Supplementary Material for Unsupervised Domain Adaptive Object Detection using Forward-Backward Cyclic Adaptation

Siqi Yang[1], Lin Wu[2], Arnold Wiliem[1], and Brian C. Lovell[1]

[1] The University of Queensland, Brisbane, Australia
[2] Hefei University of Technology, Hefei, China
siqi.yang@uq.net.au, xiaoxian.wu9188@gmail.com, arnold.wiliem@ieee.org, lovell@itee.uq.edu.au

In the following supplementary material, we fist provide detailed theoretical analysis to illustrate how our proposed Forward and Backward Cyclic Adaptation (FBC) in Algorithm 1 approximates the objective function of gradient alignment (to Eq.3 in our main submission). Details are shown in Section 1. We then provide more ablation studies in Section 2 and implementation details in Section 3. Section 4 demonstrates more examples of feature visualization on the Watercolor [1].

1 Deriving the Objective Function

We detail the theoretical analysis in the main submission to show how the proposed algorithm approximates the objective function of gradient alignment. We follow the conventions in Reptile [2] and demonstrate the gradient computations during the training. In Reptile [2], they effectively extrapolated the gradient with a number of steps taken. Let us first denote the terms following [2, 3]:

$$g_i = \frac{\partial \mathcal{L}_i(\theta_i)}{\partial \theta_i} \quad \text{(gradient obtained during SGD)}, \tag{1}$$

$$\theta_{i+1} = \theta_i - \alpha g_i \quad \text{(squence of parameter vectors)}, \tag{2}$$

$$\bar{g}_i = \frac{\partial \mathcal{L}_i(\theta_i)}{\partial \theta_1} \quad \text{(gradient at initial point)}, \tag{3}$$

$$g_i^j = \frac{\partial \mathcal{L}_i(\theta_i)}{\partial \theta_j} \quad \text{(gradient evaluated at point i with respect to parameters j)}, \tag{4}$$

$$\bar{H}_i = \frac{\partial^2 \mathcal{L}_i(\theta_i)}{\partial \theta_1^2} \quad \text{(Hessian at initial point)}, \tag{5}$$

$$H_i^j = \frac{\partial^2 \mathcal{L}_i(\theta_i)}{\partial \theta_j^2} \quad \text{(Hessian evaluated at point i with respect to parameters j)}, \tag{6}$$

where the α is learning rate and $\mathcal{L}_i$ is the loss function on the samples for each gradient updates.

According to Taylor's theorem, we have the SGD gradients as follows:

$$g_i = L_i'(\theta_1) = \mathcal{L}_i'(\theta_1) + \mathcal{L}_i''(\theta_i - \theta_1) + O(\|\theta_i - \theta_1\|^2), \tag{7}$$

$$= \bar{g}_i + \bar{H}_i(\theta_i - \theta_1) + O(\|\theta_i - \theta_1\|^2) \quad \text{(using definition of } \bar{g}_i, \bar{H}_i), \tag{8}$$

$$= \bar{g}_i - \alpha \bar{H}_i \sum_{j=1}^{i-1} g_j + O(\|\theta_i - \theta_1\|^2) \quad \text{(using gradient updates } \theta_i - \theta_1 = -\alpha \sum_{j}^{i-1} g_j), \tag{9}$$

$$= \bar{g}_i - \alpha \bar{H}_i \sum_{j=1}^{i-1} \bar{g}_j + O(\|\theta_i - \theta_1\|^2) \quad \text{(using } g_j = \bar{g}_j + O(\|\theta_i - \theta_1\|^2)). \tag{10}$$

If we consider there are two steps of parameter updates with stochastic gradient descent (SGD), where the gradient of the first step is g_1 and the one of second step is g_2. According to the Eq. 10, we have

$$g_1 = \bar{g}_1, \tag{11}$$

$$g_2 = \bar{g}_2 - \alpha \bar{H}_2 \bar{g}_1 + O(\|\theta_i - \theta_1\|^2). \tag{12}$$

Then, the overall gradient of the two SGD steps is

$$g = g_1 + g_2 = \bar{g}_1 + \bar{g}_2 - \alpha \bar{H}_2 \bar{g}_1 + O(\|\theta_i - \theta_1\|^2). \tag{13}$$

In Reptile [2], they noted that

$$\epsilon[\bar{H}_2 \bar{g}_1] = \epsilon[\bar{H}_1 \bar{g}_2] = \frac{1}{2}\epsilon[\bar{H}_2 \bar{g}_1 + \bar{H}_1 \bar{g}_2] = \frac{1}{2}\epsilon[\frac{\partial}{\partial \theta_1}(\bar{g}_1 \bar{g}_2)], \tag{14}$$

where the ϵ is the expected loss. Therefore, the overall expected loss is

$$\epsilon[g] = \epsilon[\bar{g}_1] + \epsilon[\bar{g}_2] - \frac{1}{2}\alpha\epsilon[\frac{\partial}{\partial \theta_1}(\bar{g}_1 \bar{g}_2)]. \tag{15}$$

In our work, we aim to address the domain adaptation problem for object detection. In our proposed forward and backward cyclic adaptation (Algorithm 1), we train the network with episodic training. In each episode, similar to the two-step SGD updates discussed above, we first perform the backward hopping on labeled source samples $\{\mathcal{X}_S, \mathcal{Y}_S\}$ to obtain the parameters θ_S, and then we initialize the forward passing with θ_S and train the network with pseudo labeled target samples $\{\mathcal{X}_T, \hat{\mathcal{Y}}_T\}$, obtaining the updated parameters θ_T. The shared model θ is updated by θ_S and θ_T sequentially. We can consider the gradient of forward passing, g_S, as g_1, and similarly g_T as g_2. Then we can substitute g_S and g_T to Eq. 15:

$$\mathbb{E}[g_e] = \mathbb{E}[\bar{g}_S] + \mathbb{E}[\bar{g}_T] - \frac{1}{2}\alpha\epsilon[\frac{\partial}{\partial \theta_S}(\bar{g}_S \bar{g}_T)] \ , \tag{16}$$

where $\mathbb{E}$ is the expected loss. The above equation shows that the training of our proposed adaptation method (Algorithm 1) is approximating the objective of gradient alignment:

$$\min_{\theta_S,\theta_T} \mathcal{L}_{\theta_S}(\mathcal{X}_S, \mathcal{Y}_S) + \mathcal{L}_{\theta_T}(\mathcal{X}_T, \hat{\mathcal{Y}}_T) - \alpha \frac{\partial \mathcal{L}_{\theta_S}(\mathcal{X}_S, \mathcal{Y}_S)}{\partial \theta_S} \cdot \frac{\partial \mathcal{L}_{\theta_T}(\mathcal{X}_T, \hat{\mathcal{Y}}_T)}{\partial \theta_T} \ . \quad (17)$$

2 More Ablation Studies

In this section, we evaluate the effects of the different components in our proposed adaptation method. As shown in the Eq.11 and Eq.12 in our submission, our overall objective function is

$$\begin{aligned} \min_{\theta} \mathcal{L} &= \mathcal{L}_{inv}(\mathcal{X}_S, \mathcal{Y}_S, \mathcal{X}_T) + \gamma \mathcal{L}_{div}(\mathcal{X}_S, \mathcal{X}_T) \\ &= \mathcal{L}_g(\mathcal{X}_S, \mathcal{Y}_S, \mathcal{X}_T) + \lambda \mathcal{L}_{adv}(\mathcal{X}_S, \mathcal{X}_T) + \gamma \mathcal{L}_{div}(\mathcal{X}_S, \mathcal{X}_T) \ , \end{aligned}$$

where $\mathcal{L}_g$ is the loss of gradient alignment, $\mathcal{L}_{adv}$ is the loss of local feature alignment via adversarial training and $\mathcal{L}_{div}$ is the loss of domain-diversity. λ and γ are the hyperparameters and we set $\lambda = 0.5$ and $\gamma = 0.1$ for all the experiments in this work.

In the following sections, we use G, L, and D to indicate gradient alignment, local feature alignment and domain diversity, respectively.

2.1 Effects of Gradient Alignment

To evaluate the effects of gradient alignment, we perform the forward-backward cyclic method (FBC) on four different cross-domain scenarios with gradient alignment only. The results are shown in Table 1 - 4. In the adaptation scenarios, PASCAL [4]-to-Clipart [1] (in Table 1) and PASCAL-to-Watercolor [1] (in Table 2), the FBC with gradient alignment can achieve better adaptation results than the FBC with local feature alignment only. It is because the domain discrepancy in these two adaptation scenarios is large, *i.e.,* adapting real objects to cartoon or watercolor objects. This indicates that gradient alignment has its superiority in aligning high-level semantics.

However, in the adaptation scenarios, Sim10k [5]-to-Cityscapes [6] (in Table 3 and Cityscapes [6]-to-FoggyCityscapes [7] (in Table 4 , the domain discrepancy between two domain are mainly in the low-level features, *e.g.,* textures and colors. Therefore, in these scenarios, the FBC with gradient alignment only can achieve limited gain on mAP, compared with the FBC with local feature alignment. It is more evident in Cityscapes-to-FoggyCityscapes, where the foggy images are rendered from the real images. However, the FBC with gradient alignment only is still 4.6% higher than the source only model (in Table 4). Although the FBC with local feature alignment can obtain a high mAP with 33.7%, in combination with gradient alignment and domain diversity, the mAP can be boosted to 36.7%.

2.2 Effects of Local Feature Alignment

The local feature alignment is conducted via adversarial training, which aligns the marginal feature distributions between the source and target domains. As discussed in the main submission, the alignment of marginal feature distributions does not perform well when the domain discrepancy is large. This is also demonstrated in our experiments. In Table 1 and Table 2, the FBC with local feature alignment only does not perform better than the gradient alignment, when the domain discrepancy is large. But when the domain discrepancy is small, *i.e.,* in low-level semantics, the FBC with local feature alignment demonstrates its superiority, as shown in Table 3 and Table 4.

It is worthy to mention that the gradient alignment and local feature alignment are complementary, as gradient alignment can achieve category-level alignment for high-level semantics and local feature alignment via adversarial training has its advantages for aligning low-level semantics. The combination of these two alignment and domain diversity can achieve the state-of-the-art performance.

2.3 Effects of Domain Diversity

Here we evaluate the effects of the domain-diversity. As shown in Table 1 - Table 4, the domain diversity can consistently improve the adaptation results.

We also analyze the sensitivity of hyper-parameter γ on the adaptation from Cityscapes to FoggyCityscapes. Results are shown in Table 5. It shows that when the value of γ is too large, the entropy regularization in domain diversity will affect the accuracy of classification.

Method	G	L	D	aero	bcy-cle	bird	boat	bot-tle	bus	car	cat	chair	cow	ta-ble	dog	hrs	mo-tor	prsn	plnt	sheep	sofa	train	tv	mAP
Source Only				24.2	47.1	24.9	17.7	26.6	47.3	30.4	11.9	36.8	26.4	10.1	11.8	25.9	74.6	42.1	24.0	3.8	27.2	37.9	29.9	29.5
FBC	✓			28.8	64	21.1	19.1	39.7	60.7	29.5	14.2	46.4	29.3	21.8	8.9	28.8	72.7	51.3	32.9	12.8	28.1	52.7	49.5	35.6
	✓		✓	32.1	57.6	24.4	23.7	34.1	59.3	32.2	9.1	40.3	41.3	27.8	11.9	30.2	72.9	48.8	38.3	6.1	33.1	46.5	48	35.9
		✓		31.8	53.0	21.3	25.0	36.1	55.9	30.4	11.6	39.3	21.0	9.4	14.5	32.4	79.0	44.9	37.8	6.2	35.6	43.0	53.5	34.1
	✓	✓	✓	43.9	64.4	28.9	26.3	39.4	58.9	36.7	14.8	46.2	39.2	11.0	11.0	31.1	77.1	48.1	36.1	17.8	35.2	52.6	50.5	**38.5**

Table 1. The results (%) on the adaptation from PASCAL [4] to Clipart Dataset [1].

Method	G	L	D	bike	bird	car	cat	dog	prsn	mAP
Source Only (ours)				66.7	43.5	41	26.0	22.9	58.9	43.2
FBC (ours)	✓			90.9	46.5	51.3	33.2	29.5	65.9	52.9
	✓		✓	88.7	48.2	46.6	38.7	35.6	64.1	**53.6**
		✓		89.0	47.2	46.1	39.9	27.7	65.0	52.5
	✓	✓	✓	90.1	49.7	44.1	41.1	34.6	70.3	**55.0**

Table 2. The results (%) on the adaptation from PASCAL [4] to Watercolor Dataset [1].

Method	G	L	D	AP on Car
Source Only (ours)				31.2
FBC (ours)	✓			38.2
	✓		✓	39.2
		✓		41.4
	✓	✓	✓	**42.7**

Table 3. The results (%) on the adaptation from Sim10k [5] to Cityscapes Dataset [6].

Method	G	L	D	person	rider	car	truck	bus	train	motor	bcycle	mAP
Source Only (ours)				22.4	34.2	27.2	12.1	28.4	9.5	20.0	27.1	22.9
FBC (ours)	✓			25.8	35.6	35.5	18.4	29.6	10.0	24.5	30.3	26.2
	✓		✓	29.0	37.0	35.6	18.9	32.1	10.7	25.0	31.3	27.5
		✓		31.6	45.1	42.6	26.4	37.8	22.1	29.4	34.6	33.7
	✓	✓	✓	31.5	46.0	44.3	25.9	40.6	39.7	29.0	36.4	**36.7**

Table 4. Results (%) on the adaptation from Cityscapes [6] to FoggyCityscapes Dataset [7].

3 More Implementation Details

In this section, we provide more implementation details of our experiments.

3.1 Details of Local Feature Alignment.

In this work, we utilize the Gradient Reversal Layer (GRL) proposed by Ganin and Lempitsky [8] for adversarial training. We extract local features from low-level layer as input of the domain classifier D and the least-squares loss [9, 10]. To make a fair comparison, our domain classifier is the same as the local domain classifier in SWDA, which consists of three layered convolutional layers with kernel size as 1.

For the local features, the features output from *conv3-3* are extracted in the case of VGG16 model and the features output from the last *res3c* layer are extracted in ResNet101 model. The name of the layer follows the prototxt in Caffe [11].

3.2 Training Details

We optimize the network using Stochastic Gradient Descent (SGD) with a learning rate of 0.001. Following the implementation details of SWDA [12], we resize the training and test images with the shorter side of 600 pixels and set the training batch size as 1. Without specific notation, we set λ as 0.5 and γ as 0.1.

4 Feature Visualization

As a supplement of Fig.5 in the main submission, we also adopt the Grad-cam [13] to visualize the features on the Watercolor dataset in Fig. 1.

γ	0	0.1	0.5
FBC (ours)	35.0	36.7	32.0

Table 5. Results (%) on the adaptation from Cityscapes [6] to FoggyCityscapes Dataset [7].

References

1. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: CVPR. (2018)
2. Nichol, A., Schulman, J.: Reptile: a scalable metalearning algorithm. arXiv preprint arXiv:1803.02999 **2** (2018)
3. Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., Tesauro, G.: Learning to learn without forgetting by maximizing transfer and minimizing interference. ICLR (2019)
4. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88** (2010) 303–338
5. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R.: Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? arXiv preprint arXiv:1610.01983 (2016)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. (2016)
7. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. IJCV (2018) 1–20
8. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML. (2015)
9. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: ICCV. (2017)
10. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: ICCV. (2017)
11. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia, ACM (2014) 675–678
12. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: CVPR. (2019)
13. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV. (2017)

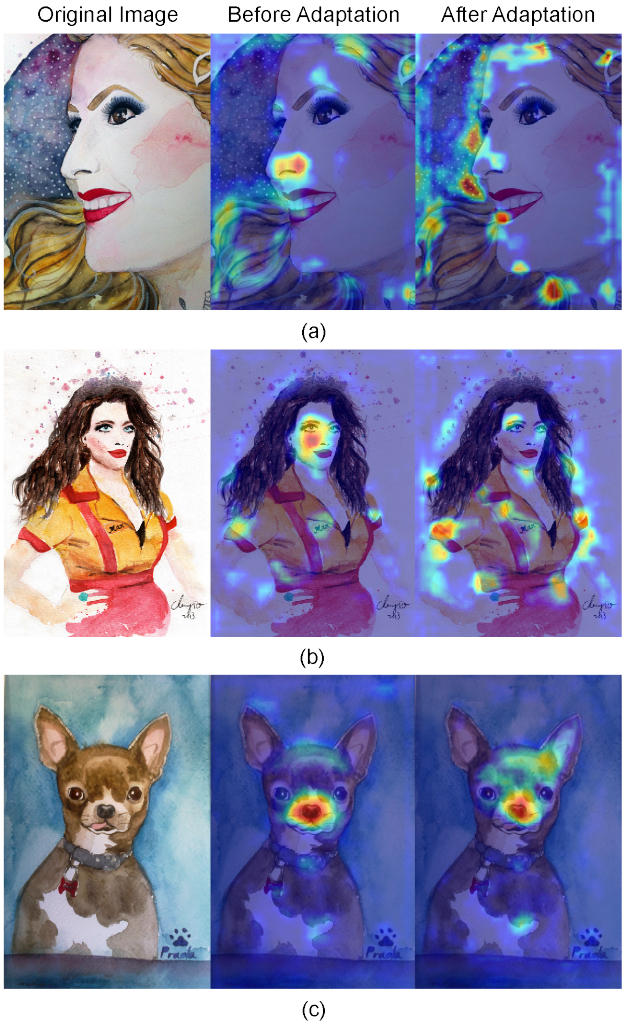

Fig. 1. Feature visualization shows the evidence for improvements in classifiers before and after domain adaptation using Grad-cam [13] on the Watercolor dataset [1]. The images in the middle column show the attention for the classifier before adaptation and the one on the right show the attention for the classifier after adaptation. This figure demonstrates that the adapted detector utilizes more semantics to classify the objects, which indicates the effectiveness of our proposed domain adaptation method.