Feature-Critic Networks for Heterogeneous Domain Generalisation

Yiying Li*, Yongxin Yang*, Wei Zhou, Timothy M. Hospedales

National University of Defense Technology, China
University of Edinburgh, UK
Samsung AI Centre, UK
Motivation

Domain Shift:
- Model performance degrades when deployed to a new target domain with different statistics to training.

To Ameliorate Domain Shift:
- Domain Adaptation
  - \( \{X_T\} \) or \( \{X_T, Y_T\} \) accessible during training
Motivation

Domain Shift:

➢ Model performance degrades when deployed to a new target domain with different statistics to training.

To Ameliorate Domain Shift:

➢ Domain Adaptation
  • $\{X_T\}$ or $\{X_T, Y_T\}$ accessible during training

➢ Domain Generalisation (Harder)
  • $\{X_T\}$ not accessible during training
  • Several Methods: Muandet ICML’13, Li ICCV’17, Balaji NeurIPS’18.
  • Common assumption: Shared Label Space (Homogeneous DG)
Heterogeneous DG is a Common Workflow

Heterogeneous DG:

- **Disjoint label space** in source + target → Feature generalisation.
- “ImageNet trained CNN as feature extractor”
Heterogeneous DG is a Common Workflow

Heterogeneous DG:

- Disjoint label space in source + target → Feature generalisation.
- “ImageNet trained CNN as feature extractor”

Source domains:

Train split of target domains:
- Extract features
- Train a SVM/KNN classifier

Test split of target domains:
- Evaluate performance
Heterogeneous DG is a Common Workflow

Heterogeneous DG:

- **Disjoint label space** in source + target → Feature generalisation.
- **“ImageNet trained CNN as feature extractor”**

**Source domains:**

- ImageNet CNN
- Hetero DG trained CNN
  
  **Fix the Feature Extractor**

**Train split of target domains:**

- Extract features
- Train a SVM/KNN classifier

**Test split of target domains:**

- Evaluate performance
Methodology: Key Idea

Loss Learning:
- Simulate domain-shift among a set of source domains.
- Meta-learn a loss function that promotes domain robustness.
Methodology: Key Idea

Loss Learning:

- Simulate domain-shift among a set of source domains.
- Meta-learn a loss function that promotes domain robustness.
- Loss function is defined on extracted features alone

Interpretation: Feature quality critic.
Introduce a learnable auxiliary loss $\ell_{\omega}^{\text{(Aux)}}$.

Conventional vs feature critic updates:

- $\theta^{\text{(OLD)}} = \theta - \alpha \nabla_{\theta} \ell^{\text{(CE)}}(D_{\text{meta-train}} | \theta)$
- $\theta^{\text{(NEW)}} = \theta - \alpha \nabla_{\theta} (\ell^{\text{(CE)}}(D_{\text{meta-train}} | \theta) + \ell_{\omega}^{\text{(Aux)}}(D_{\text{meta-train}} | \theta))$
Introduce a learnable auxiliary loss $\mathcal{L}_\omega^{(\text{Aux})}$

Conventional vs feature critic updates:

- $\theta^{(\text{OLD})} = \theta - \alpha \nabla_\theta \mathcal{L}^{(\text{CE})} (D_{\text{meta-train}} | \theta)$
- $\theta^{(\text{NEW})} = \theta - \alpha \nabla_\theta \left( \mathcal{L}^{(\text{CE})} (D_{\text{meta-train}} | \theta) + \mathcal{L}_\omega^{(\text{Aux})} (D_{\text{meta-train}} | \theta) \right)$

Meta-loss optimizes the resulting domain invariance

$$\min_\omega \tanh \left( \mathcal{L}^{(\text{CE})} (D_{\text{meta-test}} | \theta^{(\text{NEW})}) - \mathcal{L}^{(\text{CE})} (D_{\text{meta-test}} | \theta^{(\text{OLD})}) \right)$$
Algoirthm

- Introduce a learnable auxiliary loss $\ell_\omega (\text{Aux})$
- Conventional vs feature critic updates:
  - $\theta (\text{OLD}) = \theta - \alpha \nabla_{\theta} \ell (\text{CE}) (\mathcal{D}_{\text{meta-train}} | \theta)$
  - $\theta (\text{NEW}) = \theta - \alpha \nabla_{\theta} (\ell (\text{CE}) (\mathcal{D}_{\text{meta-train}} | \theta) + \ell_\omega (\text{Aux}) (\mathcal{D}_{\text{meta-train}} | \theta))$
- Meta-loss optimizes the resulting domain invariance
  $\min_\omega \tanh (\ell (\text{CE}) (\mathcal{D}_{\text{meta-test}} | \theta (\text{NEW})) - \ell (\text{CE}) (\mathcal{D}_{\text{meta-test}} | \theta (\text{OLD})))$
- Auxiliary loss design:
  $\ell_\omega (\text{Aux}) := \text{mean(softplus}(h_\omega (f_\theta (x_i))))}$
Results

Heterogeneous DG: Visual Decathlon - ResNet18
## Results

### Heterogeneous DG: Visual Decathlon - ResNet18

**Table 1.** Recognition accuracy (%) and VD scores on four held out target datasets in Visual Decathlon using ResNet-18 extractor.

<table>
<thead>
<tr>
<th>Target</th>
<th>SVM Classifier</th>
<th>KNN Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Im.N. PT</td>
<td>CrossGrad</td>
</tr>
<tr>
<td>D. Textures</td>
<td>41.70</td>
<td>36.54</td>
</tr>
<tr>
<td>VGG-Flowers</td>
<td>51.57</td>
<td>57.84</td>
</tr>
<tr>
<td>UCF101</td>
<td>44.93</td>
<td>45.80</td>
</tr>
<tr>
<td>Ave.</td>
<td>38.71</td>
<td>40.03</td>
</tr>
<tr>
<td>VD-Score</td>
<td>308</td>
<td>280</td>
</tr>
</tbody>
</table>

*ImageNet 38.7% → Combined Domains 40.3% → Feature Critic 42.3%.***
Results

Table 1. VD recognition accuracy differences (%) against AGG with different proportions of training data available.

<table>
<thead>
<tr>
<th>Data</th>
<th>SVM Classifier</th>
<th>KNN Classifier</th>
<th>AGG Im.N.</th>
<th>PT CrossGrad</th>
<th>MR</th>
<th>MR-FL</th>
<th>Reptile</th>
<th>FC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>18.27 ± 1.74</td>
<td>13.93 ± 1.15</td>
<td>-2.38</td>
<td>-1.50</td>
<td>+1.83</td>
<td>+1.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25%</td>
<td>30.10 ± 5.14</td>
<td>23.80 ± 3.62</td>
<td>-8.44</td>
<td>-2.48</td>
<td>-2.13</td>
<td>+0.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>34.63 ± 3.55</td>
<td>30.19 ± 5.02</td>
<td>-7.04</td>
<td>-0.07</td>
<td>+2.12</td>
<td>+2.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>40.27 ± 1.56</td>
<td>32.52 ± 0.69</td>
<td>-6.25</td>
<td>-1.52</td>
<td>+2.02</td>
<td>+2.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ave.</td>
<td>30.82 ± 2.13</td>
<td>25.11 ± 2.05</td>
<td>-6.03</td>
<td>-1.39</td>
<td>+1.58</td>
<td>+1.58</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2. VD recognition accuracy differences (%) against AGG with different proportions of training data available.

<table>
<thead>
<tr>
<th>Data</th>
<th>SVM Classifier</th>
<th>KNN Classifier</th>
<th>AGG Im.N.</th>
<th>PT CrossGrad</th>
<th>MR</th>
<th>MR-FL</th>
<th>Reptile</th>
<th>FC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>18.27 ± 1.74</td>
<td>13.93 ± 1.15</td>
<td>-2.38</td>
<td>-1.50</td>
<td>+1.83</td>
<td>+1.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25%</td>
<td>30.10 ± 5.14</td>
<td>23.80 ± 3.62</td>
<td>-8.44</td>
<td>-2.48</td>
<td>-2.13</td>
<td>+0.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>34.63 ± 3.55</td>
<td>30.19 ± 5.02</td>
<td>-7.04</td>
<td>-0.07</td>
<td>+2.12</td>
<td>+2.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>40.27 ± 1.56</td>
<td>32.52 ± 0.69</td>
<td>-6.25</td>
<td>-1.52</td>
<td>+2.02</td>
<td>+2.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ave.</td>
<td>30.82 ± 2.13</td>
<td>25.11 ± 2.05</td>
<td>-6.03</td>
<td>-1.39</td>
<td>+1.58</td>
<td>+1.58</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3. VD recognition accuracy differences (%) against AGG with different proportions of training data available.

<table>
<thead>
<tr>
<th>Data</th>
<th>SVM Classifier</th>
<th>KNN Classifier</th>
<th>AGG Im.N.</th>
<th>PT CrossGrad</th>
<th>MR</th>
<th>MR-FL</th>
<th>Reptile</th>
<th>FC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>18.27 ± 1.74</td>
<td>13.93 ± 1.15</td>
<td>-2.38</td>
<td>-1.50</td>
<td>+1.83</td>
<td>+1.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25%</td>
<td>30.10 ± 5.14</td>
<td>23.80 ± 3.62</td>
<td>-8.44</td>
<td>-2.48</td>
<td>-2.13</td>
<td>+0.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50%</td>
<td>34.63 ± 3.55</td>
<td>30.19 ± 5.02</td>
<td>-7.04</td>
<td>-0.07</td>
<td>+2.12</td>
<td>+2.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100%</td>
<td>40.27 ± 1.56</td>
<td>32.52 ± 0.69</td>
<td>-6.25</td>
<td>-1.52</td>
<td>+2.02</td>
<td>+2.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ave.</td>
<td>30.82 ± 2.13</td>
<td>25.11 ± 2.05</td>
<td>-6.03</td>
<td>-1.39</td>
<td>+1.58</td>
<td>+1.58</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Recognition accuracy (%) averaged over 10 train+test runs on Rotated MNIST.

<table>
<thead>
<tr>
<th>Target</th>
<th>CrossGrad</th>
<th>MetaReg</th>
<th>Reptile</th>
<th>AGG</th>
<th>Feature-Critic-MLP</th>
<th>Feature-Critic-Flatten</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>86.03 ± 0.69</td>
<td>85.70 ± 0.31</td>
<td>87.78 ± 0.30</td>
<td>86.42 ± 0.24</td>
<td><strong>89.23 ± 0.25</strong></td>
<td>87.04 ± 0.31</td>
</tr>
<tr>
<td>M15</td>
<td>98.92 ± 0.53</td>
<td>98.87 ± 0.41</td>
<td>99.44 ± 0.22</td>
<td>98.61 ± 0.27</td>
<td><strong>99.68 ± 0.24</strong></td>
<td>99.53 ± 0.27</td>
</tr>
<tr>
<td>M30</td>
<td>98.60 ± 0.51</td>
<td>98.32 ± 0.44</td>
<td>98.42 ± 0.24</td>
<td>99.19 ± 0.19</td>
<td>99.20 ± 0.20</td>
<td><strong>99.41 ± 0.18</strong></td>
</tr>
<tr>
<td>M45</td>
<td>98.39 ± 0.29</td>
<td>98.58 ± 0.28</td>
<td>98.80 ± 0.20</td>
<td>98.22 ± 0.24</td>
<td>99.24 ± 0.18</td>
<td><strong>99.52 ± 0.24</strong></td>
</tr>
<tr>
<td>M60</td>
<td>98.68 ± 0.28</td>
<td>98.93 ± 0.32</td>
<td>99.03 ± 0.28</td>
<td>99.48 ± 0.19</td>
<td><strong>99.53 ± 0.23</strong></td>
<td>99.23 ± 0.16</td>
</tr>
<tr>
<td>M75</td>
<td>88.94 ± 0.47</td>
<td>89.44 ± 0.37</td>
<td>87.42 ± 0.33</td>
<td>88.92 ± 0.43</td>
<td>91.44 ± 0.34</td>
<td><strong>91.52 ± 0.26</strong></td>
</tr>
<tr>
<td>Ave.</td>
<td>94.93</td>
<td>94.97</td>
<td>95.15</td>
<td>95.14</td>
<td><strong>96.39</strong></td>
<td>96.04</td>
</tr>
</tbody>
</table>

Cross-domain feature encoding quality (PCA):

![Baseline](image1.png) ![Feature-Critic](image2.png)
Thanks for Listening!

• Please see our poster: Pacific Ballroom #77

• Code: https://github.com/liyiying/Feature_Critic