Decomposed Soft Prompt Guided Fusion Enhancing for Compositional Zero-Shot Learning – Supplementary Materials

The supplementary file provides more details for paper "Decomposed Soft Prompt Guided Fusion Enhancing for Compositional Zero-Shot Learning", including the following aspects:

2. Backbone Study.
3. Hyper-Parameters Analysis.
4. Pseudocode.
5. Qualitative Results.

1. Summary Statistics of Datasets

We analyse three datasets included MIT-States [2], UT-Zappos [5] and C-GQA [4] statistically and the summary statistics can be seen in Tab. 1. s and o denote the number of state and object concepts, and i represents the number of images. Also, c_s and c_u are the pair concepts of seen and unseen classes.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>s</th>
<th>o</th>
<th>Train</th>
<th>Validation</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIT-States</td>
<td>115</td>
<td>245</td>
<td>1262</td>
<td>30338</td>
<td>400</td>
</tr>
<tr>
<td>UT-Zappos</td>
<td>16</td>
<td>12</td>
<td>83</td>
<td>22998</td>
<td>15</td>
</tr>
<tr>
<td>CGQA</td>
<td>453</td>
<td>870</td>
<td>6963</td>
<td>26920</td>
<td>1022</td>
</tr>
</tbody>
</table>

2. Backbone Study

To evaluate the performance of DFSP with various backbone, we also retrain the model with same parameters (α = 0.01, β = 0.1 and K = 1) and only replace the backbone with ViT-B/32, Vit-B/16 and ViT-L/14@336px due to the limitation of image encoder in DFSP which is based on transformer networks [1]. Metrics contists of S, U, H and AUC and the model (DFSP(t2i)) are tested on MIT-States and UT-Zappos with the settings of Closed-World (CW) and Open-World (OW). Meanwhile, we evaluate DFSP with 5 random seeds to report the standard error, which can be seen in Tab. 2.

From the results in Tab. 2, we can see that ViT-14/L and ViT-L/14@336px achieve state-of-the-art (SOTA) results both on Closed-World and Open-World. Meanwhile, all backbones work well on multiple metrics, especially on MIT-States, such as AUC %12.8 versus %5.5 on SCEN [3] with the setting of Closed-World.

3. Hyper-Parameters Analysis

The loss function of DFSP is \( L = L_{dfm} + \alpha L_{st+obj} + \beta L_{spm} \), which contains two hyper-parameters \( \alpha \) and \( \beta \). \( L_{dfm} \) is the final pair loss in DFM, \( L_{st+obj} \) is the decomposed features pair loss and \( L_{spm} \) is the pair loss before fusion in SPM. To evaluate the influence of them for DFSP(t2i), we show the hyper-parameters analysis in this section on MIT-States and UT-Zappos with the settings of Closed-World and Open-World. \( \alpha \) and \( \beta \) are set to seven different orders of magnitude.
Table 2. DFSP with different backbone results on MIT-States and UT-Zappos with the settings of Closed-World (CW) and Open-World (OW). We also report the standard error with 5 random seeds.

<table>
<thead>
<tr>
<th>Backbone</th>
<th>MIT-States</th>
<th>UT-Zappos</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U</td>
<td>H</td>
</tr>
<tr>
<td>ViT-B/32</td>
<td>36.7±0.25</td>
<td>29.4±0.18</td>
</tr>
<tr>
<td>ViT-B/16</td>
<td>39.6±0.17</td>
<td>31.5±0.18</td>
</tr>
<tr>
<td>ViT-L/14</td>
<td>46.8±0.54</td>
<td>37.4±0.35</td>
</tr>
<tr>
<td>ViT-L/14@336px</td>
<td>45.6±0.21</td>
<td>36.9±0.57</td>
</tr>
</tbody>
</table>

Additionally, the recompose function is reverse to decompose function and the code is shown as Code 1.

4. Pseudocode

We illustrate the core pseudocode in this section, including the decompose and recompose function. Given the pair_idx, we can obtain the att_idx and obj_idx, which could be utilized to decompose the state feature text_att and object feature text_obj. Additionally, the recompose function is reverse to decompose function and the code is shown as Code 1.
def decompose(text_feature, pair_idx):
    t, l, c = text_feature.shape
    att_idx, obj_idx = pair_idx[:, 0].cpu().numpy(), pair_idx[:, 1].cpu().numpy()
    text_att = torch.zeros(t, self.attributes, c).cuda()
    text_obj = torch.zeros(t, self.classes, c).cuda()
    for i in range(self.attributes):
        text_att[:, i, :] = text_feature[:, np.where(att_idx == i)[0], :].mean(-2)
    for i in range(self.classes):
        text_obj[:, i, :] = text_feature[:, np.where(obj_idx == i)[0], :].mean(-2)
    text_feature_plus = torch.cat([text_att, text_obj], dim=1)
    return text_feature_plus

def recompose(text_feature_plus, pair_idx):
    t, l, c = text_feature.shape
    att_idx, obj_idx = pair_idx[:, 0].cpu().numpy(), pair_idx[:, 1].cpu().numpy()
    text_com_feature = torch.zeros(t, len(idx), c).cuda()
    text_com_feature = text_feature[:, att_idx, :] * text_feature[:, obj_idx + offset, :]
    return text_feature

5. Qualitative Results

We report the top-1 qualitative results in the body of the paper. To better prove the effectiveness of DFSP, we show the top-3 qualitative results on MIT-States, UT-Zappos and CGQA in this section, which can be seen in Fig. 3. From the prediction results of top-3, it can be seen that even if top-1 has no successful cases, most of top-3 results can be predicted correctly. The compositions that the model has not seen can still be predicted correctly, which proves the generalization ability of the model to unseen concepts. Due to the abstract nature of state, it is more difficult to identify a state than an object, which can be seen more failure cases in Fig. 3 for states.

References

Figure 3. Qualitative top-3 results on MIT-States [2], UT-Zappos [5] and C-GQA [4]. blue denotes the wrong prediction and green represents the right case. The three columns on the left are success cases, and the two columns on the right are wrong cases for the top-1 prediction.