Adversarial Teacher-Student Representation Learning for Domain Generalization
Supplementary Material

Fu-En Yang\textsuperscript{1,2}, Yuan-Chia Cheng\textsuperscript{1}, Zu-Yun Shiau\textsuperscript{1}, Yu-Chiang Frank Wang\textsuperscript{1,2}
\textsuperscript{1}Graduate Institute of Communication Engineering, National Taiwan University, Taiwan
\textsuperscript{2}ASUS Intelligent Cloud Services, Taiwan
{f07942077, r08942154, r09942069, ycwang}@ntu.edu.tw

A Pseudo-Code of Adversarial Teacher-Student Representation Learning

We provide the pseudo-code of our Adversarial Teacher-Student Representation Learning in Algorithm 1.

\begin{algorithm}
\caption{Adversarial Teacher-Student Representation Learning}
\begin{algorithmic}[1]
\Statex \textbf{Input:} Number of iterations $N_{\text{iter}}$, number of warm up iterations $N_{w\text{arm}}$, learning rate $\gamma$, Teacher $F_T$, Student $F_S$, novel-domain augmenter $G$ and classifier $C$
\Statex \textbf{Data:} $N$ source domains $D_{\text{tr}} = \{D_1, D_2, ..., D_N\}$
\Statex \textbf{Output:} Teacher $F_T$
\For{$i$ in 1 : $N_{\text{iter}}$}
\State Randomly sample a minibatch $(x, y)$ from source domains ;
\If{$i < N_{\text{warm}}$}
\State Update $F_T$ and $C$ with $L_{\text{ce}}(C(F_T(x)), y)$;
\Else
\State Domain Generalized Representation Learning
\Statex $\tilde{x} = G(x)$;
\Statex $z = F_T(x)$, $\tilde{z} = F_S(\tilde{x})$;
\Statex Compute $L_{\text{dis}}^F$ (Eq.1) and $L_{\text{ce}}(C(\tilde{z}), y)$;
\Statex Update $F_S$ via back propagation. $\theta_S \leftarrow \theta_S - \gamma \frac{\partial (L_{\text{dis}}^F(z, \tilde{z}) + L_{\text{ce}}(C(\tilde{z}), y))}{\partial \theta_S}$ (Eq.2);
\Statex Update $F_T$ via EMA. $\theta_T \leftarrow \tau \theta_T + (1 - \tau) \theta_S$, where $\tau \in [0, 1]$ (Eq.3);
\State Novel Domain Augmentation
\Statex Compute $L_{\text{dis}}^G$ (Eq.4) and $L_{\text{ce}}(C(\tilde{z}), y)$;
\Statex Update $G$ via back propagation. $\theta_g \leftarrow \theta_g - \gamma \frac{\partial (-L_{\text{dis}}^G(z, \tilde{z}) + L_{\text{ce}}(C(\tilde{z}), y))}{\partial \theta_g}$ (Eq.5);
\EndIf
\EndFor
\end{algorithmic}
\end{algorithm}

B Further Quantitative Comparisons

For multiple source domain generalization, in addition to PACS, Office-Home, and DomainNet datasets included in main paper, we further evaluate the effectiveness of our approach on two benchmark datasets, Digits-DG and VLCS.

Table 1: Comparisons to existing methods on Digits-DG in leave-one-domain-out settings. **Bold** denotes the best result.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST</td>
<td>95.3</td>
<td>95.2</td>
<td>96.5</td>
<td>96.7</td>
<td>96.7</td>
<td>96.9</td>
<td>97.9 ± 0.1</td>
<td></td>
</tr>
<tr>
<td>MNIST-M</td>
<td>61.1</td>
<td>58.2</td>
<td>58.4</td>
<td>61.1</td>
<td>64.1</td>
<td>63.9</td>
<td>63.5</td>
<td></td>
</tr>
<tr>
<td>SVHN</td>
<td>62.3</td>
<td>65.5</td>
<td>65.0</td>
<td>65.3</td>
<td>68.6</td>
<td>68.6</td>
<td>64.7</td>
<td></td>
</tr>
<tr>
<td>SYN</td>
<td>79.5</td>
<td>79.1</td>
<td>78.4</td>
<td>80.2</td>
<td>81.0</td>
<td>83.2</td>
<td>81.2</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>74.5</td>
<td>74.5</td>
<td>74.6</td>
<td>75.8</td>
<td>77.6</td>
<td>78.1</td>
<td>76.5</td>
<td>78.4</td>
</tr>
</tbody>
</table>

Table 2: Comparisons to existing methods on VLCS using AlexNet in leave-one-domain-out settings. **Bold** denotes the best result.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PASCAL</td>
<td>66.3</td>
<td>67.3</td>
<td>67.4</td>
<td>70.6</td>
<td>69.1</td>
<td>70.3</td>
<td>69.8</td>
<td>73.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LabelMe</td>
<td>61.4</td>
<td>62.6</td>
<td>61.3</td>
<td>64.3</td>
<td>60.9</td>
<td>64.9</td>
<td>62.7</td>
<td>63.5</td>
<td>61.9</td>
<td>62.9 ± 0.7</td>
</tr>
<tr>
<td>Caltech</td>
<td>97.2</td>
<td>94.4</td>
<td>94.4</td>
<td>94.1</td>
<td>96.9</td>
<td>94.8</td>
<td>97.4</td>
<td>97.3</td>
<td>97.6</td>
<td>97.2 ± 0.1</td>
</tr>
<tr>
<td>Sun</td>
<td>68.1</td>
<td>64.4</td>
<td>64.4</td>
<td>65.9</td>
<td>64.3</td>
<td>67.6</td>
<td>67.9</td>
<td>68.0</td>
<td>68.2</td>
<td>69.6 ± 0.3</td>
</tr>
<tr>
<td>Average</td>
<td>73.3</td>
<td>72.3</td>
<td>72.3</td>
<td>72.9</td>
<td>73.2</td>
<td>74.1</td>
<td>74.5</td>
<td>74.7</td>
<td>75.4</td>
<td>76.1</td>
</tr>
</tbody>
</table>

B.1 Datasets

**Digits-DG** consists of four domains, MNIST [1], MNIST-M [2], SVHN [3] and SYN [2], with digit images of varying font styles and background colors. Each domain contains 10 categories, with 6000 images in total. Images are divided into the training split and the validation split at a ratio of 8:2.

**VLCS** [4] is a domain generalized visual classification benchmark, which includes five categories from four domains (PASCAL VOC 2007, LabelMe, Caltech, and Sun datasets), with the domain gap mainly from camera viewpoints, types of camera, or illumination conditions, etc. Images are divided into the training split and the validation split at a ratio of 9:1.

B.2 Implementation Details and Results

For Digits-DG, input images are resized to 32 × 32 pixels, and the backbone of \( F_T \) and \( F_S \) consists of four convolution layers, with the kernel size 3 and channel size 64. Each convolution layer is followed by a ReLU and a maxpooling layer with the kernel size 2. Classifier \( C \) is realized by a fully-connected layer, and maps a flattened feature vector to a 10 dimensional output. \( F_S \) is trained with SGD, initial learning rate of 0.05 and batch size of 128 for 60 epochs. For VLCS, input images are resized to 224 × 224 pixels, and we use AlexNet [5] pre-trained on ImageNet [6] as the backbone of our teacher and student networks. \( F_S \) is trained with the SGD optimizer, with an initial learning rate of 0.0005, and a batch size of 32 for 60 epochs. The learning rate is decayed by 0.1 after 30 epochs. For both Digits-DG and VLCS, \( F_T \) is updated via EMA with the momentum coefficient \( \tau \) of 0.999 by default. Our novel-domain augmenter \( G \) is realized by a fully convolutional network similar to the generator’s architecture in [7] and trained with the SGD optimizer.

Tables 1 and 2 show the quantitative comparisons with existing DG methods on Digit-DG and VLCS, respectively. Our approach still achieved satisfactory performance over the state-of-the-art models on all domains, with the reported highest average accuracy on both Digits-DG (78.4%) and VLCS (76.1%). The above experimental results further support the effectiveness and robustness of our method to tackle domain generalized visual classification tasks.

B.3 Generalization from A Single Source Domain

We conduct additional experiments with the ResNet-50 backbone on PACS using Art painting, Cartoon, and Sketch, respectively, as the single source domain to further confirm the use of our method to deal with such a challenging setting. As shown in Tables 3, 4, and 5, our approach performed favorably against the baseline (DeepAll) and the existing DG methods regardless of the source domain we selected. The above quantitative experiments thus confirmed the effectiveness and the domain generalization ability of our proposed model.
Table 3: Single source domain generalization on PACS using ResNet-50 as the backbone. Note that Art painting of PACS is selected as the single source domain for training.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Photo</th>
<th>Cartoon</th>
<th>Sketch</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepAll</td>
<td>96.9</td>
<td>57.0</td>
<td>42.8</td>
<td>65.6</td>
</tr>
<tr>
<td>JiGen [15]</td>
<td>96.3</td>
<td>61.4</td>
<td>52.7</td>
<td>70.1</td>
</tr>
<tr>
<td>CrossGrad [10]</td>
<td>97.3</td>
<td>62.5</td>
<td>45.9</td>
<td>68.6</td>
</tr>
<tr>
<td>DDAIG [7]</td>
<td>97.0</td>
<td>61.5</td>
<td>54.1</td>
<td>70.9</td>
</tr>
<tr>
<td>M-ADA [23]</td>
<td>97.2</td>
<td>63.7</td>
<td>47.0</td>
<td>69.3</td>
</tr>
<tr>
<td>Ours</td>
<td>97.4 ± 0.1</td>
<td>64.0 ± 0.4</td>
<td>56.1 ± 0.2</td>
<td>72.5</td>
</tr>
</tbody>
</table>

Table 4: Single source domain generalization on PACS using ResNet-50 as the backbone. Note that Cartoon of PACS is selected as the single source domain for training.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Photo</th>
<th>Art painting</th>
<th>Sketch</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepAll</td>
<td>87.0</td>
<td>64.0</td>
<td>55.8</td>
<td>68.9</td>
</tr>
<tr>
<td>JiGen [15]</td>
<td>87.1</td>
<td>65.3</td>
<td>66.3</td>
<td>72.9</td>
</tr>
<tr>
<td>CrossGrad [10]</td>
<td>86.8</td>
<td>66.4</td>
<td>65.4</td>
<td>72.9</td>
</tr>
<tr>
<td>DDAIG [7]</td>
<td>86.8</td>
<td>68.5</td>
<td>65.9</td>
<td>73.7</td>
</tr>
<tr>
<td>M-ADA [23]</td>
<td>87.7</td>
<td>67.7</td>
<td>63.1</td>
<td>72.8</td>
</tr>
<tr>
<td>Ours</td>
<td>87.6 ± 0.2</td>
<td>70.0 ± 0.1</td>
<td>68.4 ± 0.8</td>
<td>75.3</td>
</tr>
</tbody>
</table>

Table 5: Single source domain generalization on PACS using ResNet-50 as the backbone. Note that Sketch of PACS is selected as the single source domain for training.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Photo</th>
<th>Art painting</th>
<th>Cartoon</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepAll</td>
<td>25.1</td>
<td>21.0</td>
<td>43.7</td>
<td>29.9</td>
</tr>
<tr>
<td>JiGen [15]</td>
<td>37.5</td>
<td>37.1</td>
<td>55.6</td>
<td>43.4</td>
</tr>
<tr>
<td>CrossGrad [10]</td>
<td>31.6</td>
<td>25.5</td>
<td>49.1</td>
<td>35.4</td>
</tr>
<tr>
<td>DDAIG [7]</td>
<td>28.6</td>
<td>30.0</td>
<td>59.3</td>
<td>39.3</td>
</tr>
<tr>
<td>M-ADA [23]</td>
<td>26.0</td>
<td>23.1</td>
<td>52.0</td>
<td>33.7</td>
</tr>
<tr>
<td>Ours</td>
<td>39.2 ± 0.2</td>
<td>31.0 ± 0.3</td>
<td>61.0 ± 0.2</td>
<td>43.7</td>
</tr>
</tbody>
</table>

Figure 1: Impact of the epoch of warm-up stage on (a) PACS and (b) Office-Home using ResNet-50 as the backbone. Note that x and y axes denote the epoch of warm-up stage and top-1 classification accuracy (%), respectively.

Moreover, we provide additional comparisons of single-source domain generalization on Digit datasets. Following two very recent SOTAs of [20, 21] using MNIST [1] as the single source domain, our method showed promising results 61.7%, 83.2%, 69.3%, and 87.4% on SVHN [3], MNIST-M [2], SYN [2], and USPS [22], respectively. We achieved an average accuracy of 75.4% and outperformed [20] and [21] which resulted in 74.8% and 61.3%, respectively. And, as discussed in Section 2, PDEN [20] has a much larger memory requirement and a more complex model (i.e., more hyperparameters to select). The above additional experiments further confirm the effectiveness of our method for single-source domain generalization.
B.4 Parameter Analysis

Impact of the Warm-up Stage  In this sub-section, we conduct a detailed analysis of the impact of our warm-up stage on both PACS and Office-Home with ResNet-50 as the backbone. As shown in Fig. 1, the performance does not exhibit drastic fluctuations despite using different warm-up epochs, further showing that our model is stable and robust and that the training epoch of warm-up stage is not the most influential factor to the result. We eventually choose 10 epochs for warm-up training as default in our experiments.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 2. All DG methods share the same limitation which can generalize to unseen but similar target domains.
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code are proprietary. Nevertheless, we are happy to release it with the approval from our collaborator.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.2 and supplementary material
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See our Table 1-4 and 6
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.2

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4.2
   (b) Did you mention the license of the assets? [Yes] See Section 4.2
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] The code are proprietary. Nevertheless, we are happy to release it with the approval from our collaborator.
   (d) Did you discuss whether and how consent was obtained from people whose data you’re using/curating? [Yes] See Section 4.1. All datasets we used are public.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]
References


