How to Configure Good In-Context Sequence for Visual Question Answering

Li Li[1†] Jiawei Peng[1†] Huiyi Chen[1†] Chongyang Gao[2] Xu Yang[1*]
[1] School of Computer Science & Engineering, Key Laboratory of New Generation Artificial Intelligence
Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
[2] Department of Computer Science, Northwestern University
lilyli@seu.edu.cn, pengjiawei@seu.edu.cn, huiyichen@seu.edu.cn,
chongyanggao2026@u.northwestern.edu, xuyang_palm@seu.edu.cn

Abstract

Inspired by the success of Large Language Models in dealing with new tasks via In-Context Learning (ICL) in NLP, researchers have also developed Large Vision-Language Models (LVLMs) with ICL capabilities. However, when implementing ICL using these LVLMs, researchers usually resort to the simplest way like random sampling to configure the in-context sequence, thus leading to sub-optimal results. To enhance the ICL performance, in this study, we use Visual Question Answering (VQA) as case study to explore diverse in-context configurations to find the powerful ones. Additionally, through observing the changes of the LVLM outputs by altering the in-context sequence, we gain insights into the inner properties of LVLMs, improving our understanding of them. Specifically, to explore in-context configurations, we design diverse retrieval methods and employ different strategies to manipulate the retrieved demonstrations. Through exhaustive experiments on three VQA datasets: VQAv2, VizWiz, and OK-VQA, we uncover three important inner properties of the applied LVLM and demonstrate which strategies can consistently improve the ICL VQA performance. Our code is provided in: https://github.com/GaryJiajia/OFv2_ICL_VQA.

1. Introduction

Recently, Large Language Models (LLMs) [5, 8, 38] have showed remarkable abilities in solving new tasks through prompt engineering [27] and In-Context Learning (ICL) [9]. However, despite their success, LLMs still remain inscrutable to the research community. To unravel the properties of these models, researchers have drawn inspiration from the “outside-in” methodologies to comprehend complex systems. Analogously, as scientists treat unknown systems as black boxes, conducting experiments to discern the effects of varied inputs on outputs, researchers introduce diverse prompts and analyze the resultant feedback. This strategy provides crucial insights into the inner properties of LLMs[33, 34].

Compared to the standard single sentence prompt, which is one kind of zero-shot prompt, ICL sequences few-shot **demonstrations** where each one contains knowledge about the input and the corresponding label of the task that needs to be solved. Such few-shot nature of ICL enables it to encapsulate more information, resulting in enhanced performance. However, the ICL performance is heavily influenced by various demonstration configurations, such as the selection or ordering of the demonstrations [14, 26, 30]. Consequently, many NLP studies [11, 40, 44, 50] explore how to configure demonstrations to enhance the ICL performance. Meanwhile, NLP researchers also use ICL to unravel the inner properties of large models, owing to its flexible controllability. For example, by controlling the label space of the demonstrations, researchers [34] find that the ICL ability may be demonstrated by two distinct functions: Task Recognition (TR), *i.e.*, the ability to identify the task formulation, and Task Learning (TL), the ability to learn the mapping between input and labels of the demonstrations.

Nowadays, multi-modal learning becomes attractive with the development [39, 46, 47, 52]. Inspired by the success of LLM, researchers in the vision-language (VL) domain have also developed large models with ICL capabilities, such as Flamingo [1] and its corresponding open-source version, Open-Flamingo [2]. However, there is limited research on how to effectively configure demonstrations in these models, both in terms of enhancing the performance of Large Vision-Language Model (LVLM) and exploring its properties. To the best of our knowledge, currently, only one study [45] has explored demonstration configurations for image captioning. Unfortunately, this re-

[*] Corresponding author.
[†] These authors contributed equally to this work.

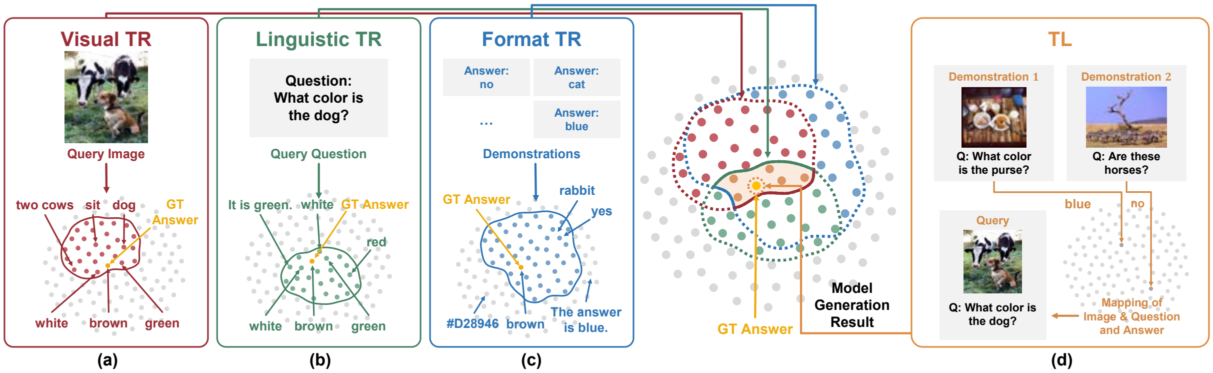


Figure 1. ICL shows two different functions: Task Recognition (TR) and Task Learning (TL). In VQA, TR includes three components: (a) Visual TR and (b) Linguistic TR narrow the label space based on the query image and question, and (c) Format TR recognizes the answer formats from the demonstrations. After combining them, the label space can be narrowed down for better answer the question. While TL learns the mapping between inputs (images&questions) and outputs (answers) from demonstrations to make LVLM get the right answer.

search still fails to make use of ICL to explore the properties of LVLM.

In VL, Visual Question Answering (VQA) is more suitable for exploring the inner properties of LVLM through ICL for two reasons. First, most NLP tasks employed to explore LLMs can be considered as question-answering tasks, *i.e.*, the sentiment classification can be viewed as answering "what is the sentiment of this sentence?". Therefore, VQA is well-suited for adapting the methodologies used in QA studies into the VL domain. Second, VQA encompasses various visual understanding tasks, including classification, counting, locating and so on, allowing for a more comprehensive exploration of LVLM. Therefore, in this study, we explore demonstration configurations in the VQA task with a dual-purpose: *(1) to explore effective demonstration configuration strategies for enhancing VQA performance* and *(2) to gain a better understanding of the inner properties of LVLM*.

To achieve the dual-purpose, we design various demonstration configuration strategies, including retrieving demonstrations based on similarity via images or texts (questions and answers) and using different ways to manipulate the in-context sequence constructed by the retrieved demonstrations, *e.g.*, mismatching the (image, question, answer) triplets, incorporating the instructions, and reordering the demonstrations. Through exhaustive experiments, our research makes the following three key contributions.

- We extend the TR and TL hypothesis to the field of LVLM by refining this hypothesis to interpret and measure the ICL capabilities of LVLM, as depicted in Fig. 1.
- Based on the refined hypothesis, we uncover three important inner properties of LVLM during ICL: limited TL capabilities, the presence of a short-cut effect, and partial compatibility between vision and language modules.
- Building upon these findings, we explain in detail the roles played by various demonstration configuration strategies in LVLM and design new demonstration configuration methods.

2. Related Work

In-Context Learning in NLP. Recently, NLP has witnessed significant advancements in Large Language Models (LLMs). With the increase in model and corpus sizes [8, 37], researchers discovered their emergent capabilities, particularly in prompt engineering [16, 23, 27, 28]. The introduction of even larger models like GPT-3 [4] has unveiled the potential for In-Context Learning (ICL). ICL, a form of specialized prompt engineering, enables LLMs to make predictions based on contextual information supplemented by a few illustrative examples. Numerous investigations have demonstrated the proficient performance of LLMs in various tasks through ICL [35, 42, 43]. This led to a surge of studies exploring the configuration of in-context sequence [10, 13, 17, 30, 36, 41]. However, most of these studies have been limited to NLP tasks, and there is a need to extend this research to other domains.

In-Context Learning in VL. Inspired by the success of LLMs in NLP, the vision-language field has also witnessed the emergence of corresponding large vision-language models (LVLMs) [6, 19, 22, 29]. Some of these models, such as BLIP2 [21], MiniGPT-4 [52], and LLAVA [25], are pretrained by aligning image and text data using adapters [20, 49] to alleviate training burdens. Specifically, they freeze a well-trained LLM and train a smaller network alongside it, leveraging this alignment to enable joint learning from both modalities during pretraining. Although there are numerous large VLMs available, it is important to note that not all models support in-context learning (ICL). For example, mPLUG-Owl [48] and MiniGPT-4 [52] lack the capabilities for ICL because they have not undergone dedicated few-shot pre-training and cannot handle the input distribution associated with in-context learning. In contrast, models like Flamingo [1] and Otter [18] are specifically designed to support this task. However, since Flamingo is not open-source, we utilize its open-source version called Open-Flamingo [2] and IDEFICS [15]. Among them, Otter derives from Open-Flamingo through instruction fine-tuning. In our research, we utilize Open-Flamingo, removing the interference caused by instruction fine-tuning. Additionally, the recently released MMICL [51] model includes pre-training data from the classic VQA datasets such as VQAv2 [12], VizWiz [3], and OK-VQA [32]. Open-Flamingo and IDEFICS, on the other hand, does not use these datasets for pre-training, thus eliminating any interference from being exposed to them during the pre-training process. Therefore, Open-Flamingo and IDEFICS emerges as the most suitable choice for conducting ICL research at present.

Currently, there is limited research on multimodal ICL, with only one study focusing on captioning [45]. We are the first to explore demonstration configuration in the context of the Visual Question Answering (VQA) task.

Configuring In-Context Sequence for QA. In NLP, there is a significant body of research dedicated to demonstration configuration. This research encompasses techniques such as leveraging similarity measures to retrieve more relevant in-context examples [26] or employing machine-generated demonstrations [14]. During the research process, some studies have also identified certain properties of LLMs when applied to in-context learning. For instance, [33] discovered that randomly replacing labels in demonstrations has minimal impact on performance, and as long as the demonstration maintains consistency in terms of format, input distribution, label space, and query, the model can achieve favorable results. [30] have empirically demonstrated that order sensitivity is a common and persistent challenge across various models. Additionally, [34] proposed a deconstruction of ICL into task recognition and task learning, investigating the TR and TL capabilities of models with different shot numbers and scales. Furthermore, [31] observed the presence of a "copying effect" phenomenon within LLMs, which is a type of short-cut inference.

3. In-Context Learning (ICL) for VQA

Given a well-trained Large Vision-Language Model (LVLM) *e.g.*, Flamingo [1], we can use it to solve VQA by ICL. To achieve this, we need to prepare a multi-modal in-context sequence $\mathcal{S} = \{(\boldsymbol{I}_1, \boldsymbol{Q}_1, \boldsymbol{A}_1); (\boldsymbol{I}_2, \boldsymbol{Q}_2, \boldsymbol{A}_2); ...; (\boldsymbol{I}_n, \boldsymbol{Q}_n, \boldsymbol{A}_n); (\hat{\boldsymbol{I}}, \hat{\boldsymbol{Q}})\}$ that consists of n-shot (image $\boldsymbol{I}$, question $\boldsymbol{Q}$, answer $\boldsymbol{A}$) triplets acting as the demonstrations and one test sample $(\hat{\boldsymbol{I}}, \hat{\boldsymbol{Q}})$. Then we input $\mathcal{S}$ to the LVLM for generating the corresponding answer $\hat{\boldsymbol{A}} = \{\hat{a}_1, ..., \hat{a}_T\}$, where the t-th word $\hat{a}_t$ is sampled from the following word distribution $P(\cdot)$ calculated by the LVLM:

$$P(\hat{a}_t | \mathcal{S}, \hat{a}_{1:t-1}) \tag{1}$$

Next, we first extend the hypothesis that ICL can be decoupled into Task Recognition (TR) and Task Learning (TL) from NLP [34] to VL domain in Sec. 3.1 where we further decouple TR into format TR, visual TR, and linguistic TR for better analyzing the ICL ability of a LVLM. Then we introduce the techniques used to configure the demonstrations. The applied techniques include two parts where the first part in Sec. 3.2 shows how to retrieve the samples from a supporting set and the second part in Sec. 3.3 discusses how to manipulate the in-context sequence constructed by the retrieved demonstrations. Due to space constraints, only the effective techniques we utilize are presented here and some other less effective ones are given in the supplementary materials.

3.1. TR and TL in the VL domain

The ability of ICL can be demonstrated by two distinctive functions: Task Recognition (TR) and Task Learning (TL) [34]. TR recognizes the task based on the demonstrations, *e.g.*, recognizing the data distribution of the task, and applying pre-trained priors of LLM. While TL focuses on learning the correct input-output mapping from the demonstrations, which can be regarded as an implicit learning process analogous to explicit fine-tuning [7].

In this paper, we further refine this hypothesis, providing a more detailed interpretation within the VL realm. Specifically, we further decouple TR into three aspects: format TR, visual TR, and linguistic TR, as shown in Fig. 1. Format TR pertains to the capacity of the LVLM to identify the task format, input distribution, and label space based on demonstrations. For example, in Fig. 1 (c), the question-answer format of the demonstration helps the model determine that the potential answer should be a single word or a simple phrase rather than a complete sentence.

Visual and linguistic TR correspond to the recall of corresponding pre-trained knowledge preserved in the LVLM. As Fig. 1 (a) shows, visual TR uses a visual encoder to identify relevant labels associated with the query image, including the appeared entities, colors, relationships, and more. Linguistic TR (Fig. 1 (b)) recognizes the query question (e.g., "What color is the dog?") through the language component. Drawing from pre-training experience, the model delimits the potential answer space, indicating that only labels related to colors are admissible as the answers. By combining three TR abilities, the label space can be narrowed down for LVLM to make better prediction.

On the other hand, TL refers to the ability of the LVLM model to learn the mapping relationship between (image, question) pairs and their corresponding answers from the demonstrations. As shown in Fig. 1 (d), TL treats the questions and ground truth answers from demonstrations as “training samples”, from which it learns the mapping. Then if the LVLM can successfully achieve TL, it can directly map the query into the correct answer.

3.2. Retrieving Demonstrations

Recognizing that each component (*e.g.*, image, question, and answer) of a VQA sample can be used as an index, we can respectively use them to retrieve n examples from the supporting set $\mathcal{D} = \{(\boldsymbol{I}_1, \boldsymbol{Q}_1, \boldsymbol{A}_1), ..., (\boldsymbol{I}_N, \boldsymbol{Q}_N, \boldsymbol{A}_N))\}$ as the demonstrations for n-shot setting. After that, we can sequence the n-shot triplets to construct the in-context sequence $\mathcal{S}$. Next we introduce specific retrieval strategies.

Random Sampling (RS) (Fig. 2 (a)). We obey the uniform distribution to randomly sample n-shot triplets from $\mathcal{D}$.

Retrieving via Similar Image (SI) (Fig. 2 (b)). We retrieve n images from $\mathcal{D}$ which are most similar to the query image and then use the corresponding triplets of these retrieved images as the demonstrations. For example, given the query sample $(\hat{\boldsymbol{I}}, \hat{\boldsymbol{Q}})$, suppose the i-th image $\boldsymbol{I}_i$ is similar to $\hat{\boldsymbol{I}}$, then the whole i-th triplet $(\boldsymbol{I}_i, \boldsymbol{Q}_i, \boldsymbol{A}_i)$ will be used as one demonstration. Here we use the CLIP embeddings of the images to calculate the cosine similarity.

Retrieving via Similar Texts. Besides retrieving via images, we can also retrieve n triplets which contain most similar texts to the query sample, where the CLIP embeddings of these texts are used to calculate the cosine similarity. We consider three kinds of texts.

(1) **Retrieving via Similar Questions (SQ)** (Fig. 2 (c)). We use the question as the text for retrieving, *i.e.*, comparing the similarity between $\hat{\boldsymbol{Q}}$ and each $\boldsymbol{Q}_i \in \mathcal{D}$.

(2) **Retrieving via Similar Question&Answer (SQA)** (Fig. 2 (d)). We concatenate the question and answer into a text sequence for retrieving, *i.e.*, comparing the similarity between $(\hat{\boldsymbol{Q}}, \hat{\boldsymbol{A}})$ and each $(\boldsymbol{Q}_i, \boldsymbol{A}_i) \in \mathcal{D}$. Although this strategy cannot be applied in practice since we do not have the ground-truth answer of the query sample, it can give us an “upper-bound result” of diverse retrieval methods that can help us better analyze other retrieval strategies.

(3) **Retrieving via Similar Question&Pseudo Answer (SQPA)** (Fig. 2 (e)). Since the ground truth answer is not available during inference, we cannot implement SQA in practice. To exploit the knowledge of the answers in $\mathcal{D}$, we generate the pseudo answer $\hat{\boldsymbol{A}}_i^P$ and then concatenate it with $\hat{\boldsymbol{Q}}$ for retrieving. To get $\hat{\boldsymbol{A}}_i^P$, we can apply the ICL with the demonstrations retrieved by RS and SI.

3.3. Manipulating Demonstrations

Mismatching the Triplet. To explore whether the correctness of demonstrations affects results, we implement mismatched configurations for the image, answer, and question-answer pair in each demonstration. The following $\tilde{\boldsymbol{I}}$, $\tilde{\boldsymbol{Q}}$, $\tilde{\boldsymbol{A}}$ respectively denotes the mismatched images, questions, and answers.

(1) **Mismatching Image (MI)**. We replace the image with a random one from from $\mathcal{D}$. Consequently, $\mathcal{S}$ is transformed to $\{(\tilde{\boldsymbol{I}}_1, \boldsymbol{Q}_1, \boldsymbol{A}_1); ...; (\tilde{\boldsymbol{I}}_n, \boldsymbol{Q}_n, \boldsymbol{A}_n); (\hat{\boldsymbol{I}}, \hat{\boldsymbol{Q}})\}$.

(2) **Mismatching Answer (MA)**. We replace the answer with a random answer in the same label space. $\mathcal{S}$ is transformed to $\{(\boldsymbol{I}_1, \boldsymbol{Q}_1, \tilde{\boldsymbol{A}}_1); ...; (\boldsymbol{I}_n, \boldsymbol{Q}_n, \tilde{\boldsymbol{A}}_n); (\hat{\boldsymbol{I}}, \hat{\boldsymbol{Q}})\}$.

(3) **Mismatching Question-Answer pair (MQA)**. We replace the question-answer pair with a random pair from $\mathcal{D}$. $\mathcal{S}' = \{(\boldsymbol{I}_1, \tilde{\boldsymbol{Q}}_1, \tilde{\boldsymbol{A}}_1); ...; (\boldsymbol{I}_n, \tilde{\boldsymbol{Q}}_n, \tilde{\boldsymbol{A}}_n); (\hat{\boldsymbol{I}}, \hat{\boldsymbol{Q}})\}$.

Reordering in Another Modality. We reorder the demonstrations based on the similarity of another modality, ensuring that the final sequence is visually and linguistically similar to the query sample.

(1) **Reordering SI demonstrations via question similarity (SI-Q)**. We use SI to retrieve the demonstrations, and then reorder these demonstrations based on the similarity between the question of each demonstration and $\hat{\boldsymbol{Q}}$.

(2) **Reordering SQ demonstrations via image similarity (SQ-I)**. Similar to SQ-I, we start with SQ to get the initial demonstrations based on linguistically relevance, then reorder the demonstrations by image-lead similarity.

Using Instructions. To investigate how the model behaves when given a specific instruction, we add an instruction at the beginning of the in-context sequence $\mathcal{S}' = \{\boldsymbol{Inst}; (\boldsymbol{I}_1, \boldsymbol{Q}_1, \boldsymbol{A}_1); ...; (\boldsymbol{I}_n, \boldsymbol{Q}_n, \boldsymbol{A}_n); (\hat{\boldsymbol{I}}, \hat{\boldsymbol{Q}})\}$, where $\boldsymbol{Inst}$ denotes the instruction. Besides using instructions written by humans, we utilize instructions prompted from GPT-4 to further guide the LVLM.

4. Experiments

4.1. Datasets and Implementation Details

We utilize three VQA datasets: VQAv2 [12], VizWiz [3], and OK-VQA [32]. The VQAv2 dataset consists of images from the MSCOCO dataset [24], with more conventional questions. The VizWiz dataset contains low-quality images and questions, and it includes a significant number of unanswerable questions. The OK-VQA dataset requires external knowledge to answer the questions. In each VQA dataset, we use the training set as our supporting set for the experiments, and the validation set serves as our query set. We employ the Open-Flamingo v1(OFv1, the first version of OF) and v2(OFv2, the second version of OF) as the LVLM to evaluate the strategies of demonstration configurations. During retriving, to calculate the embedding similarity, we use the ViT-B/32 model as the vision encoder and a 12-layer Transformer from a well-trained CLIP model [39] as the language encoder to extract image and sentence embeddings, respectively. In ICL, we use 4, 8, 16-shot demonstrations. All experiments are conducted on the RTX 3090 GPU with FP16 precision.

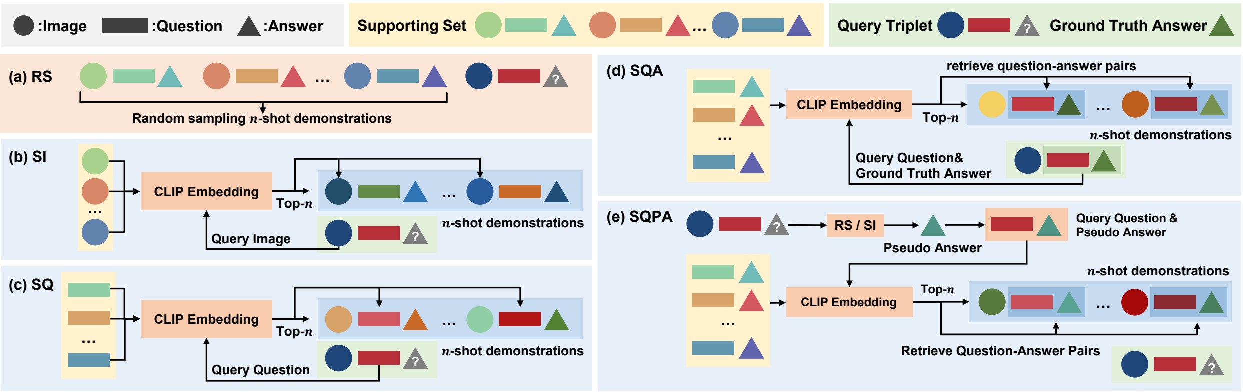


Figure 2. The schematic representation of the demonstrations retrieval strategies. Circles, rectangles, and triangles respectively represent the images, questions, and answers in triplets. The color proximity between these elements indicates their similarity level.

4.2. Results and Analyses

Since exhaustive retrieval and manipulation techniques are applied to configure in-context sequence, presenting all these results could lead to disarray. To avoid confusions and emphasize the major conclusions, we show the corresponding experiment results of each claim in the following and list all the results in the Supplementary Material. Next, in Section 4.2.1, we first show the inner properties of the applied LVLM, Open-Flamingo (OF), which are concluded from the experiment observations. In this part, we will especially show the limitations of OF that will harm the ICL performance of VQA. Then in Section 4.2.2, we will show which configuration strategies can be used to alleviate these limitations for improving the performance.

4.2.1 The Properties of Open-Flamingo

In our demonstration configuration experiments and a series of auxiliary experiments, we observe three main properties of OF. Although these properties are specifically observed in OF, which is currently the most suitable LVLM for ICL, the methods used to observe these properties can be applied to all LVLMs since that these conclusions are also validated by partial complementary experiments on the IDEFICS. These properties provide new perspectives for interpreting and evaluating the ICL capabilities of LVLMs.

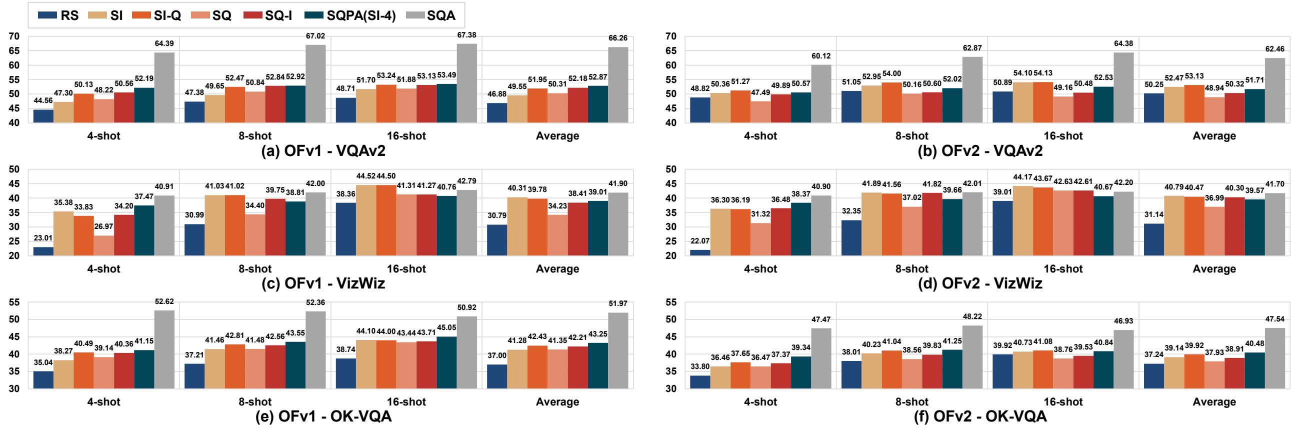
Task Recognition (TR) is More Crucial than Task Learning (TL). This is the first property about OF supported by two experiment observations. Firstly, from Fig. 3 we observe that when shot number increases, the accuracy does not consistently increase. For instance, in VQAv2, expanding the shot count from 8 to 16 offers a modest accuracy increase of 1.33 points, compared to a 2.82 point rise from 4-shot to 8-shot(OFv1-RS). This suggests that TR outperforms TL in OF, aligning with prior findings that format TR does not significantly benefit from additional shots [34]. This is because TR focuses on the label space, format, and input distribution, which means that more shots bring negligible benefits. However, TL aims at learning input-output mappings. Then given more demonstrations the model can better grasp the mapping relationships and thus enhancing TL performance.

	VQAv2		VizWiz		OK-VQA	
	OFv1	OFv2	OFv1	OFv2	OFv1	OFv2
RS	45.97	49.94	27.00	27.21	36.13	36.68
RS(MI)	45.13	49.73	26.92	26.92	35.96	36.16
RS(MA)	45.65	48.94	12.77	11.51	35.56	29.88
SI	48.48	51.66	38.21	39.10	39.87	38.35
SI(MQA)	47.76	49.94	27.57	26.73	37.37	35.82
SI(MA)	47.64	50.40	13.11	11.48	34.70	29.55
SQ	49.53	48.83	30.69	34.17	40.31	37.52
SQ(MI)	48.01	46.21	30.57	31.95	38.38	33.25
SQ(MQA)	45.32	48.98	27.19	26.45	36.71	35.52
SQ(MA)	47.18	42.72	14.82	15.42	29.50	20.50
SQA	65.71	61.50	41.46	41.46	52.49	47.85
SQA(MI)	65.52	60.88	41.09	40.66	51.31	46.88
SQA(MQ)	50.62	60.62	40.31	40.16	40.25	33.04

Table 1. Average results over 4&8-shot of mismatching triplets.

Secondly, Table 1 presents the results of using mismatching triplets. When the disturbed triplets are used, anti-intuitively, the VQA performance does not significantly degrade, *e.g.*, even when all input-output mappings are disturbed, the RS accuracy on VQAv2 only decreases by less than 1 point. Such phenomenon can be explained from the perspectives of TR and TL. Specifically, TR focus on recognizing the question format, input distribution, and label space from the demonstrations, which can be provided from the disturbed demonstrations as the non-disturbed ones. TL needs to capture the correct input-out mapping while the disturbed QA pairs will damage this. Therefore, TR is less affected by disturbances compared to TL. As the ICL performance is less affected by disturbances, we can conclude that TR plays is more crucial than than TL. In Table 2, mismatched images(RS(MI))/answers(RS(MA)) have minimal impact, also indicating the dominance of TR in IDEFICS.

Besides these two experiment observations, we conduct



(a) OFv1 - VQAv2

(b) OFv2 - VQAv2

(c) OFv1 - VizWiz

(d) OFv2 - VizWiz

(e) OFv1 - OK-VQA

(f) OFv2 - OK-VQA

Figure 3. Experimental results of different demonstrations retrieval strategies on OFv1 and OFv2, comparing 4-shot, 8-shot, 16-shot, and average results across these configurations. SQPA(SI-4) refers to using the result of 4-shot SI as the pseudo answer.

further validation following the approach proposed in [34]. This approach disentangles TR and TL by utilizing different demonstration settings to reflect their respective capabilities. Details and full results of our experiments can be found in the supplementary material. We discover that the accuracy of TR is significantly higher than TL and it is comparable to the results obtained from the standard ICL. Such observation further confirms that TR plays a dominant role in ICL. Additionally, we observe that with more data pretraining and an improved language backbone, the TL capability of OFv2 significantly increases compared to OFv1. This indicates that increasing the amount of pre-training can enhance the TL capability of the model.

Short-cut Inference. In NLP, using similar text demonstrations often enhances performance. However, our experiments show that similar demonstrations do not always improve results and can sometimes damage them. For instance, in Fig. 3, on VQAv2, using demonstrations with similar questions (SQ) performs worse than randomly sampled ones (RS), *e.g.*, 50.25 vs. 48.94.

We believe this happens because *OF tends to build short-cut for predicting*. After analyzing, we find that when the demonstration has a question similar to the query, OF often copies the answer from the demonstration with the similar question instead of using visual information, thus building a short-cut. For example, in Fig. 4 (b1), when asked about bed sheet patterns, SQ incorrectly returns “alligator and bear” from the demonstration with a similar question, even though it does not match the query image.

Besides the qualitative observations, we also quantitatively measure the short-cut effect. In Tab. 3 and Tab. 2, we compute the probability that predicted answers also appear in the demonstrations. For SQ, OFv1/OFv2/IDEFICS exhibit copy rates of 77.26%/79.84%/75.34%, respectively, while SQA further increases the copy rates to 87.74%/89.47%/87.32%, while RS and SI achieve only 43.64%/37.34%/44.1% and 50.44%/54.38%/55.66%.

	RS	SI	SQ	SI-Q	SQA	RS(MI)	RS(MA)	RS w/o instruction
IDEFICS	53.70	54.23	52.34	55.84	64.05	52.29	52.85	52.22

Table 2. Average results over 4&8-shot of IDEFICS on VQAv2 and the copy rate (16 shot). We follow IDEFICS[15] to add instruction and “w/o instruction” denotes not using the instruction.

	RS	SI	SQ	SQA	SQA(sole)	SQA(sole wrong)
OFv1	43.64	50.44	77.26	87.74	47.39	37.07
OFv2	37.34	54.38	79.84	89.47	45.82	45.71
IDEFICS	44.10	55.66	75.34	87.32	52.70	39.66

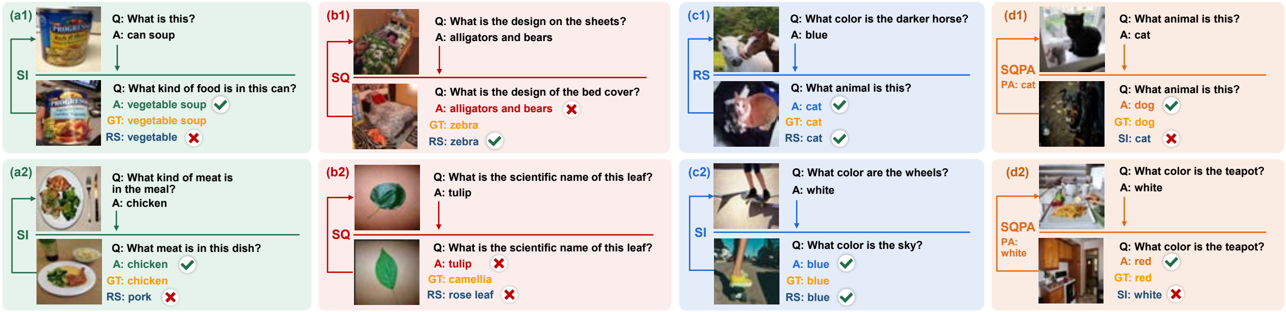
Table 3. The copy rate (%) of short-cut on VQAv2 (16-shot).

Moreover, we conduct an experiment following [31] that using demonstrations with identical test inputs and correct or incorrect labels. OF predicts the same answer as the identical input in 47.39%/45.82% of cases with correct labels and 37.07%/45.71% of cases with incorrect labels, suggesting that even when there is only one question similar to the query in the demonstration, it can still trigger more severe short-cut inference.

This short-cut effect, prevalent in NLP [31] and Image Captioning [45], its influence beyond LLMs to also impact LVLMs across various tasks. One possible reason is that these models have limited TL ability and are influenced by biases instead of learning from the demonstrations.

Image and Language Decoders are not totally Compatible. This is the third conclusion about OF and can be demonstrated in two aspects. First, *the language encoder is much stronger than the vision encoder, which causing that linguistic TR plays a more substantial role than visual TR in VQA*. Second, *the vision and language modules are not aligned well, causing some language reasoning ability lose efficacy in the VL case*.

For the first aspect, OF shows heightened sensitivity to text quality, with compromised textual input leading to a more significant decline in performance. Results in Tab. 1 demonstrate that replacing the answer in SQ causes a significant 5-point drop, while replacing the image only leads to



(a) SI

(b) Short-cut

(c) Mismatch

(d) SQPA

Figure 4. Four in-context learning scenarios. The arrow on the left signifies the retrieval strategy, between a retrieved demonstration (top) and the query sample (bottom). (a) **SI** shows effective TL through similar image retrieval, such as learning "soup can" in (a1). (b) **Short-cut** illustrates inference errors where the model replicates the incorrect answer "tulip" from a similar question in (b2). (c) **Mismatch** highlights that mismatched answers within the same label space do not significantly impact model effectiveness, with the TR correctly identifying the answer as "cat" in (c1). (d) **SQPA** reveals how the model corrects a wrong answer ("white") to "red" by using pseudo answers for learning the input-output mapping in (d2).

a minor 1-point decrease. Similarly, substituting the question of SQA (MQ) lead to a notable decline of 15 points, whereas replacing the image (MI) did not cause a significant decrease. Moreover, replacing the text with noisy text leads to a more significant performance decline compared to replacing the image with a blurred image (full results in supplementary material). These findings validate that linguistic TR plays a more substantial role than visual TR, potentially due to the greater power and scale of language module compared to the visual module in OF, *e.g.*, the language module is LLaMA/MPT, which containing 7 billion parameters and pre-trained on one trillion tokens, while the visual module is CLIP ViT/L-14, containing 428 million parameters and pre-trained on 400 million data. This indicates that in VLMs, language and vision do not play equally important roles. Instead, linguistic TR demonstrates greater potency and exerts a stronger influence on overall performance.

For the second aspect, we find that some useful strategies for solving QA lose their efficacy in OF. For instance, reformulating a QA pair into a declarative sentence to better adapt to the pre-training language model and changing the orders of demonstrations, known to improve performance in NLP, fails to have the same effect on LVLM and detailed experimental descriptions and complete results will be presented in the supplementary materials. Additionally, adding instructions before the in-context sequence, which is effective in NLP, only works in OFv2 and not OFv1. Before we think that the language module of OF is stronger than the vision module, then why some useful NLP strategies lose the efficacy? We think the major reason is that the vision and language modules are not aligned well, *i.e.*, the language reasoning of the original LLM does not totally inherited into the VLM after vision-language alignment finetuning. Such assumption can be supported from the comparison between OFv1 and OFv2. Compared with OFv1, OFv2 uses more image-text pairs for aligning vision and language modules(180M vs.15M pairs) and thus can inherit more language reasoning ability for solving VL task, and thus we find that adding instructions works better in OFv2 than OFv1, where more details are given in Section 4.2.2.

4.2.2 Effective Configuration Strategies

Although OF is one of the SOTA LVLMs for ICL, in section 4.2.1, we observe that it has three major limitations: weak TL capabilities, the short-cut effect, and not totally compatible vision-language modules. However, in this section, we still observe that some strategies can improve the ICL ability for VQA.

Similar images and texts lead to better performance. Despite we previously show that using demonstrations with similar questions leads to short-cut inference, we now present evidence that using demonstrations that simultaneously contain similar images and questions can enhance performance. Although the improvements vary depending on the dataset, such strategy is still a powerful way to improve the performance.

First, as Fig. 3 and Tab. 2 shows, using the demonstrations with similar image (SI) consistently boosts LVLM performance, *e.g.*, on VQAv2/VizWiz/OK-VQA/(OFv2, 4-shot), we observe 1.54/14.23/2.66 point improvements. We assume that more similar images in the in-context sequence can compensate more visual information that may have been missed or incorrectly recognized during the visual TR stage. For instance, in Fig. 4, while both RS and SQ could only recognize the term "vegetable" for an image of a soup can, SI identify it as a "Progresso vegetable soup" since one in-context image also has this soup can. Such visual compensation works more obvious on VizWiz since the image quality of this dataset is quite low and using similar images help OF pinpoint a more accurate label space, *i.e.*, enhancing the visual TR ability.

	Dataset	4-shot	8-shot	16-shot
RS(OFv1)	VQAv2	44.56	47.38	48.71
Instruct1(OFv1)	VQAv2	43.75	46.91	48.67
RS	VQAv2	48.82	51.05	50.89
Instruct1	VQAv2	**49.93**	**52.71**	**50.95**
RS	VizWiz	22.07	32.35	39.01
Instruct1	VizWiz	**25.70**	**34.71**	**39.32**
RS	OK-VQA	34.82	38.54	39.55
Instruct1	OK-VQA	35.72	39.38	40.46
Instruct2	OK-VQA	**36.45**	40.17	**41.11**
Instruct3	OK-VQA	35.53	**40.19**	40.02

Instruct1: According to the previous question and answer pair, answer the final question.
Instruct2: Consider the semantic relationship between the question and the image.
Instruct3: You will be engaged in a two-phase task. Phase 1: Absorb the information from a series of image-text pairs. Phase 2: Use that context, combined with an upcoming image and your own database of knowledge, to accurately answer a subsequent question.

Table 4. The results of using instructions.

Secondly, the SQ approach also brings improvements, although these enhancements are not consistently stable due to the presence of the short-cut. However, as shown in Fig. 3, for the VizWiz dataset with lower-quality text and the OK-VQA dataset requiring additional knowledge, demonstrations containing similar questions and reference answers still assist the model in finding the correct answers.

Thirdly, in section 3.3, we show how to reorder the retrieved demonstrations based on their similarity in another modality, *i.e.*, it retrieves similar images/questions and rearranges them based on the similarity of their associated questions/images. As Fig. 3 shows, when applied to two versions of OF and across three varied datasets, this method consistently showcased superiority over base methods. Such findings suggest that both visually and textually similar in-context examples can greatly enhance the performance of LVLMs in TL.

Instruction enhances the performance of linguistically advanced model. Providing instructions notably enhances the format TR and TL capabilities of the LVLM. As evident in Tab. 4, the OFv2 model exhibits substantial improvements across various datasets when using instructions, especially in limited demonstration scenarios. For instance, adding instructions to the 4-shot experiment on VizWiz results in a 3.63 points increment. Given the necessity for additional knowledge in VQA tasks on OK-VQA, we utilize GPT-4 to design two types of instructions: concise and straightforward instructions (Instruct2 in Tab. 4) and detailed, hierarchical instructions (Instruct3 in Tab. 4). Providing instructions enhances the format TR and TL capabilities of LVLMs by increasing information density in demonstrations, akin to providing more demonstrations. Compared to additional demonstrations, it saves selection time and reduces the processing burden on the visual encoder of LVLM, making it simpler and more convenient. However, the instructions do not yield significant improvements in experiments with the OFv1 model due to the inferior language encoder of the v1 model, impacting its capability to process these instructions.

Pseudo answers have potential for expeditious enhancement of performance. From the results in Tab. 5, we can observe that at 4-shot, SQPA generally improves performance. Intuitively, as shown in Fig. 4, when the first-round model generates an incorrect answer ("cat"), the demonstration obtained through SQA retrieval using the question and the erroneous answer will be dissimilar to the content of the query image (which is actually a dog). This provides the model with an opportunity to discover that "cat" is not the correct answer and to reason and infer a new answer. Therefore, the accuracy of the second-round model using SQPA is expected to surpass that of the first round. However, as the number of shots increases, only on OK-VQA does SQPA still show improvement. This may be because too many incorrect QA pairs interfere with the reasoning process of the model, while OK-VQA requires additional knowledge. By using pseudo-answers to search, the model may be able to find more related knowledge.

	VQAv2			VizWiz			OK-VQA		
	4-shot	8-shot	16-shot	4-shot	8-shot	16-shot	4-shot	8-shot	16-shot
RS	48.82	51.05	50.89	22.07	32.35	39.01	34.82	38.54	39.55
SQPA(RS-4)	49.85	51.03	51.96	30.02	31.93	34.25	38.92	41.16	40.06
SI	50.36	**52.95**	**54.10**	36.30	**41.89**	**44.17**	36.46	40.23	40.73
SQPA(SI-4)	**50.57**	52.02	52.53	**38.37**	39.66	40.67	**39.34**	**41.25**	**40.84**

Table 5. The results of SQPA on OFv2. SQPA(RS/SI-4) refers to using the result of 4-shot RS/SI as the pseudo answer.

5. Conclusion

In this paper, our focus is to investigate the diverse in-context configurations and delve into the inner properties of LVLMs using VQA as a case study. We design various methods to retrieve and manipulate in-context samples. Through exhaustive experiments, we uncover three important inner properties of the applied LVLMs. Furthermore, we identify the strategies that consistently enhance the performance of ICL VQA. These findings contribute to a deeper understanding of LVLMs and provide valuable insights for optimizing their ICL performance in VQA.

In the future, we plan to validate the effectiveness of our proposed demonstration configuration strategies on a wider range of LVLMs. Additionally, we will analyze and evaluate the capabilities of more LVLMs from the perspectives of the three properties observed in Open-Flamingo and IDEFICS in Sec. 4.2.1.

Acknowledgments

This work is supported by National Science Foundation of China (62206048), Natural Science Foundation of Jiangsu Province (BK20220819), Young Elite Scientists Sponsorship Program of Jiangsu Association for Science and Technology Tj-2022-027 and the Big Data Computing Center of Southeast University.

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. 1, 2, 3

[2] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 1, 3

[3] Jeffrey P Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342, 2010. 3, 4

[4] 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. 2

[5] 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

[6] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020. 2

[7] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023. 3

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 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. 1

[10] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020. 2

[11] 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

[12] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 3, 4

[13] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8: 423–438, 2020. 2

[14] 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. 1, 3

[15] Hugo Laurenccon, Lucile Saulnier, L'eo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander M. Rush, Douwe Kiela, Matthieu Cord, and Victor Sanh. Obelisc: An open web-scale filtered dataset of interleaved image-text documents. *ArXiv*, abs/2306.16527, 2023. 3, 6

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

[17] Itay Levy, Ben Bogin, and Jonathan Berant. Diverse demonstrations improve in-context compositional generalization. *arXiv preprint arXiv:2212.06800*, 2022. 2

[18] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv preprint arXiv:2305.03726*, 2023. 2

[19] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, Hehong Chen, Guohai Xu, Zheng da Cao, Ji Zhang, Songfang Huang, Feiran Huang, Jingren Zhou, and Luo Si. mplug: Effective and efficient vision-language learning by cross-modal skip-connections. In *Conference on Empirical Methods in Natural Language Processing*, 2022. 2

[20] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 2

[21] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 2

[22] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020. 2

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

[24] 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. 4

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023. 2

[26] 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. 1, 3

[27] 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. 1, 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. 2

[29] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019. 2

[30] 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. 1, 2, 3

[31] Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. Z-icl: Zero-shot in-context learning with pseudo-demonstrations. *arXiv preprint arXiv:2212.09865*, 2022. 3, 6

[32] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019. 3, 4

[33] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022. 1, 3

[34] Jane Pan. *What In-Context Learning "Learns" In-Context: Disentangling Task Recognition and Task Learning*. PhD thesis, Princeton University, 2023. 1, 3, 5, 6

[35] Chengwei Qin, Wenhan Xia, Fangkai Jiao, and Shafiq R. Joty. Improving in-context learning via bidirectional alignment. *ArXiv*, abs/2312.17055, 2023. 2

[36] Chengwei Qin, Aston Zhang, Anirudh Dagar, and Wenming Ye. In-context learning with iterative demonstration selection. *ArXiv*, abs/2310.09881, 2023. 2

[37] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[38] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1

[39] 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, 4

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

[41] Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*, 2022. 2

[42] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022. 2

[43] 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

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

[45] Xu Yang, Yongliang Wu, Mingzhuo Yang, Haokun Chen, and Geng Xin. Exploring diverse in-context configurations for image captioning. *arXiv preprint arXiv:2305.14800*, 2023. 1, 3, 6

[46] Yang Yang, De-Chuan Zhan, Yi-Feng Wu, Zhi-Bin Liu, Hui Xiong, and Yuan Jiang. Semi-supervised multi-modal clustering and classification with incomplete modalities. *IEEE Trans. Knowl. Data Eng.*, 33(2):682–695, 2021. 1

[47] Yang Yang, Hongchen Wei, Hengshu Zhu, Dianhai Yu, Hui Xiong, and Jian Yang. Exploiting cross-modal prediction and relation consistency for semisupervised image captioning. *IEEE Transactions on Cybernetics*, 54(2):890–902, 2024. 1

[48] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 2

[49] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 2

[50] Yiming Zhang, Shi Feng, and Chenhao Tan. Active example selection for in-context learning. *arXiv preprint arXiv:2211.04486*, 2022. 1

[51] Haozhe Zhao, Zefan Cai, Shuzheng Si, Xiaojian Ma, Kaikai An, Liang Chen, Zixuan Liu, Sheng Wang, Wenjuan Han, and Baobao Chang. Mmicl: Empowering vision-language model with multi-modal in-context learning. *arXiv preprint arXiv:2309.07915*, 2023. 3

[52] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language

understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2