55.6	85.2

Table 5: Per class accuracy (in %) for different methods on the VisDA 2017 validation set as per challenge evaluation protocol. Results are obtained with ResNet-101 unless otherwise denoted.

DSS _{Pre}	DSS _{Ada}	Reweigh	S→R _{val} Pre	S→R _{val}	S→R _{test} Pre	S→R _{test}	Ar→Cl Pre	Ar→Cl	Pr→Ar Pre	Pr→Ar
×	×	×	50.5	77.9	49.4	78.7	44.0	52.5	51.1	58.1
\mathcal{S}	×	×	52.3	77.5	51.3	82.5	46.0	54.1	54.1	58.8
\mathcal{S}	\mathcal{S}	×	52.3	77.9	51.3	83.8	46.0	54.6	54.1	61.8
\mathcal{S}	\mathcal{T}	×	52.3	66.3	51.3	67.0	46.0	38.6	54.1	47.5
\mathcal{S}	\mathcal{S}, \mathcal{T}	×	52.3	65.7	51.3	67.8	46.0	47.4	54.1	53.6
\mathcal{S}	\mathcal{S}	SL	52.3	81.4	51.3	85.4	46.0	56.7	54.1	64.1
\mathcal{S}	\mathcal{S}	DE	52.3	85.0	51.3	89.5	46.0	57.4	54.1	65.1
\mathcal{S}	\mathcal{S}	DE+SL	52.3	85.2	51.3	89.8	46.0	58.3	54.1	67.0

Table 6: Ablation study using ResNet-101 on the VisDA 2017 S→R task (mean class accuracy) for both validation and test set as well as ResNet-50 on the Office-Home Ar→Cl and Pr→Ar tasks (accuracy). Transfer tasks marked with $\xrightarrow{\text{Pre}}$ indicate results after the source only pretraining and before the adaptation step. The last table row represents our full UBR²S method.

References

- [1] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019.
- [2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [3] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=rkpoTaxA->.
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [5] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Unsupervised domain adaptation with hierarchical gradient synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4043–4052, 2020.
- [6] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
- [7] Vinod Kumar Kurmi, Shanu Kumar, and Vinay P Namboodiri. Attending to discriminative certainty for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 491–500, 2019.
- [8] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 91–100, 2019.
- [9] Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. Enhanced transport distance for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [10] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.
- [11] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *International Conference on Machine Learning*, pages 4013–4022, 2019.
- [12] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

- [13] Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [14] Adam Paszke, Sam Gross, Soumith Chintala, et al. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.
- [15] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017.
- [16] Pedro O Pinheiro. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8004–8013, 2018.
- [17] Tobias Ringwald and Rainer Stiefelhausen. Unsupervised Domain Adaptation by Uncertain Feature Alignment. In *The British Machine Vision Conference (BMVC)*, 2020.
- [18] Alina Roitberg, Ziad Al-Halah, and Rainer Stiefelhausen. Informed democracy: voting-based novelty detection for action recognition. *arXiv preprint arXiv:1810.12819*, 2018.
- [19] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*. Springer, 2010.
- [20] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [21] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.
- [22] Jogendra Nath Kundu Surbhi Aggarwal et al. WAMDA: Weighted Alignment of Sources for Multi-source Domain Adaptation. In *The British Machine Vision Conference (BMVC)*, 2020.
- [23] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [24] Jindong Wang, Yiqiang Chen, Wenjie Feng, Han Yu, Meiyu Huang, and Qiang Yang. Transfer learning with dynamic distribution adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(1):1–25, 2020.
- [25] Qian Wang, Penghui Bu, and Toby P Breckon. Unifying unsupervised domain adaptation and zero-shot visual recognition. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.
- [26] Ximei Wang, Ying Jin, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Transferable normalization: Towards improving transferability of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 1951–1961, 2019.
- [27] Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2019.