A. Complete Table of Inter- and Intra-class Generalizability

Due to the differences in dataset difficulty, we established criteria and filtered categories for presentation in the main manuscript. We display the complete result in Figure 1 to show the inter- and intra-class generalizability of InMeMo’s performance in mIoU on Psacal-5i dataset [3].

According to the column (class-level generalizability of test set in $S_{\omega}'$), we found bus and sheep are general classes. The learnable prompt, trained on any classes, performed well on the test classes of bus (56.87 ± 3.57) and sheep (49.01 ± 8.07) in visual ICL. Regarding other classes, we found that some classes are difficult tasks for visual ICL, leading to poor performance, such as bicycle (12.35 ± 2.26), bottle (25.27 ± 3.36), chair (11.21 ± 2.30), diningtable (20.81 ± 5.28), pottedplant (15.82 ± 2.13), sofa (28.71 ± 4.24), and tvmonitor (20.90 ± 4.68). The person is the least generalizable class, where the learnable prompt trained on person is most effective on the test set of person, while that trained on other classes perform poorly.

The mean mIoU score for all 20 class pairs is 34.32%, which suffers from a significant drop from InMeMo’s mean score over all folds. This indicates the need to adjust the learnable prompt for target tasks.

B. More Results of Domain Shift Analysis

To assess the robustness of incorporating the learnable prompt at various variants regarding the domain shift issues, we present a comprehensive comparison in Table 1. This analysis was carried out on the COCO-5i dataset [5], with identical settings mentioned in the main manuscript.

Overall, the addition of the learnable prompt at each variant is robust. Specifically, the results for the drop score of Means on different variants are as follows: baseline (2.42), $I$ (2.74), $Q$ (2.51), $I$ & $Q$ (2.54), $I$ & $L$ (i.e. InMeMo, 3.11), $I$, $L$ & $Q$ (1.5).

We observed that the rankings for performance on Means did not change. On the relatively easy splits (based on baseline) on Fold-0 and Fold-1, $I$ achieved the best performance of all variants (40.55% & 44.53%). On the more hard splits Fold-2 and Fold-3, InMeMo performed better (40.45% & 37.12%). The $I$ meets a relatively minor drop compared to InMeMo, which means giving the learnable prompt to in-context label images can improve the overall performance but sacrifice the robustness of domain shift issues. In variant $I$, $L$ & $Q$, it exhibits a minimal decrease in domain shift issue, making it the most robust among all variants. Consequently, compared to the baseline, all variants demonstrate robustness to domain shift issues. It would be an intriguing research direction to explore the development of learnable prompts that are more robust for visual ICL.

C. The Padding Size of Prompt Enhancer $t_\phi$

We configured the padding size of InMeMo to 30 followed [1], but were curious about the impact of various padding sizes on InMeMo’s performance. Consequently, we conducted a thorough analysis to assess the effects of altering the padding size as shown in Table 2.

We found that our InMeMo performed the best regarding Mean in all padding size settings. Increasing the padding size from 10 to 30 resulted in improved performance for InMeMo. In contrast, when we increase the padding size from 30 to 60, we found a decrease in InMeMo’s performance.

When the padding size is set to 20, sub-optimal results are obtained on the Mean, while the best performance is achieved on Fold-1 and Fold-2 (48.11% and 42.68%, respectively). The performances of InMeMo at settings 20 and 30 achieved similar results. However, increasing the padding size from 30 to 40 led to a significant decrease in performance (from 43.14% to 26.90%). Conversely, there was only a relatively slight decrease from 40 to 60. We claim that setting the padding size to 40 causes the learnable prompt to cover much more original information or distribution of the in-context pair, resulting in a considerable loss.
of performance. Therefore, a padding size of 30 is optimal for InMeMo. Larger padding sizes with additional parameters would result in inferior performance in our proposed InMeMo model.

D. More Visual Examples

In this section, we illustrate more visual examples from different folds. The main paper only presents a few spe-
Table 1. The results of domain shift analysis on all the variants of InMeMo. *Pascal → Pascal* means in-context pairs and query images both source from PASCAL, *COCO → Pascal* indicates that in-context pairs are from COCO-5 and query images are from Pascal-5. The baseline scores are our reproduction of pixel-level retrieval in [4]. The best results in each column in *COCO → Pascal* are represented in bold.

<table>
<thead>
<tr>
<th>Combination</th>
<th>Pascal → Pascal</th>
<th>COCO → Pascal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>35.69</td>
<td>33.83</td>
</tr>
<tr>
<td>I</td>
<td>42.57</td>
<td>40.55</td>
</tr>
<tr>
<td>Q</td>
<td>39.56</td>
<td>37.70</td>
</tr>
<tr>
<td>I &amp; Q</td>
<td>38.31</td>
<td>35.74</td>
</tr>
<tr>
<td>I &amp; L (InMeMo)</td>
<td>41.65</td>
<td>38.74</td>
</tr>
<tr>
<td>I, L &amp; Q</td>
<td>39.84</td>
<td>38.67</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Combination</th>
<th>Fold-0</th>
<th>Fold-1</th>
<th>Fold-2</th>
<th>Fold-3</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>35.69</td>
<td>38.25</td>
<td>35.86</td>
<td>33.37</td>
<td>35.79</td>
</tr>
<tr>
<td>I</td>
<td>42.57</td>
<td>47.08</td>
<td>41.60</td>
<td>39.44</td>
<td>42.67</td>
</tr>
<tr>
<td>Q</td>
<td>39.56</td>
<td>44.57</td>
<td>41.40</td>
<td>38.06</td>
<td>40.90</td>
</tr>
<tr>
<td>I &amp; Q</td>
<td>38.31</td>
<td>44.37</td>
<td>39.98</td>
<td>37.80</td>
<td>40.12</td>
</tr>
<tr>
<td>I &amp; L (InMeMo)</td>
<td>41.65</td>
<td>47.68</td>
<td>42.43</td>
<td>40.80</td>
<td>43.14</td>
</tr>
<tr>
<td>I, L &amp; Q</td>
<td>39.84</td>
<td>43.49</td>
<td>35.58</td>
<td>27.39</td>
<td>36.58</td>
</tr>
</tbody>
</table>

Table 2. The mIoU scores for varying padding sizes of prompt enhancer $\phi$. The Para. represents the number of tunable parameters. The highest score in each fold is marked in bold.

<table>
<thead>
<tr>
<th>Padding size</th>
<th>Para.</th>
<th>Fold-0</th>
<th>Fold-1</th>
<th>Fold-2</th>
<th>Fold-3</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>25,680</td>
<td>40.19</td>
<td>46.16</td>
<td>40.87</td>
<td>40.06</td>
<td>41.82</td>
</tr>
<tr>
<td>20</td>
<td>48,960</td>
<td>40.82</td>
<td>48.11</td>
<td>42.68</td>
<td>39.12</td>
<td>42.68</td>
</tr>
<tr>
<td>30 (InMeMo)</td>
<td>69,840</td>
<td>41.65</td>
<td>47.68</td>
<td>42.43</td>
<td>40.80</td>
<td>43.14</td>
</tr>
<tr>
<td>40</td>
<td>88,320</td>
<td>24.12</td>
<td>29.77</td>
<td>27.60</td>
<td>26.09</td>
<td>26.90</td>
</tr>
<tr>
<td>50</td>
<td>104,400</td>
<td>20.55</td>
<td>28.83</td>
<td>24.04</td>
<td>27.34</td>
<td>25.19</td>
</tr>
<tr>
<td>60</td>
<td>118,080</td>
<td>18.73</td>
<td>28.50</td>
<td>24.78</td>
<td>27.51</td>
<td>24.88</td>
</tr>
</tbody>
</table>

Specific examples, so we want to provide more accessible and intuitive examples of our InMeMo.

**D.1. Foreground segmentation**

In the foreground segmentation task, we provide visual examples of the baseline, prompt-Self [4], InMeMo, and the ground-truth label (GT). The Fold-0 (Figure 2), Fold-1 (Figure 3), Fold-2 (Figure 4), and Fold-3 (Figure 5) are represented in this section. We have arranged the format based on the different categories (columns) and demonstrated 20 examples for each fold.

**D.2. Single object detection**

We also illustrated more examples of the single object detection in Figure 6.

As mentioned in the main manuscript, InMeMo generates highly accurate details that align with ground-truth label images. Moreover, InMeMo exhibits robustness against variations in similarity, including color, size, viewpoints, and poses, between in-context and query images. Conversely, InMeMo’s performance is comparable to the baseline and prompt-Self when the similarity between the in-context and query images is high.

We have observed that InMeMo performs significantly better than previous works in single object detection tasks. This is due to our adherence to the setting proposed by [2], wherein we eliminate general samples whose bounding box occupies more than 50% of the entire image, to retain non-trivial samples. A more challenging scenario involves retaining only the samples in the test set where a single object occupies less than 20% of the entire image. Therefore, such tough setting demonstrates that our InMeMo performs well on fine-grained datasets. Specifically, the in-context pairs do not adequately instruct large-scale vision models.
Figure 2. The visual examples in Fold-0. Each column represents a different class. We arrange them in the order of in-context pair, query image, baseline, prompt-Self, InMeMo, and the ground-truth label (GT). We show all five classes in this fold: aeroplane, bicycle, bird, boat, bottle.
Figure 3. The visual examples in Fold-1: bus, car, cat, chair, cow.
Figure 4. The visual examples in Fold-2: diningtable, dog, horse, motorbike, person.
Figure 5. The visual examples in Fold-3: pottedplant, sheep, sofa, train, tvmonitor.
Figure 6. The visual examples in the single object detection task.
References


