

Supplementary Materials: Alleviating the Equilibrium Challenge with Sample Virtual Labeling for Adversarial Domain Adaptation

Anonymous Authors

0.1 More Theoretical Insight

Most existing domain adaption methods are based on the adaption theory proposed by [1], which is described as follows:

THEOREM 0.1. Let \mathcal{H} be the hypothesis space and $\epsilon_S(h)$ and $\epsilon_T(h)$ are the generalization error of a hypothesis $h \in \mathcal{H}$ on the source domain \mathcal{D}_s and target domain \mathcal{D}_t , respectively. For any hypothesis $h \in \mathcal{H}$, there is:

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2}d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda \quad (1)$$

The Eq.(1) shows that the upper bound of the expected target error is mainly relative to three terms: (i) The first term, the expected source error $\epsilon_S(h)$, is expected to be small because reliable labels are owned in the source domain; (ii) As for the second term, the domain discrepancy $d_{\mathcal{H}}(\mathcal{S}, \mathcal{T})$ can be made smaller by alignment learning; (iii) The third term, the combined error λ of the ideal joint hypothesis h^* (i.e., $h^* = \arg \min_{h \in \mathcal{H}} (\epsilon_S(h) + \epsilon_T(h))$), is considered sufficiently small.

With the thoughts above, to minimize the upper bound of the expected target error, the designed model needs to keep the second term (i.e., $d_{\mathcal{H}}(\mathcal{S}, \mathcal{T})$) of the above to a small value. For the cross-domain adversarial alignment, the domain discrepancy represented by $d_{\mathcal{H}}(\mathcal{S}, \mathcal{T})$ is usually measured by \mathcal{H} -divergence.

Potential Problem of \mathcal{H} -divergence. As for the sample x_s and x_t drawn from the source domain \mathcal{S} (domain label $d=1$) and target domain \mathcal{T} (domain label $d=0$), the domain discriminator \mathcal{D} aims to predict the source sample x_s to be 1 and target sample x_t to be 0. Then \mathcal{H} -divergence across domains is formulated as:

$$d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) = 2[1 - \min(\text{err}(\mathcal{D}(x_s)) + \text{err}(\mathcal{D}(x_t)))] \quad (2)$$

where $\text{err}(\mathcal{D}(x_s))$ and $\text{err}(\mathcal{D}(x_t))$ represent the prediction error of the domain classifier \mathcal{D} on the source and the target domain samples, respectively. The Eq.(2) implies that the domain distance between source and target is inversely proportional to the error rate of the domain classifier \mathcal{D} . Intuitively, larger domain prediction error means smaller domain discrepancy.

Rationality of Virtual- \mathcal{H} -divergence. In order to fully fool the discriminator, we propose the *Virtual- \mathcal{H} -divergence* which aims at minimizing the domain discrepancy without information loss. Specifically, we introduce the *virtual* source domain $\hat{\mathcal{S}}$ (domain label $d=0$) and the *virtual* target domain $\hat{\mathcal{T}}$ (domain label $d=1$) by copying the real source \mathcal{S} (domain label $d=1$) and target \mathcal{T} (domain label $d=0$), and assign labels different from the real domain to improve the error rate of the classifier h . We measure the distances between \mathcal{S} and $\hat{\mathcal{S}}$, and the distances between the \mathcal{T} and $\hat{\mathcal{T}}$. Formally,

$$d_{\mathcal{H}}(\mathcal{S}, \hat{\mathcal{S}}) = 2[1 - \min(\text{err}(\mathcal{D}(x_s)) + \text{err}(\mathcal{D}(\hat{x}_s)))] \quad (3)$$

$$d_{\mathcal{H}}(\hat{\mathcal{T}}, \mathcal{T}) = 2[1 - \min(\text{err}(\mathcal{D}(\hat{x}_t)) + \text{err}(\mathcal{D}(x_t)))] \quad (4)$$

Due to the $\text{err}(\cdot)$ refers to the cross entropy loss with the averaging operation, and the source and target domains have the same number

of instance-level samples during the batch training, the *Virtual- \mathcal{H} -divergence* can be written as:

$$\begin{aligned} d_{V-\mathcal{H}}(\mathcal{S}, \mathcal{T}) &= \frac{1}{2}[d_{\mathcal{H}}(\mathcal{S}, \hat{\mathcal{S}}) + d_{\mathcal{H}}(\hat{\mathcal{T}}, \mathcal{T})] \\ &= 2 - \min(\text{err}(\mathcal{D}(\hat{x}_s)) + \text{err}(\mathcal{D}(x_s)) \\ &\quad + \text{err}(\mathcal{D}(\hat{x}_t)) + \text{err}(\mathcal{D}(x_t))) \\ &= 2 \left[1 - \min(\text{err}(\mathcal{D}(x_s \oplus \hat{x}_s)) \right. \\ &\quad \left. + \text{err}(\mathcal{D}(\hat{x}_s \oplus x_t))) \right] \\ &= d_{\mathcal{H}}(\mathcal{S} \oplus \hat{\mathcal{T}}, \hat{\mathcal{S}} \oplus \mathcal{T}) \end{aligned} \quad (5)$$

where \oplus denotes the *union* operator. Because the features of the $(\mathcal{S} \oplus \hat{\mathcal{T}})$ and $(\hat{\mathcal{S}} \oplus \mathcal{T})$ are identical except for the domain labels, the error rate of the domain classifier \mathcal{D} is very large. This means the value of $d_{\mathcal{H}}(\mathcal{S} \oplus \hat{\mathcal{T}}, \hat{\mathcal{S}} \oplus \mathcal{T})$ is much smaller than $d_{\mathcal{H}}(\mathcal{S} \oplus \mathcal{S}, \mathcal{T} \oplus \mathcal{T})$. Formally,

$$\begin{aligned} d_{V-\mathcal{H}}(\mathcal{S}, \mathcal{T}) &= d_{\mathcal{H}}(\mathcal{S} \oplus \hat{\mathcal{T}}, \hat{\mathcal{S}} \oplus \mathcal{T}) \\ &\leq d_{\mathcal{H}}(\mathcal{S} \oplus \mathcal{S}, \mathcal{T} \oplus \mathcal{T}) = d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) \end{aligned} \quad (6)$$

Considering Eq.(1) and Eq.(6) jointly, the upper bound of the expected target error can be written as:

$$\begin{aligned} \epsilon_T(h) &\leq \epsilon_S(h) + \frac{1}{2}d_{V-\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda \\ &\leq \epsilon_S(h) + \frac{1}{2}d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda \end{aligned} \quad (7)$$

In this way, the upper bound of the expected target error, i.e., $\epsilon_T(h)$, can be effectively reduced in our work.

0.2 More Experimental Results

Visualization. For Figure 4 (a) and (b) in our main manuscript, to save space, we only present the visualization results of one improved method. Here, we also present the detailed results for other improved methods in Figure 1. The detection results fully demonstrate our method VFDD could alleviate the domain shift and improve the universality of detectors. The visualization feature distributions employed VFDD have better clustering effect and have fewer samples distributed across class boundaries, which intuitively boosts the feature discriminability.

Comparisons with SOTA Methods. In our main manuscript, Tables 1, 2, and 3 primarily pertain to object detection. To showcase the effectiveness of VFDD, we extend our experimentation to the image classification datasets of Office-31, as presented in Table 1. We can observe that our method VFDD can be easily plugged and played in the existing alignment-based UDA methods to enhance their recognition performance on image classification task. The methods(DANN [4], JAN [11], CAT [3], ETD [7], MCD [12], CDAN [10], TADA [13], MDD [15], GVB [2], DWL [14], DAN [9], DRCN

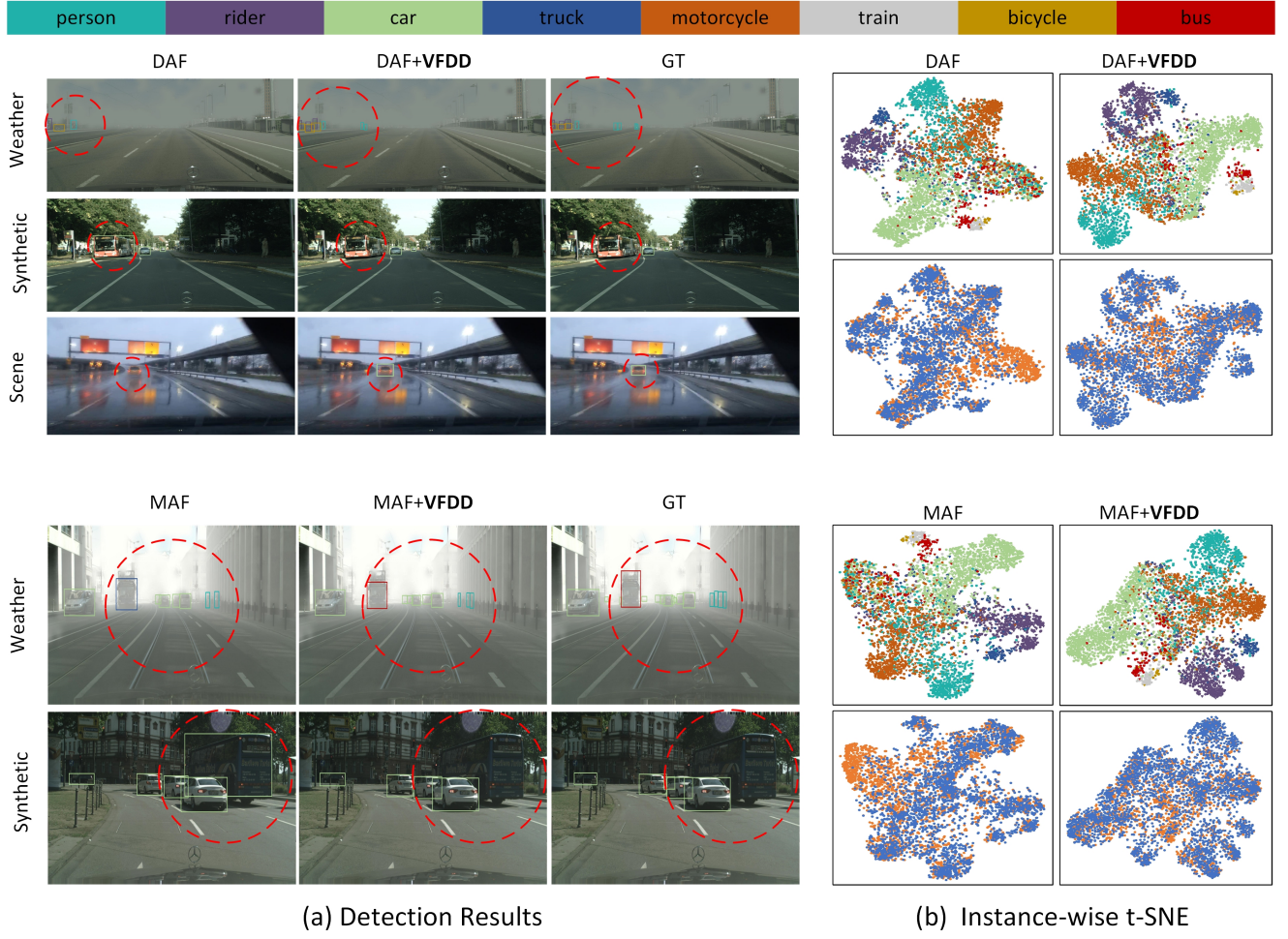


Figure 1: (a) Qualitative comparison of DAF+VFDD and MAF+VFDD with previous SOTA method and GT in different scenarios. The red circle area reflects the superiority of our method. (b) In Cityscapes to Foggy Cityscapes, instance-level feature t-SNE results. Colors in the first row represent classes, while orange signifies the source domain and blue signifies the target domain in second row.

Methods	A → D	A → W	D → W	W → D	D → A	W → A	Average
DANN	79.7	82.0	96.9	99.1	68.2	67.4	82.2
JAN	84.7	85.4	97.4	99.8	68.6	70.0	84.3
CAT	90.6	91.1	98.6	99.6	70.4	66.5	86.1
ETD	88.0	92.1	100.0	100	71.0	67.8	86.2
MCD	92.2	88.6	98.5	100.0	69.5	69.7	86.5
CDAN	92.9	93.1	98.6	100.0	71.0	69.3	87.5
TADA	91.6	94.3	98.7	99.8	72.9	73.0	88.4
MDD	93.5	94.5	98.4	100.0	74.6	72.2	88.9
GVB	91.4	92.0	98.7	100.0	74.9	73.4	88.3
DWL	91.2	89.2	99.2	100.0	73.1	69.8	87.1
CDAN(baseline)	92.9	93.1	98.6	100.0	71.0	69.3	87.5
CDAN+VFDD	94.5(+1.6)	94.3(+1.2)	99.2(+0.6)	100.0(+0.0)	74.6(+3.6)	72.7(+3.4)	89.2(+1.7)

Table 1: Performance (%) comparisons with the previous UDA approaches on Office-31. All experiments are conducted based on ResNet-50 pre-trained on ImageNet.

Methods	per	rider	car	truck	bus	train	mcy	bicy	mAP
DAF	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
DAF+Flip	28.7	40.6	42.2	20.4	33.3	5.3	22.8	33.6	28.4(+0.8)
DAF+Repeat	27.4	39.6	42.0	22.3	35.0	11.8	20.2	31.8	28.7(+1.1)
DAF+VFDD	31.4	40.8	43.3	16.4	38.7	27.6	23.5	33.2	31.9(+4.3)
MAF	28.2	39.5	43.9	23.8	39.9	33.3	29.2	33.9	34.0
MAF+Flip	32.1	44.9	43.5	24.6	34.9	26.4	32.4	36.6	34.4(+0.4)
MAF+Repeat	32.9	43.4	47.9	26.2	39.1	31.4	23.2	33.3	34.7(+0.7)
IDF+VFDD	30.6	41.7	46.1	24.7	42.0	43.1	30.7	35.5	36.8(+2.8)

Table 2: Results on the task from Cityscape to Foggy Cityscape.

[5], CoGAN [8], CyCADA [6]) contrasted herein are commonplace methodologies; as such, exhaustive elaboration of their particulars is deemed unnecessary.

Comparisons with Data Augmentation Methods. In Table 2, a comparative analysis is conducted between VFDD and conventional data augmentation techniques (Flip, Repeat) for the task of transitioning from Cityscape to Foggy Cityscape. It should be noted that the term ‘virtual copy’ of our VFDD refers to altering the domain label (be it real or virtual) while retaining the original feature representation of each sample. As evidenced by the results presented in Table 2, the proposed VFDD method demonstrates a significant performance advantage over traditional data augmentation methods.

REFERENCES

- [1] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2007. Analysis of representations for domain adaptation. In *NeurIPS*. 137–144.
- [2] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Chi Su, Qingming Huang, and Qi Tian. 2020. Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*. 12455–12464.
- [3] Zhijie Deng, Yucen Luo, and Jun Zhu. 2019. Cluster alignment with a teacher for unsupervised domain adaptation. In *CVPR*. 9944–9953.
- [4] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *ICML*. PMLR, 1180–1189.
- [5] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. 2016. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*. 597–613.
- [6] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*. Pmlr, 1989–1998.
- [7] Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. 2020. Enhanced transport distance for unsupervised domain adaptation. In *CVPR*. 13936–13944.
- [8] Ming-Yu Liu and Oncel Tuzel. 2016. Coupled generative adversarial networks. In *NeurIPS*, Vol. 29.
- [9] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*. PMLR, 97–105.
- [10] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. 2018. Conditional adversarial domain adaptation. In *NeurIPS*, Vol. 31.
- [11] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *ICML*. PMLR, 2208–2217.
- [12] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*. 3723–3732.
- [13] Ximei Wang, Liang Li, Weirui Ye, Mingsheng Long, and Jianmin Wang. 2019. Transferable attention for domain adaptation. In *AAAI*, Vol. 33. 5345–5352.
- [14] Ni Xiao and Lei Zhang. 2021. Dynamic weighted learning for unsupervised domain adaptation. In *CVPR*. 15242–15251.
- [15] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. 2019. Bridging theory and algorithm for domain adaptation. In *ICML*. 7404–7413.