1 Comparison without Post-Processing

Since some brand-new methods utilize post-processing, such as calibration stacking [5] or domain detector [2, 12], to alleviate the domain shift problem, we report the results of our Dual Progressive Prototype Network (DPPN) with post-processing in Table 3 of the main paper for fair comparison. In this part, we further compare our DPPN with recent methods that clearly report their results without post-processing, of which the comparison results are shown in Table 1. APN [15] only reports their results with calibration stacking. Our DPPN outperforms the best one by respectively 15.3%, 8.8%, and 7.3% for $H$ on CUB, AWA2, and aPY datasets, and obtains comparable performance on SUN dataset. This demonstrates the effectiveness of learning representations that progressively explore category discrimination and attribute-region correspondence.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CUB $MCA_u$</th>
<th>CUB $MCA_s$</th>
<th>AWA2 $H$</th>
<th>aPY $MCA_u$</th>
<th>aPY $MCA_s$</th>
<th>aPY $H$</th>
<th>SUN $MCA_u$</th>
<th>SUN $MCA_s$</th>
<th>SUN $H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SJE [1]</td>
<td>23.5</td>
<td>59.2</td>
<td>33.6</td>
<td>8.0</td>
<td>73.9</td>
<td>14.4</td>
<td>3.7</td>
<td>55.7</td>
<td>6.9</td>
</tr>
<tr>
<td>DEVISE [7]</td>
<td>23.8</td>
<td>53.0</td>
<td>32.8</td>
<td>17.1</td>
<td>74.7</td>
<td>27.8</td>
<td>4.9</td>
<td>76.9</td>
<td>9.2</td>
</tr>
<tr>
<td>SYNC [3]</td>
<td>11.5</td>
<td>70.9</td>
<td>19.8</td>
<td>10.0</td>
<td>90.5</td>
<td>18.0</td>
<td>7.4</td>
<td>66.3</td>
<td>13.3</td>
</tr>
<tr>
<td>MLSE [6]</td>
<td>22.3</td>
<td>71.6</td>
<td>34.0</td>
<td>23.8</td>
<td>83.2</td>
<td>37.0</td>
<td>12.7</td>
<td>74.3</td>
<td>21.7</td>
</tr>
<tr>
<td>LFGAA [10]</td>
<td>36.2</td>
<td>80.9</td>
<td>50.0</td>
<td>27.0</td>
<td>93.4</td>
<td>41.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>AREN [13]</td>
<td>38.9</td>
<td>78.7</td>
<td>52.1</td>
<td>15.6</td>
<td>92.9</td>
<td>26.7</td>
<td>9.2</td>
<td>76.9</td>
<td>16.4</td>
</tr>
<tr>
<td>DAZLE [8]</td>
<td>42.0</td>
<td>65.3</td>
<td>51.1</td>
<td>25.7</td>
<td>82.5</td>
<td>39.2</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DPPN</td>
<td>56.8</td>
<td>82.8</td>
<td>67.4</td>
<td>34.7</td>
<td>94.1</td>
<td>50.7</td>
<td>17.7</td>
<td>80.8</td>
<td>29.0</td>
</tr>
</tbody>
</table>

Table 1: Results of GZSL on four classification benchmarks under no post-processing settings. The best result is bolded.

2 Conventional ZSL Results

We provide the comparison results with several brand-new methods that are most related to DPPN under CZSL setting in Table 2. $MCA_u$ is used as evaluation metric. DPPN obtains comparable, even better performance, compared to SOTA methods, which demonstrates the transferability of our method. Notably, the performance of DPPN on CZSL is not as impressive as in GZSL. The reason is that DPPN focuses on improving representation discrimination to alleviate the domain bias problem.
Table 2: Results of CZSL. $MCA_u$ is used as the evaluation metric. The best result is bolded, and the second best is underlined.

### 3 Comparison on Evaluation Metric AUSUC

In this section, we give the comparison results with recent related methods on the evaluation metric of AUSUC [4]. As shown in Table 3, our DPPN outperforms the best one by respectively 19.7% and 7.7% for AUSUC on CUB and AWA2 datasets, and obtains comparable performance on SUN dataset. The robust improvements over various metrics prove that DPPN can effectively alleviate the domain bias problem in GZSL.

Table 3: Comparison results on AUSUC metric. The best result is bolded.

### 4 Results of Input Size 224 × 224

For fair comparison, we use the setting of input size 448 × 448 following recent SOTA methods, e.g., VSE-S [18], GEM-ZSL [11], and AREN [13]. In this part, we conduct experiments comparing DPPN with recent methods under the setting of input size 224 × 224, of which the results are shown in Table 4. From the results, our DPPN outperforms the best previous method by respectively 3.8%, 6.7%, and 2.9% on CUB, aPY, and SUN datasets, and achieves the second-best performance on AWA2. The SOTA performance on different resolutions demonstrates the effectiveness and generalization of our DPPN.

Table 4: Results of GZSL on four classification benchmarks using input size 224 × 224. The best result is bolded, and the second best is underlined.

### 5 Results of w/o Finetuning

We adopt a two-step training schedule that first trains DPPN with the fixed ResNet-101 backbone and then fine-tunes the whole network. As shown in Table 5, we also provide the results without finetuning. Our DPPN outperforms the best previous method by respectively 5.0%, 5.1%, 6.5%, and 0.9% for $H$ on CUB, AWA2, aPY, and SUN datasets. The results show that our DPPN is superior to previous methods even without finetuning.
6 Different Combinations of Representations in Inference

Only the visual representation \( f^K(X) \) at the \( K \)-th iteration is used in inference for simplicity. In this section, we conduct experiments of using different combinations of representations as the final image representation in inference, of which the results are shown in Table 6. From the table, we can see that using only the \( K \)-th representation obtains comparable results with summing up all the representations. Since the representation derives from the preceding representation, the preceding representations bring limited supplement to the final performance. Besides, concatenating all the representations results in a final representation with a large dim, which is harder to train and obtains worse performance. Moreover, using \( f^1(X) \) as the final image representation can obtain considerable performance. However, our DPPN learns more transferable and distinctive representations through updating, which will achieve better results if using the final representation in inference.

7 Visualization of Representations

To intuitively demonstrate the distinction of our visual representations, we compare the representations of DPPN and Base-V2S using t-SNE on randomly selected categories, of which the results are shown in Fig. 1. Compared to Base-V2S, the representations are more compact inside a category and the margins between seen and unseen categories are enlarged in DPPN. This reflects that DPPN obtains more distinctive and transferable representations through progressively learning attribute-region correspondence and category discrimination.

8 Broader Impact

This paper proposes a novel method DPPN to tackle the domain shift problem in Generalized Zero-Shot Learning. The positive impacts of this paper can be summarized in two-fold: a) The proposed method enables people to recognize novel categories without elaborate labeling, which can reduce
annotation costs and difficulties. b) Our proposed DPPN brings negligible additional computation over the feature extractor network, which can save computing resources. The potential negative societal impacts contain: a) Our DPPN can alleviate the burden of labeling data, which may cause unemployment of data annotators. b) Our DPPN may be not suitable for data with large #categories and small #images in each category. c) The recognition of unseen domain remains still not good enough for industrial use, which should be further studied.

References


