Instruct Me More! Random Prompting for Visual In-Context Learning: Supplementary Material

Jiahao Zhang[1], Bowen Wang[2], Liangzhi Li[2], Yuta Nakashima[2], Hajime Nagahara[2]
Osaka University, Japan
[1]jiahao@is.ids.osaka-u.ac.jp
[2]{wang, li, n-yuta, nagahara}@ids.osaka-u.ac.jp

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-5^i dataset [3].

According to the column (class-level generalizability of test set in $\mathcal{S}_{\omega'}$), 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 $\pm$ 3.57) and `sheep` (49.01 $\pm$ 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 $\pm$ 2.26), `bottle` (25.27 $\pm$ 3.36), `chair` (11.21 $\pm$ 2.30), `diningtable` (20.81 $\pm$ 5.28), `pottedplant` (15.82 $\pm$ 2.13), `sofa` (28.71 $\pm$ 4.24), and `tvmonitor` (20.90 $\pm$ 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-5^i 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_ϕ

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

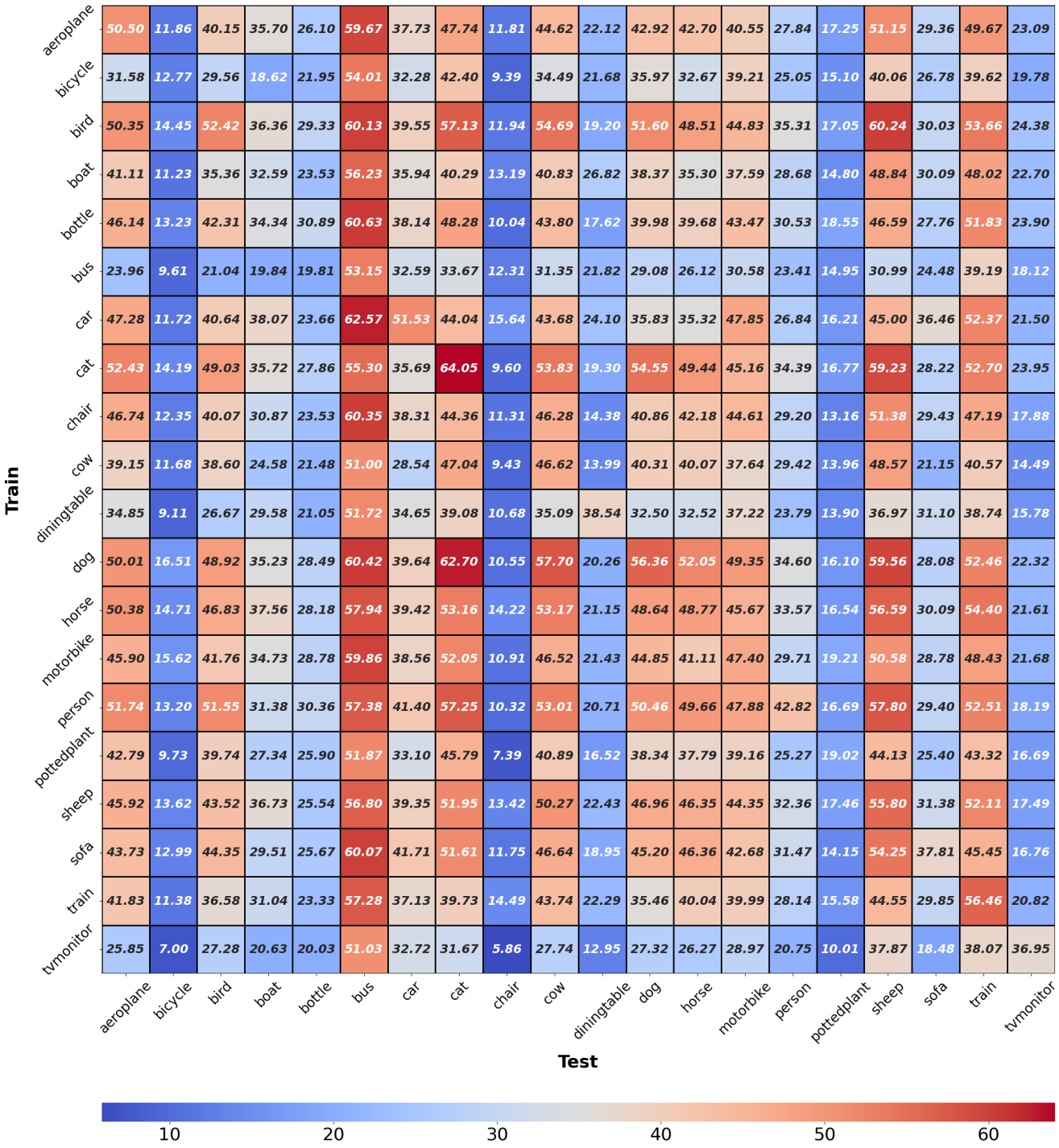

Figure 1. The complete Inter- and intra-class generalization performance in mIoU. The horizontal and vertical axes are the classes used for prediction and training, respectively. The diagonal elements show intra-class performance.

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* $\rightarrow$ *Pascal* means in-context pairs and query images both source from PASCAL. *COCO* $\rightarrow$ *Pascal* indicates that in-context pairs are from COCO-5^i and query images are from Pascal-5^i. The baseline scores are our reproduction of pixel-level retrieval in [4]. The best results in each column in *COCO* $\rightarrow$ *Pascal* are represented in **bold**.

	Combination	Fold-0	Fold-1	Fold-2	Fold-3	Means
Pascal $\rightarrow$ *Pascal*	Baseline	35.69	38.25	35.86	33.37	35.79
	I	42.57	47.08	41.60	39.44	42.67
	Q	39.56	44.57	41.40	38.06	40.90
	I & Q	38.31	44.37	39.98	37.80	40.12
	I & L (InMeMo)	41.65	47.68	42.43	40.80	43.14
	I, L & Q	39.84	43.49	35.58	27.39	36.58
COCO $\rightarrow$ *Pascal*	Baseline	33.83	36.11	32.89	30.64	33.37
	I	**40.55**	**44.53**	38.62	36.01	39.93
	Q	37.26	42.40	39.33	34.56	38.39
	I & Q	35.70	41.96	37.58	35.06	37.58
	I & L (InMeMo)	38.74	43.82	**40.45**	**37.12**	**40.03**
	I, L & Q	38.67	41.86	33.33	26.44	35.08

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

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

cific 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.

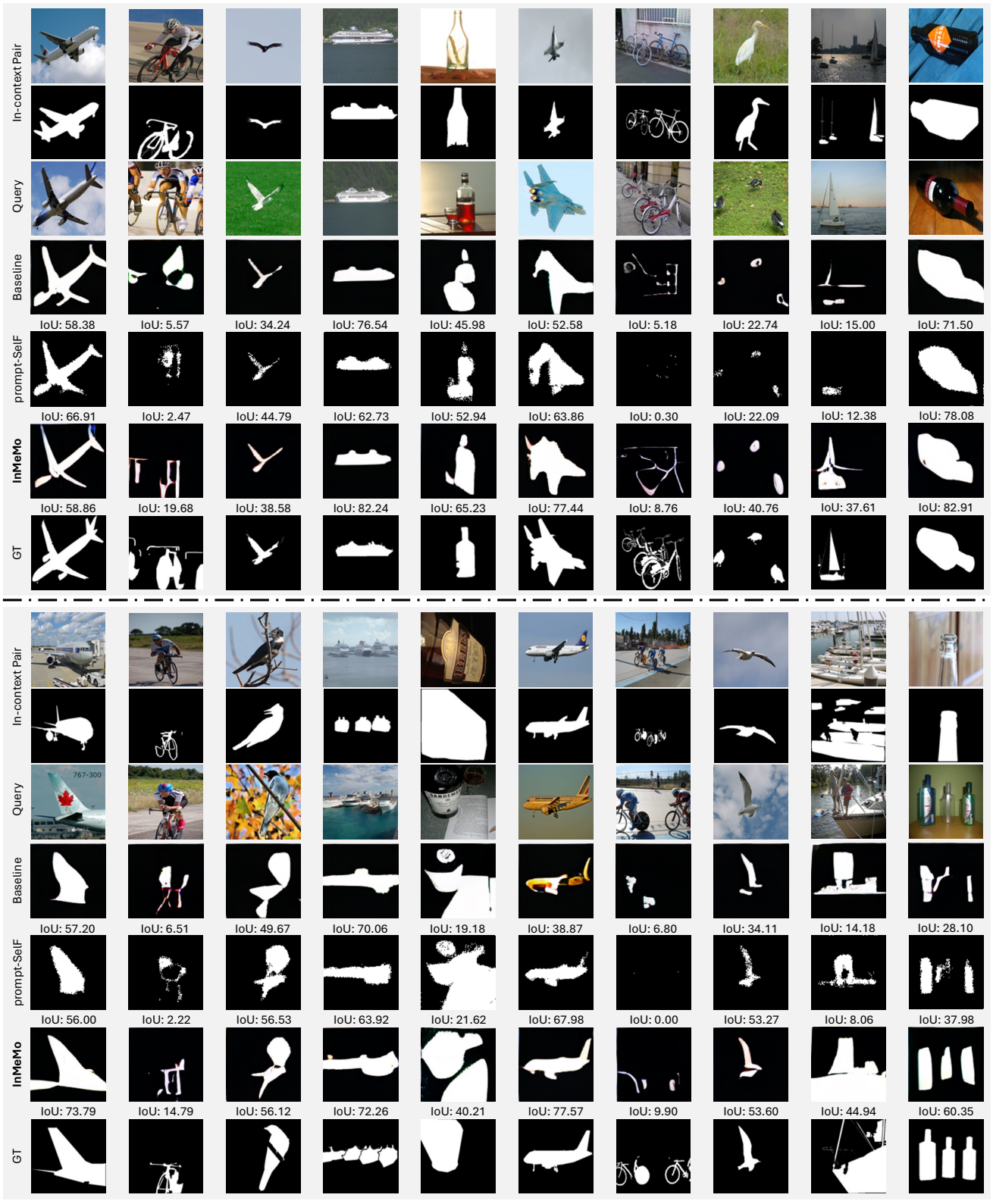

Figure 2. The visual examples in Fold-0. Each column represents a different class. We arrange them in the order of in-context pair, query image, baseline, prompt-SelF, **InMeMo**, and the ground-truth label (GT). We show all five classes in this fold: *aeroplane*, *bicycle*, *bird*, *boat*, *bottle*.

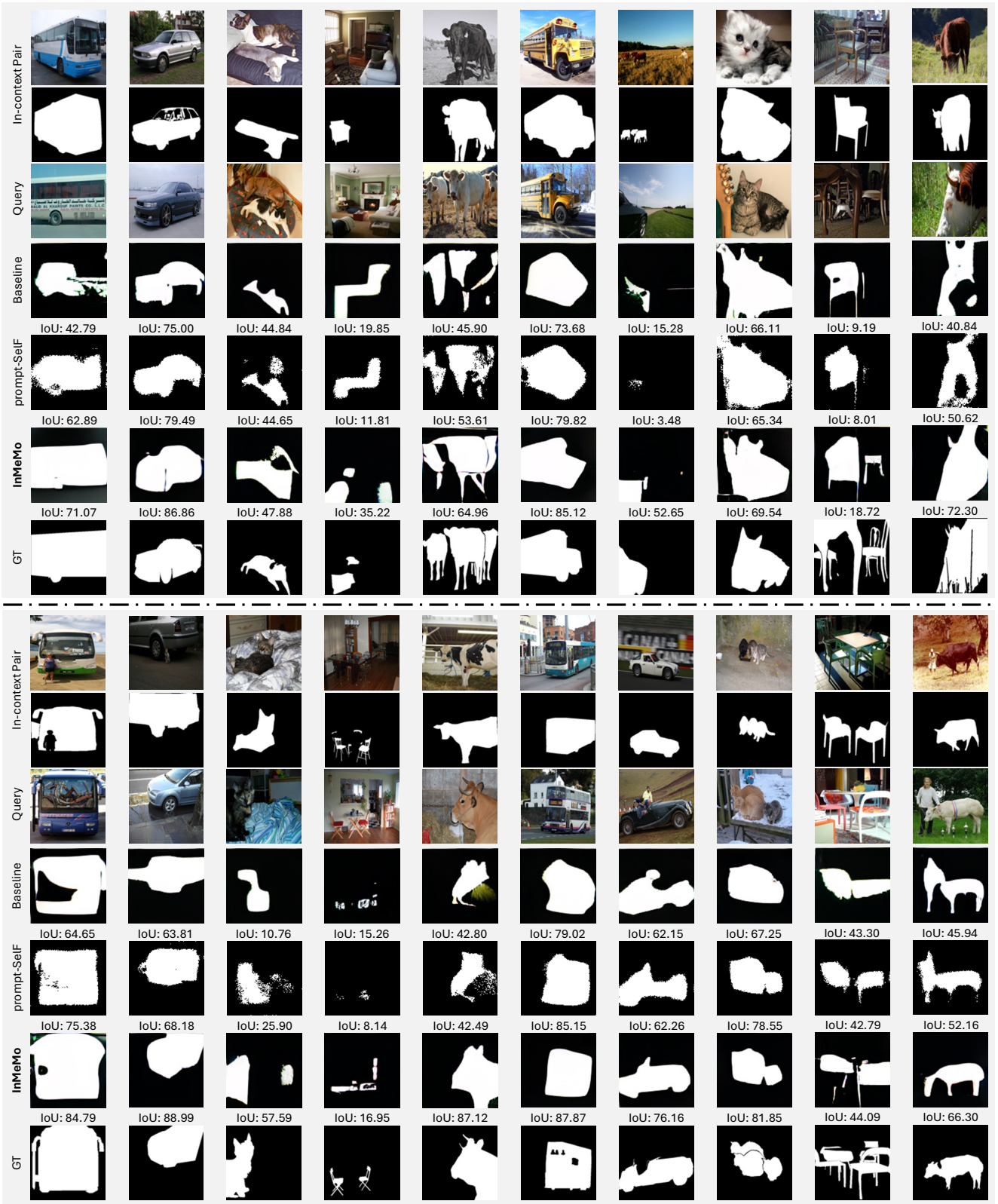

Figure 3. The visual examples in Fold-1: *bus*, *car*, *cat*, *chair*, *cow*.

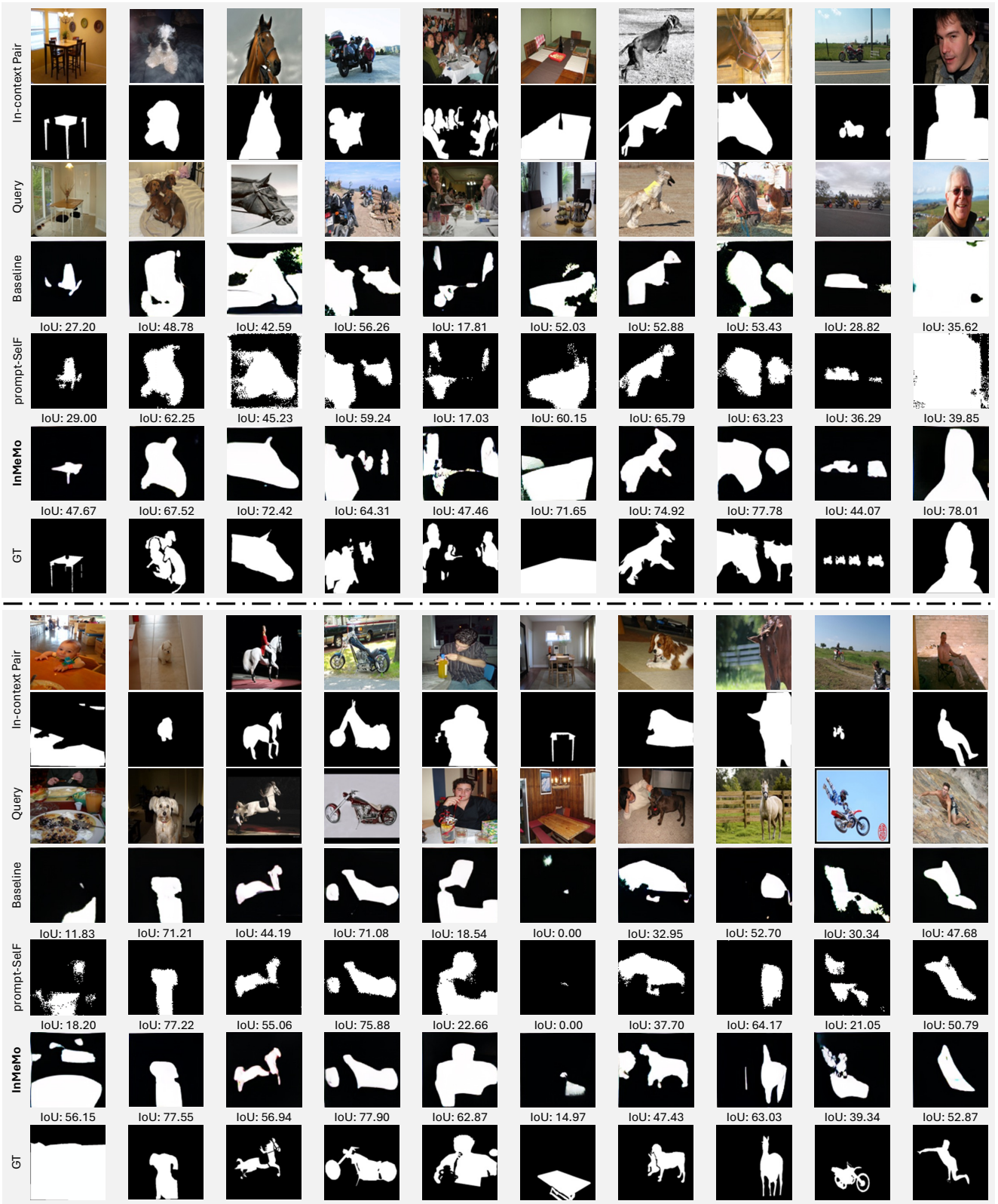

Figure 4. The visual examples in Fold-2: *dinningtable*, *dog*, *horse*, *motorbike*, *person*.

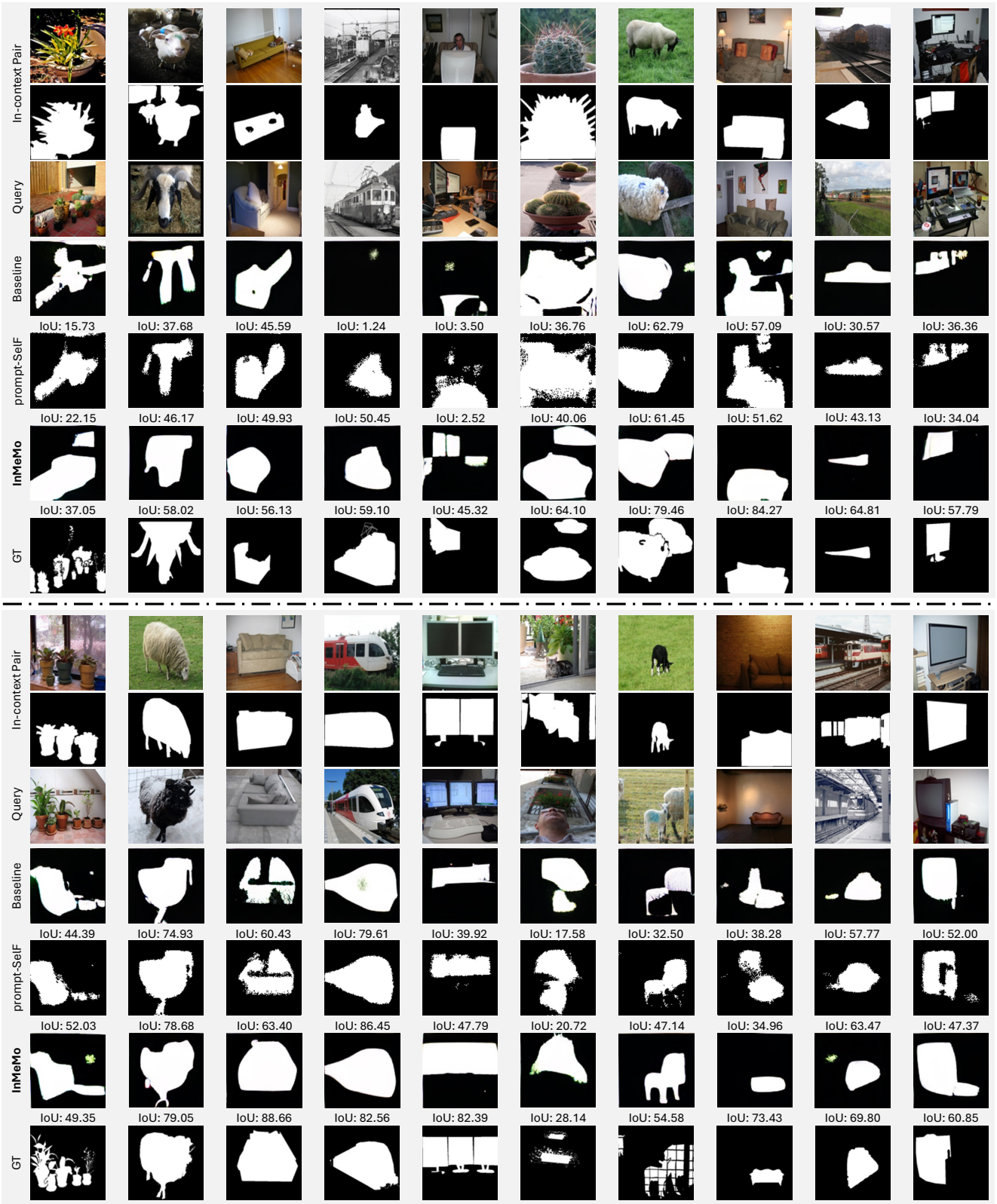

Figure 5. The visual examples in Fold-3: *pottedplant*, *sheep*, *sofa*, *train*, *tvmonitor*.

Figure 6. The visual examples in the single object detection task.

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 1

[2] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei Efros. Visual prompting via image inpainting. *Advances in Neural Information Processing Systems*, 35:25005–25017, 2022. 3

[3] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 1

[4] Yanpeng Sun, Qiang Chen, Jian Wang, Jingdong Wang, and Zechao Li. Exploring effective factors for improving visual in-context learning. *arXiv preprint arXiv:2304.04748*, 2023. 3

[5] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. What makes good examples for visual in-context learning? *arXiv preprint arXiv:2301.13670*, 2023. 1