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

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

Abstract

*Large-scale models trained on extensive datasets, have emerged as the preferred approach due to their high generalizability across various tasks. In-context learning (ICL), a popular strategy in natural language processing, uses such models for different tasks by providing instructive prompts but without updating model parameters. This idea is now being explored in computer vision, where an input-output image pair (called an in-context pair) is supplied to the model with a query image as a prompt to exemplify the desired output. The efficacy of visual ICL often depends on the quality of the prompts. We thus introduce a method coined **In**struct **Me** **Mo**re (InMeMo), which augments in-context pairs with a learnable perturbation (prompt), to explore its potential. Our experiments on mainstream tasks reveal that InMeMo surpasses the current state-of-the-art performance. Specifically, compared to the baseline without learnable prompt, InMeMo boosts mIoU scores by 7.35 and 15.13 for foreground segmentation and single object detection tasks, respectively. Our findings suggest that InMeMo offers a versatile and efficient way to enhance the performance of visual ICL with lightweight training. Code is available at https://github.com/Jackieam/InMeMo.*

1. Introduction

The advancement of large-scale models has been profound in recent years. They have demonstrated remarkable abilities to generalize and hold potential for diverse downstream tasks [5, 10, 36]. Models such as ChatGPT/GPT-3 [6], have emphasized the intrinsic capacity of *in-context learning* (ICL) for Natural Language Processing (NLP) tasks [14, 17, 30, 38, 40, 45, 48]. ICL allows models to undertake new tasks using prompts to predict unseen samples, eliminating the need for model parameter adjustments and reducing training costs. While teeming with potential as a fundamental approach for real-world applications of large-scale models, ICL for computer vision tasks still remains in its exploratory stages [49].

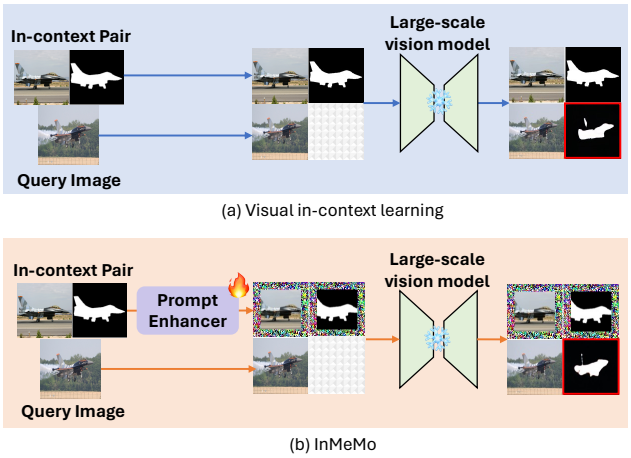

Figure 1. A schematic comparison of current visual ICL and InMeMo. (a) Visual ICL compiles a query image and in-context pair to create a four-cell grid canvas with an empty cell for a prediction (located in the bottom-right cell in this diagram), which forms a *prompt* for visual ICL. The prediction (depicted in the red box) is obtained by feeding the prompt into a frozen large-scale vision model. (b) InMeMo additionally uses a *learnable prompt*, which is a perturbation to amend the distribution of prompts.

MAE-VQGAN [4] marks a pioneering effort, showcasing the feasibility of ICL in computer vision across various tasks, such as image segmentation, inpainting, and style transfer. This method employs visual prompts in a grid format as in Figure 1(a), comprising a query image and an input-output pair, called an in-context pair, that exemplifies the task to be solved with an input image and its corresponding label image. Some studies emphasize the pivotal role of in-context pairs for better instructing a model in generating desired outputs. That is, visual ICL demands an in-context image that is similar to the query image in terms of its semantics, viewpoint, *etc*. [49]) as shown in Figure 2, making in-context pair retrieval an indispensable step.

Despite the notable success [42, 49] achieved, retrieved

*Corresponding author.

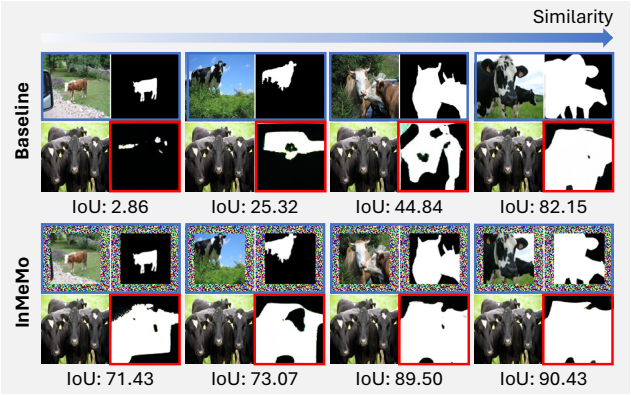

Figure 2. The performance of visual ICL on a foreground segmentation task. Blue boxes and red boxes are in-context pairs and predicted label images (query images are not marked). The in-context pair largely affects the performance. Without a learnable prompt, the performance depends much on the similarity of the query and in-context images. InMeMo, which uses learnable prompts, generates more consistent predictions.

in-context pairs may not be optimal due to the finite size of the dataset to retrieve and a gap between prompts and knowledge in a large-scale vision model. This observation inspires us with an idea: *Can we transform the prompt to better instruct the model for downstream tasks in visual ICL?*

Learnable prompting[1] [3,7,33], which applies a transformation to the model's inputs without modifying the model itself for adapting to various downstream tasks, shows superior performance in image classification. This method, which can be seen as a type of parameter-efficient transfer learning (PETL) [21, 24, 25, 52], is particularly effective in large-scale models compared to fine-tuning, primarily because large-scale models involve enormous training parameters and require significant computational resources even for fine-tuning [3, 7, 11, 32]. Notably, the learnable prompt has demonstrated a robust capability to fit data, even when there are significant discrepancies present [3, 7, 33].

We are thus pioneering our visual ICL method **In**struct **Me** **Mo**re (**InMeMo**) for instructing a large-scale model by a visual learnable prompt. After in-context pair retrieval, we amend the pair with our prompt enhancer, as in [3]. As with the existing visual ICL methods, InMeMo compiles the enhanced pair and the query image into a single image called canvas, which is then fed into a pre-trained large-scale vision model [4]. Our learnable prompt is trained in a supervised manner to generate the corresponding ground-truth label image for the query.

Contributions. InMeMo is a PETL approach, enjoying a lightweight training process. A learnable prompt dedicated to a given downstream task translates the distribution of entire prompts to make them more task-specific and improve the large-scale model's encoding and decoding efficiency. Our experimental results successfully support our claim by showing new state-of-the-art (SOTA) performance in foreground segmentation and single object detection tasks. Although training is indispensable for InMeMo, it effectively alleviates the challenges posed by lower-quality visual prompts.

[1] In [3], the idea of adding a learnable pixel-level perturbation to images is called *visual prompting*; however, as our work also involves visual prompts consisting of an in-context pair and a query, we rephrase a learnable perturbation with a *learnable prompt*.

2. Related Work

2.1. In-Context Learning

ICL is a recent paradigm in NLP for large language models (LLMs), like GPT-3 [6]. With several pre-defined input-output pairs for a specific task, this approach enables an autoregressive model to enhance performance without tuning model parameters for inference [42]. ICL has been verified to be strong enough with several advantages [9], such as offering an interpretable interface to communicate with LLMs [6, 27, 29], being similar to human decision-making processes [47], and instantiating a language model as a service [41]. It also leads to new applications in various fields [8, 22, 31], such as solving mathematics reasoning problems [46], question answering [31, 35], and compositional generalization [2, 18].

In the field of computer vision, ICL is still a new concept with limited existing work [1, 4, 44, 49]. The challenge in visual ICL lies in specifying the task that the model solves, whereas ICL for NLP uses textual instruction. Bar *et al*. [4] proposed to use an input-output image pair, called an in-context pair, with a query image to exemplify the desired output. This combination of them into one image casts the given task as a specific inpainting task. Subsequently, Zhang *et al*. [49] proposed to train a prompt selection model in a supervised manner and demonstrated prompt (in-context pair) selection, and the number of prompts provided to the model is the key to improving the performance of visual ICL. Sun *et al*. [42] suggested using pixel-level in-context pair retrieval for prompt selection. Additionally, they investigated eight different arrangements of the in-context pair and query image and fused the results to enhance ICL performance.

In-context pairs have been proven essential for optimizing performance in downstream tasks [4, 42, 49]. Nonetheless, prior literature has not yet investigated the transformation of the in-context pair to enhance the performance of visual ICL. We aim to explore the potential benefits of introducing learnable perturbation to in-context pairs for improving downstream task performance.

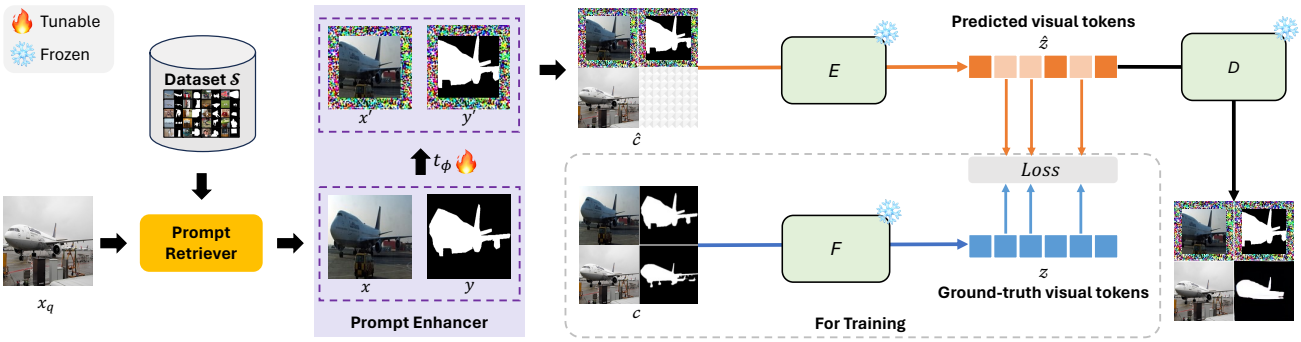

Figure 3. The overall framework of the proposed InMeMo method. First, we employ the Prompt Retriever from the dataset $\mathcal{S}$ to select an in-context pair (x, y) for a query image x_q. We then use a Prompt Enhancer $t_\phi(\cdot)$ to add perturbations to the in-context pair separately to obtain an enhanced in-context pair (x', y'). We create a four-cell grid canvas $\hat{c}$ containing $(x', y', x_\text{q}, \varnothing)$, with an empty cell at the bottom right. The $\hat{c}$ is fed into a frozen `MAE-VQGAN` (E) to generate predicted visual tokens $\hat{z}$ containing the empty cell in $\hat{c}$. For visualized prediction, the $\hat{z}$ is decoded to visual pixels by the decoder of VQGAN (D). To train our InMeMo, a ground-truth canvas c containing $(x, y, x_\text{q}, y_\text{q})$ is fed into a pre-trained encoder of VQGAN (F) to generate ground-truth visual tokens z. We calculate the cross-entropy loss upon the empty cell to **only** update the Prompt Enhancer parameter ϕ.

2.2. Learnable Prompting

In NLP, prompting can be used to guide LLMs to better adapt to downstream tasks [28]. For instance, GPT-3 [6] has shown outstanding generalization ability for different downstream tasks, but costly manually-designed prompting is often necessary. Furthermore, full fine-tuning demands enormous computational resources due to large model sizes. PETL optimizes a small subset or an additional set of parameters of LLMs to specific downstream tasks as in adapter [19, 34] and prompt tuning [20, 24], to achieve competitive performance compared to full fine-tuning.

Due to the exceptional performance of PETL in the field of NLP, numerous previous studies have endeavored different attempts in vision [21, 33, 43] and vision-and-language models [36, 51, 52]. This is usually accompanied by partially fine-tuning the model or adding learnable prompts to the input image. As a latter approach, Bahng *et al.* [3] proposed incorporating learnable pixel-level input-independent visual prompting (VP) into the input image to enhance the transferability of large-scale frozen models, such as CLIP [36], to downstream tasks. This optimization process only involves a significantly smaller set of parameters than the large-scale model, making VP a well-suited extension for visual ICL. This paper explores the potential of VP for this purpose.

3. Method

Let $\mathcal{S} = \{(x, y)\}$ denote a dataset of pairs of an input image x and a label (output) image y for a specific downstream task, where $|S| = n$. Given this dataset and a query image x_q as input, a prediction y_q of the task is generated.

Figure 3 shows an overview of InMeMo. A query image x_q is fed into the prompt retriever to find an in-context pair (x, y) from dataset $\mathcal{S}$. The prompt enhancer then takes them to obtain a pair (x', y') with a learnable prompt. The pair is concatenated with the query image x_q to form a four-cell grid canvas, denoted by a quadruple $(x', y', x_\text{q}, \varnothing)$, where $\varnothing$ represents an empty cell. The canvas is fed into a frozen pre-trained large-scale vision model E to obtain visual tokens $\hat{z} = E(x', y', x_\text{q}, \varnothing)$. The visual tokens corresponding to the empty cell encode the prediction $\hat{y}_\text{q}$ of the task. Decoder D gives prediction $\hat{y}_\text{q}$ as $\hat{y}_\text{q} = D(\hat{z})$.

The key component in InMeMo is the prompt enhancer, denoted by t_ϕ, with a set ϕ of learnable parameters. We train the prompt enhancer with dataset $\mathcal{S}$ so that it is instructive enough to specify the task even when the retriever cannot find an in-context pair with sufficient quality.

3.1. Prompt Retriever

Finding a high-quality in-context pair for a given query image is non-trivial for better performance [49]. Our prompt retriever follows pixel-level retrieval in [42] for prompt selection. We first use an off-the-shelf feature extractor (*e.g.*, CLIP visual encoder [36]) to obtain ℓ_2-normalized visual features of query image x_q and of in-context image $x \in \mathcal{S}$. The in-context pair in $\mathcal{S}$ whose visual feature is most similar to the query's, is used as in-context pair (x, y), *i.e.*,

$$(x, y) = \underset{(x^\star, y^\star) \in S}{\arg\max}\, v(x_\text{q})^\top v(x^\star), \tag{1}$$

where $v(\cdot)$ gives the visual features after the normalization.

3.2. Prompt Enhancer

The learnable prompt is conceived in [3], inspired by the notable successes of prompting in NLP [6, 15, 28]. It addresses the domain shift problem, offering a way to adapt

source domain input data to the target domain downstream task without parameter tuning of the source model. We use a pixel-level perturbation added around the edges of images as a learnable prompt as in [3] to facilitate task performance.

As the primary role of the learnable prompt is to amend the input image, our prompt enhancer adds a learnable prompt to in-context pairs. Such extended input-output examples will implicitly instruct the frozen model on the desired output and thus narrow the gap between in-context pairs and a query image. Our learnable prompt is agnostic to input, so the same prompt is shared for all in-context pairs of the same task. This means our learnable prompts can be viewed as a task identifier.

Given the pair (x, y) from the prompt retriever, the prompt enhancer adds to them a learnable prompt t_ϕ parameterized by ϕ to generate (x', y') as

$$x' = x + \delta t_\phi, \quad y' = y + \delta t_\phi, \tag{2}$$

where δ specifies the magnitude of the perturbation. The prompt t_ϕ is in the image space. ϕ denote the set of pixel around the edges that are learnable via backpropagation, and the other pixels are all zero.

3.3. Prediction

Following [4], we adopt the MAE-VQGAN model, in which pre-trained MAE [16] E generates visual tokens $\hat{z}$ from (x', y') and x_q. The VQGAN [12] decoder D, again pre-trained, generates resulting image $\hat{y}_\mathrm{q}$ from $\hat{z}$.

After compiling the in-context pair and the query into a canvas $\hat{c} = (x', y', x_\mathrm{q}, \varnothing)$, E predicts latent visual tokens $\hat{z} = (\hat{z}_1, \ldots, \hat{z}_K)$, specifically,

$$\hat{z}_k = \arg\max_w E_{kw}(\hat{c}), \tag{3}$$

where $\hat{z}_k \in \hat{z}$ is a visual token in the vocabulary $\mathcal{V}$ at spatial position k, and E_{kw} gives the probability of $w \in \mathcal{V}$ for k. D then generates a label image by

$$\hat{y}_\mathrm{q} = D(\hat{z}). \tag{4}$$

We obtain the prediction for the query x_q as $\hat{y}_\mathrm{q}$.

3.4. Training

The only learnable parameters in InMeMo are the prompt t_ϕ. We train it for a specific task on $\mathcal{S}$. The loss is the same as [4], while all parameters except for t_ϕ are frozen.

We first randomly choose a pair $(x_\mathrm{q}, y_\mathrm{q})$ as query from $\mathcal{S}$. The InMeMo prediction process from the prompt retriever is then applied to x_q to compute $\hat{z}$, but the retriever uses $\mathcal{S} \setminus \{(x_\mathrm{q}, y_\mathrm{q})\}$ instead of $\mathcal{S}$.

The label image y_q is used for training. We compile the retrieved in-context pair (x, y) and $(x_\mathrm{q}, y_\mathrm{q})$ into a canvas $c = (x, y, x_\mathrm{q}, y_\mathrm{q})$. The pre-trained VQGAN encoder F associated D gives the ground-truth visual tokens z that reconstruct y_q with D, *i.e.*,

$$z_k = \arg\max_w F_{kw}(c), \tag{5}$$

where F_{kw} again is the probability of $w \in \mathcal{V}$ for position k. The loss L to train our learnable prompt t_ϕ is given by

$$L(\phi) = \mathbb{E}[\mathrm{CE}(E_k(\hat{c}), z_k)], \tag{6}$$

where CE is the cross-entropy loss, $E_k(\hat{c}) \in \mathbb{R}^{|\mathcal{V}|}$ is the probabilities of respective tokens in $\mathcal{V}$, and the expectation is computed over all $(x_\mathrm{q}, y_\mathrm{q}) \in \mathcal{S}$ as well as all visual tokens z_k corresponding to y_q (*i.e.* over the latent visual tokens of $\varnothing$, represented as masked index).

3.5. Interpretation

Adding t_ϕ to images in a visual prompt as in Eq. (2) translates the distribution of the prompt in a certain direction. Determining t_ϕ by Eq. (6) will encode some ideas about the task described by $\mathcal{S}$ in ϕ, supplying complementary information that is not fully conveyed by the in-context pair (x, y). We consider that our training roughly aligns the distributions of image patches $\hat{c}$ and c in the latent space before visual token classification with smaller degrees of freedom in t_ϕ. This can be particularly effective as these distributions are inherently different due to the lack of the ground-truth label image y_q in the canvas. Therefore, our best expectation is that ϕ captures the distribution of y_q collectively to bring the distribution of prompts closer to the ground-truth prompts (containing ground-truth label y_q). With this, the encoder E will have better access to more plausible visual tokens that decode a label image closer to the ground-truth label.

4. Experiments

4.1. Experimental Setup

Datasets and Downstream Tasks. We follow the experimental settings of [4] to evaluate InMeMo. As downstream tasks, we perform foreground segmentation and single object detection. (1) **Foreground segmentation** aims to extract apparent objects from the query image with the in-context pair. We use the Pascal-5^i dataset [39], which is split into four-fold subsets, each containing five classes. (2) **Single object detection** evaluates whether a model can capture fine-grained features specified by a coarse-grained bounding box in the in-context pair [42]. We conduct experiments on images and bounding boxes from the PASCAL VOC 2012 [13]. To align with [4], we use a subset of the dataset whose samples only contain a single object as our dataset $\mathcal{S}$, ensuring the annotation mask occupies less than 50% of the entire image for the training set, and 20% for the test set.

Table 1. Performance of the foreground segmentation and single object detection downstream tasks. The best scores in *in-context learning* are highlighted in **bold**. The baseline scores are based on our reproduction. **Seg.** and **Det.** stand for the segmentation and single object detection tasks, respectively.

		Seg. (mIoU ↑)					**Det.**
		Fold-0	Fold-1	Fold-2	Fold-3	Mean	(mIoU ↑)
Meta-learning	OSLSM [39]	33.60	55.30	40.90	33.50	40.80	-
	co-FCN [37]	36.70	50.60	44.90	32.40	41.10	-
In-context learning	Baseline	35.69	38.25	35.86	33.37	35.79	28.08
	Random [4]	28.66	30.21	27.81	23.55	27.56	25.45
	UnsupPR [49]	34.75	35.92	32.41	31.16	33.56	26.84
	SupPR [49]	37.08	38.43	34.40	32.32	35.56	28.22
	prompt-SelF [42]	**42.48**	43.34	39.76	38.50	41.02	29.83
	InMeMo (Ours)	41.65	**47.68**	**42.43**	**40.80**	**43.14**	**43.21**

Methods for comparison. All experiments use MAE-VQGAN [4] as the pre-trained large-scale vision model. InMeMo is compared against the SOTA methods of visual ICL (*i.e.*, Random [4], UnsupPR [49], SupPR [49], and prompt-SelF [42]) as well as few-shot segmentation derived from meta-learning (*i.e.*, OSLSM [39] and co-FCN [37]). Our baseline is pixel-level retrieval [42] but without the learnable prompt.

Implementation details. For foreground segmentation, we train InMeMo for each fold of the training set separately, meaning each fold is viewed as a task, and a learnable prompt is obtained for each fold. For single object detection, we train InMeMo on the whole training set by retrieving in-context pairs from the training set. For testing, each image in the test set will be considered as a query image to retrieve an in-context pair from the training set.

We resized the image size to 224×224 for the prompt enhancer. A learnable prompt occupies 30 pixels from each edge; therefore, ϕ contains $(224^2 - (224 - 2 \times 30)^2) \times 3$ parameters. The images are then resized to 111×111 to create the canvas. An in-context pair (x', y') with a learnable prompt, a query image x_{q}, and an empty image $\varnothing$ are arranged at top-left, top-right, bottom-left, and bottom-right, respectively, following the default arrangement of [4]. We set δ to 1. InMeMo is implemented using PyTorch and trained for 100 epochs with Adam [23]. We initiate training with a learning rate of 40, which decays based on the cosine annealing warm restarts scheduler. A notable advantage of InMeMo is its efficiency—this training operates on a single NVIDIA GeForce RTX 4090 with a batch size of 32.

4.2. Comparison with State-of-the-Art

We compared InMeMo with prior visual ICL methods and meta-learning-based few-shot learning methods in Table 1. Our analysis reveals that InMeMo achieved the SOTA, surpassing the previous SOTA in both downstream tasks, particularly on single object detection. Apparently, it significantly outperformed the baseline. Our method also outperformed the meta-learning-based methods on some folds and on average. This highlights the efficacy of integrating the learnable prompt into visual ICL.

More specifically, for the foreground segmentation task, we observe that while InMeMo does not achieve the best score on Fold-0, it nonetheless considerably exceeds the baseline. Prompt-SelF's performance could be affected by the bagging effect. That is, prompt-SelF is applied to eight different arrangements of images in the canvas and fuses the results, thereby harnessing the latent expertise of the large-scale vision model. In contrast, InMeMo runs inference for a single query only once. Bagging can be an interesting tweak to improve the performance without extensive efforts, but still, we emphasize the significant gain of InMeMo by itself. Notably, InMeMo showcases outstanding proficiency in the single object detection task, surpassing the prevailing SOTA by a margin of 13.38 points. This performance gain, demonstrates the exceptional ability of InMeMo to capture fine-grained features in detecting small objects within images.

These results shed light on our direct and efficient approach. By amending in-context pairs, we can effectively harness the learnable prompt to improve performance in visual ICL. Moreover, InMeMo stands out with its lightweight nature, using only 69,840 additional parameters and demanding minimal training resources. The shared pixel-level learnable prompts of InMeMo hold the potential to pave the way toward even more efficient and effective visual ICL.

4.3. Domain Shift Analysis

Real-world applications often exhibit domain shifts from dataset $\mathcal{S}$, leading to discrepancies in model performance in comparison with in-domain evaluation due to differing distributions. Such domain shifts can be observed across datasets, and the resulting performance disparities among datasets can serve as a benchmark for evaluating model ro-

Table 2. Domain shift analysis on InMeMo. *Pascal* $\rightarrow$ *Pascal* means in-context pairs and query images both source from PASCAL (as with Table 1). *COCO* $\rightarrow$ *Pascal* indicates that in-context pairs are from COCO and query images are from PASCAL. The baseline scores are our reproduction.

	Method	Fold-0	Fold-1	Fold-2	Fold-3	Means
Pascal $\rightarrow$ *Pascal*	Baseline	35.69	38.25	35.86	33.37	35.79
	InMeMo	41.65	47.68	42.43	40.80	43.14
COCO $\rightarrow$ *Pascal*	Baseline	33.83	36.11	32.89	30.64	33.37
	InMeMo	38.74	43.82	40.45	37.12	40.03

Table 3. Segmentation performance for some combinations of images in a canvas to which the learnable prompt are added. `I`, `L`, and `Q` means in-context image, in-context label image, and query image, respectively.

	Fold-0	Fold-1	Fold-2	Fold-3	Mean
Baseline	35.69	38.25	35.86	33.37	35.79
prompt-SelF [42]	42.48	43.34	39.76	38.50	41.02
Combination (InMeMo variant)					
`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

bustness [50]. To assess InMeMo's sensitivity to domain shift, we employ the COCO dataset [26] for inference, following the same setting as in [49]. The COCO dataset is divided into four subsets, each of which mirrors the categories of Pascal-5^i, denoted as COCO-5^i [49]. We source the in-context pair from COCO-5^i and obtain the query image from the validation set of Pascal-5^i, consistent with [42]. This specific configuration is termed as *COCO* $\rightarrow$ *Pascal*.

Table 2 summarizes our domain shift evaluation results. For the *COCO* $\rightarrow$ *Pascal* configuration, the baseline marks a drop in mIoU score by 2.42. In contrast, InMeMo hits 40.03%, reflecting a drop of 3.11. The gap between them is 0.69, indicating that InMeMo is robust against domain shift. Consequently, visual ICL with the learnable prompt has the potential to be transferable, making InMeMo reliable for various real-world applications.

4.4. More Analysis on InMeMo

This section further investigates the capabilities of InMeMo through a series of experiments, primarily focusing on the foreground segmentation task.

Qualitative comparison. We qualitatively compare InMeMo with the baseline, prompt-SelF[2] [42], and the ground-truth (GT) label using examples for both foreground segmentation and single object detection tasks, which are shown in Figure 4. For the foreground segmentation task (Figure 4(a)), InMeMo produces details faithful to the ground-truth label images. Interestingly, InMeMo remains robust against variations, including when provided with an achromatic image or when a significant color disparity exists between the in-context and the query images. Moreover, the InMeMo appears resistant to variations in foreground size (the fourth column from the right), and it distinguishes the background in the query image. However, when the in-context and query images closely align in terms of their features (*e.g.*, semantics, viewpoints, sizes, poses [49]), InMeMo's performance matches that of the baseline and prompt-SelF.

In the single object detection task (Figure 4(b)), InMeMo consistently displays its detail-oriented nature and is unfazed by color variations or object size differences in the in-context pair. Particularly notable is its competency in scenarios where the presence of the foreground in the in-context pair is minimal. Nevertheless, akin to the segmentation task, when the in-context and query images bear a strong resemblance, InMeMo mirrors the performance of the prompt-SelF.

Which images should the learnable prompt be added? Our recommendation leans towards introducing the learnable prompt only to in-context pairs, which seems pivotal in enhancing visual ICL's efficacy across tasks, given the guiding nature of in-context pairs. To discern the potential impacts of different combinations of images to which the learnable prompt is added, we assessed five InMeMo variants: only *in-context image* (`I`), only *query image* (`Q`), both *in-context image* and *query image* (`I` & `Q`), *in-context image* and *in-context label image* (`I` & `L`, identical to InMeMo), and *in-context image*, *in-context label image*, and *query image* (`I`, `L`, & `Q`).

The scores of these combinations are summarized in Table 3. We found that the learnable prompt, irrespective of location, improves the visual ICL performance. Surprisingly, adding the prompt to in-context images produced suboptimal performance among the InMeMo variants but outperformed prompt-SelF. This indicates that the learnable prompt effectively improves the quality of the in-context

[2]We reproduced the prompt-SelF to generate visual examples.

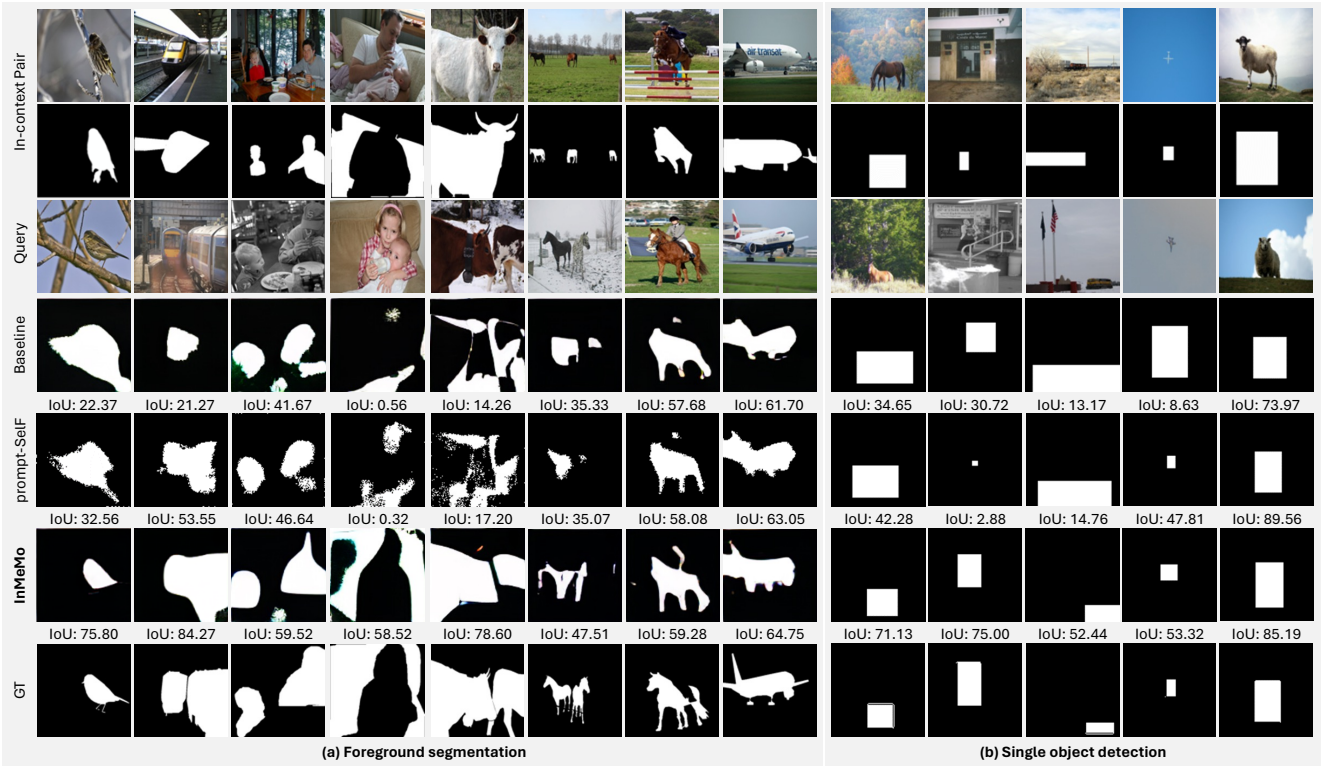

Figure 4. Some examples of baseline, prompt-SelF, and our InMeMo over the two downstream tasks: **(a)** Foreground segmentation and **(b)** Single-object detection. In each task, the upper two rows are in-context pairs, and the third row is the query image. We arrange rows from top to bottom with the order of the baseline, prompt-SelF, **InMeMo**, and the ground-truth label (GT). InMeMo can lead the visual ICL to capture detailed features and overcome inconsistency between in-context and query images. Moreover, InMeMo behaves like it neglects poor-quality in-context pairs, which is another strong advantage when the prompt retriever cannot find a similar in-context image. More examples can be found in our supplementary material.

pair and can enhance the visual ICL performance.

We also found that the performance is less effective when we add the learnable prompt to in-context and query images (`I` & `Q`) than when we add it to only one image (`I`, `Q`). This can be attributed to the agnostic learnable prompt. When the identical prompt is added to both in-context and query images, the model struggles to narrow the gap between them effectively. Similarly, this underlying factor leads to compromised performance in the `I`, `L`, & `Q` configuration, with a particularly notable performance reduction in the more challenging Fold-3.

Is InMeMo performance sensitive to the dataset size? Given the efficiency and simplicity of the learnable prompt, we sought to elucidate the relationship between the volume of the dataset $\mathcal{S}$ and the performance of InMeMo. We conducted experiments for each fold, randomly picking 16, 32, 64, 128, and 256 images from each class to compose $\mathcal{S}$. Figure 5 depicts the relationship.

Our empirical results suggest that the overall performance (represented as Mean) surpasses the baseline score (35.79%) when using at least 64 images per class (36.04%). The performance tends to improve as the number of images increases. Specifically, for Fold-1, which is comparatively easy, InMeMo achieves the mIoU accuracy of 36.63% with only 16 images and consistently outperforms the other folds. Fold-0 substantially increases accuracy, starting from 32 images, saturated at 256. Fold-2 consistently shows a significant improvement as the number of images increases. In Fold-3, there is a considerable increase from 64 to 128 images, after which the score becomes saturated and only gradually increases when all images are used. In general, for easier folds, InMeMo requires fewer images; however, when dealing with intricate scenes, increasing the dataset size can enhance the performance of InMeMo.

Inter-class generalizability of InMeMo. We have demonstrated that InMeMo works well on poor datasets and are curious about its generalizability to unseen classes not included in the dataset $\mathcal{S}$. For this, we train a learnable prompt for each of the 20 classes. Specifically, let $\mathcal{S}_\omega$ denote the subset of images and label images in Pascal-5^i for class ω. InMeMo training uses an image in $\mathcal{S}_\omega$ as a query. It also uses a pair of an image and a label image in $\mathcal{S}_\omega$ as

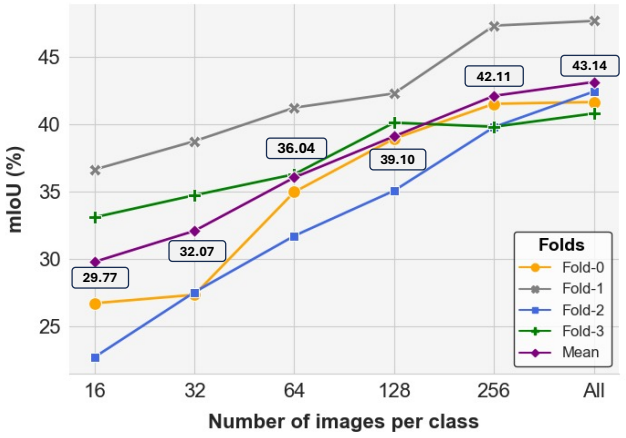

Figure 5. The performance of InMeMo in mIoU of each fold for the number of images per class in $\mathcal{S}$. All means to use all images in the training set. We annotate the scores of Mean in the figure.

an in-context pair. We then run predictions using images in $\mathcal{S}_{\omega'}$ as queries and in-context pairs, where $\omega' \neq \omega$, for measuring the inter-class generalizability. As different classes have varying levels of difficulties, we only show the classes whose *intra-class* performance is higher than the mean mIoU score (43.14%) in Table 1. Our supplementary material shows full results. We discovered that the `bus` and `sheep` are the most *general* classes, meaning that prompts trained on different classes yield a high accuracy (mIoU above 50%) on these two classes. In contrast, `person` is the least generalizable class, performing poorly on all other classes. We excluded these three classes as well, ending up with nine classes. The inter-class (as well as intra-class) scores are shown in Figure 6.

The figure indicates that intra-class scores are not always the best among all other classes. The `transportation` super-class, we can see strong generalizability between the classes in it (`aeroplane`-`train`, `car`-`train`, and `motorbike`-`train`), whereas the `transportation` classes typically have lower scores with the `animals` classes like `dog` and `horse`. Within `animals`, the classes usually show strong generalizability except for `cow`, but they do not generalize to the `transportation` classes. We can also identify some exceptions between different classes, such as `aeroplane` having a weak generalizability with `car` and `horse` having a strong generalizability with `train`. We think this is due to the similarity of their label images (*e.g.*, `train` and `cow` often occupy a larger region of the image) and the class-specific difficulty (*e.g.*, the learnable prompt trained for `cow` does not generalize in most cases).

The mean mIoU score over all pairs of 20 classes in the supplementary material is 34.32%. This score is comparable to most methods in Table 1, but suffers from a significant drop from InMeMo's mean score over all folds. This

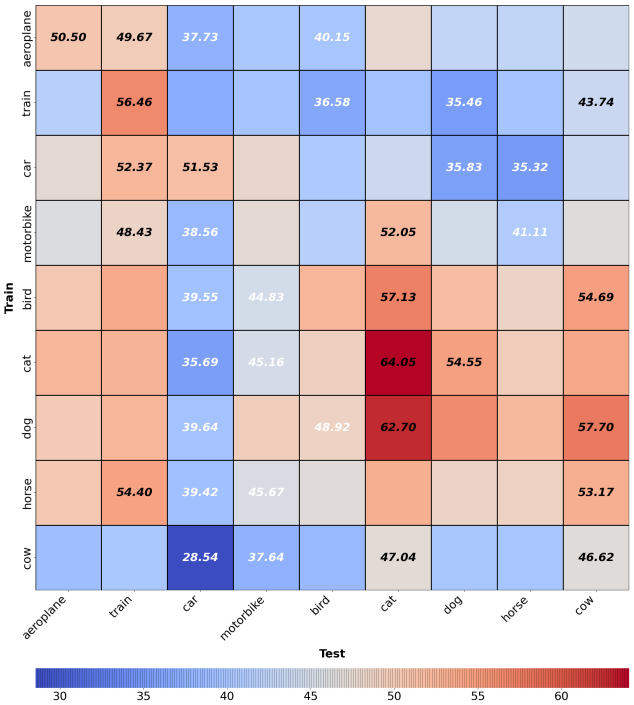

Figure 6. Inter- and intra-class generalization performance in mIoU. The horizontal and vertical axes are the classes used for prediction and training, respectively. The diagonal elements show intra-class performance. Each row shows the two largest and smallest scores in black and white. The nine classes are arranged to form *transportation* and *animal* super-classes. The supplementary material shows the scores for all possible pairs.

implies the importance of tuning the learnable prompt for target tasks.

5. Conclusion

InMeMo shows SOTA performance on the two downstream tasks by incorporating a learnable prompt to in-context pairs, a lightweight tool to facilitate visual ICL. The learnable prompt enables the visual ICL to reconstruct more fine-grained details in predictions and overcome the interference caused by low-quality in-context pairs that are not sufficiently similar to query images. We also showed that InMeMo is robust against domain shift (*e.g.*, from the Pascal dataset to the COCO dataset). **Limitations.** InMeMo requires a minimum of 64 images per class to achieve competitive performance compared with our baseline. Also, a learnable prompt for a certain class does not generalize to other classes. Therefore, the learnable prompt dedicated to the target task is the key to better performance.

Acknowledgements This work was partly supported by JSPS KAKENHI Grant No. JP23H00497, JST CREST Grant No. JPMJCR20D3, and FOREST Grant No. JPMJFR216O.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2

[2] Shengnan An, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Jian-Guang Lou, and Dongmei Zhang. How do in-context examples affect compositional generalization? *arXiv preprint arXiv:2305.04835*, 2023. 2

[3] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022. 2, 3, 4

[4] 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. 1, 2, 4, 5

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 1

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1, 2, 3

[7] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19133–19143, 2023. 2

[8] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. 2

[9] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[11] Gamaleldin F Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. Adversarial reprogramming of neural networks. *arXiv preprint arXiv:1806.11146*, 2018. 2

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 4

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 4

[14] Hila Gonen, Srini Iyer, Terra Blevins, Noah A Smith, and Luke Zettlemoyer. Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*, 2022. 1

[15] Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. Ptr: Prompt tuning with rules for text classification. *AI Open*, 3:182–192, 2022. 3

[16] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 4

[17] Or Honovich, Uri Shaham, Samuel R Bowman, and Omer Levy. Instruction induction: From few examples to natural language task descriptions. *arXiv preprint arXiv:2205.10782*, 2022. 1

[18] Arian Hosseini, Ankit Vani, Dzmitry Bahdanau, Alessandro Sordoni, and Aaron Courville. On the compositional generalization gap of in-context learning. *arXiv preprint arXiv:2211.08473*, 2022. 2

[19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 3

[20] Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*, 2021. 3

[21] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 2, 3

[22] Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *arXiv preprint arXiv:2206.08082*, 2022. 2

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[24] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021. 2, 3

[25] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021. 2

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[27] Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*, 2021. 2

[28] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023. 3

[29] Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*, 2021. 2

[30] Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Noisy channel language model prompting for few-shot text classification. *arXiv preprint arXiv:2108.04106*, 2021. 1

[31] Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*, 2021. 2

[32] Paarth Neekhara, Shehzeen Hussain, Jinglong Du, Shlomo Dubnov, Farinaz Koushanfar, and Julian McAuley. Cross-modal adversarial reprogramming. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2427–2435, 2022. 2

[33] Changdae Oh, Hyeji Hwang, Hee-young Lee, YongTaek Lim, Geunyoung Jung, Jiyoung Jung, Hosik Choi, and Kyungwoo Song. Blackvip: Black-box visual prompting for robust transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24224–24235, 2023. 2, 3

[34] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020. 3

[35] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*, 2022. 2

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 3

[37] Kate Rakelly, Evan Shelhamer, Trevor Darrell, Alyosha Efros, and Sergey Levine. Conditional networks for few-shot semantic segmentation. 2018. 5

[38] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*, 2021. 1

[39] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*, 2017. 4, 5

[40] Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*, 2022. 1

[41] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022. 2

[42] 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. 1, 2, 3, 4, 5, 6

[43] Yun-Yun Tsai, Pin-Yu Chen, and Tsung-Yi Ho. Transfer learning without knowing: Reprogramming black-box machine learning models with scarce data and limited resources. In *International Conference on Machine Learning*, pages 9614–9624. PMLR, 2020. 3

[44] Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6830–6839, 2023. 2

[45] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022. 1

[46] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. 2

[47] Patrick H Winston. Learning and reasoning by analogy. *Communications of the ACM*, 23(12):689–703, 1980. 2

[48] Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning. *arXiv preprint arXiv:2212.10375*, 2022. 1

[49] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. What makes good examples for visual in-context learning? *arXiv preprint arXiv:2301.13670*, 2023. 1, 2, 3, 5, 6

[50] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 6

[51] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16816–16825, 2022. 3

[52] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 2, 3