RESEARCH-ARTICLE

Enhancing Multimodal In-Context Learning for Image Classification through Coreset Optimization

HUIYI CHEN, Southeast University, Nanjing, Jiangsu, China

JIAWEI PENG, Southeast University, Nanjing, Jiangsu, China

KAIHUA TANG, Huawei Technologies Co., Ltd., Shenzhen, Guangdong, China

XIN GENG, Southeast University, Nanjing, Jiangsu, China

XU YANG, Southeast University, Nanjing, Jiangsu, China

Published: 27 October 2025

Citation in BibTeX format

MM '25: The 33rd ACM International Conference on Multimedia
October 27 - 31, 2025
Dublin, Ireland

Conference Sponsors:
SIGMM

Enhancing Multimodal In-Context Learning for Image Classification through Coreset Optimization

Huiyi Chen
Southeast University
School of Computer Science and Engineering
Nanjing, China
huiyichen@seu.edu.cn

Jiawei Peng
Southeast University
School of Computer Science and Engineering
Nanjing, China
pengjiawei@seu.edu.cn

Kaihua Tang
Huawei Singapore Research Center
Singapore, Singapore
tkhchipaomian@gmail.com

Xin Geng
Southeast University
School of Computer Science and Engineering
Nanjing, China
xgeng@seu.edu.cn

Xu Yang*
Southeast University
School of Computer Science and Engineering
Nanjing, China
xuyang_palm@seu.edu.cn

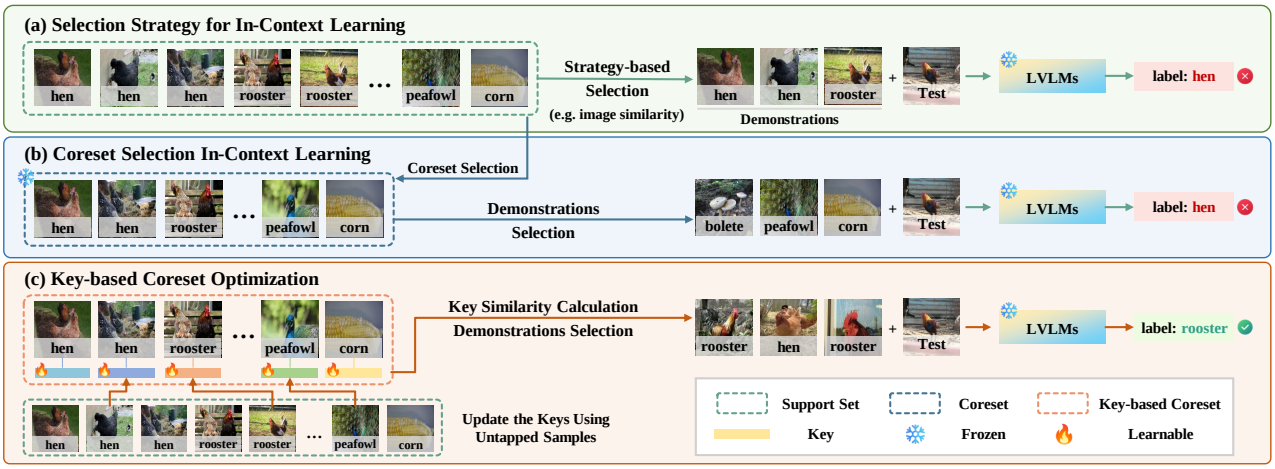

Figure 1: (a) Strategy-Based Selection In-Context Learning: requires storing the full support set and computing similarities between each test input and all support samples before inference. (b) Coreset Selection In-Context Learning: uses a coreset selection strategy to explore an informative subset from the entire support set, reducing the cost of selecting demonstrations. (c) Key-based Coreset Optimization: the untapped samples in the support set are used to update the coreset. Specifically, this involves updating the key of each sample, where the key refers to the visual feature of the sample.

Abstract

In-context learning (ICL) enables Large Vision-Language Models (LVLMs) to adapt to new tasks without parameter updates, using a few demonstrations from a large support set. However, selecting informative demonstrations leads to high computational and memory costs. While some methods explore selecting a small and representative coreset in the text classification, evaluating all support set samples remains costly, and discarded samples lead to unnecessary information loss. These methods may also be less effective for image classification due to differences in feature spaces. Given these limitations, we propose Key-based Coreset Optimization (KeCO), a novel framework that leverages untapped data to construct a compact and informative coreset. We introduce visual features as keys within the coreset, which serve as the anchor for identifying samples to be updated through different selection strategies. By leveraging untapped samples from the support set, we update the keys of selected coreset samples, enabling the randomly initialized

*Corresponding author.

MM '25, Dublin, Ireland

ACM ISBN 979-8-4007-2035-2/2025/10
https://doi.org/10.1145/3746027.3755764

coreset to evolve into a more informative coreset under low computational cost. Through extensive experiments on coarse-grained and fine-grained image classification benchmarks, we demonstrate that KeCO effectively enhances ICL performance for image classification task, achieving an average improvement of more than 20%. Notably, we evaluate KeCO under a simulated online scenario, and the strong performance in this scenario highlights the practical value of our framework for resource-constrained real-world scenarios.

CCS Concepts

• **Computing methodologies** → **Knowledge representation and reasoning**; **Computer vision**; **Natural language processing**.

Keywords

In-context Learning; Large Vision-Language Model; Coreset

ACM Reference Format:

Huiyi Chen, Jiawei Peng, Kaihua Tang, Xin Geng, and Xu Yang. 2025. Enhancing Multimodal In-Context Learning for Image Classification through Coreset Optimization. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM '25), October 27–31, 2025, Dublin, Ireland.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746027.3755764

1 Introduction

In-context learning (ICL) has emerged as a powerful paradigm [4, 7] that enables Large Language Models (LLMs) to solve novel tasks by conditioning on a few in-context examples (ICEs), without additional training. This paradigm is particularly appealing in real-world scenarios where fine-tuning is costly or infeasible. Recently, ICL has been extended to the multi-modal setting, where Large Vision-Language Models (LVLMs) leverage a few image-text pairs as ICEs to perform downstream tasks [1, 16, 20, 39].

Prior studies have shown that ICL performance is highly sensitive to the selection of ICEs [3, 5]. Compared to conventional random selection, recent works adopt more effective strategies, such as similarity-based retrieval, to select informative ICEs [23, 32, 33, 50]. However, as illustrated in Figure 1 (a), these methods require storing and traversing the entire support set, resulting in significant computational and memory overhead. To mitigate this issue, recent studies in Natural Language Processing (NLP) have explored *coreset* selection techniques, which aim to identify a small yet representative subset of the support set [26, 38]. For example, [26] use language model feedback to identify samples whose influence on model predictions surpasses that of other samples in the support set. As shown in Figure 1 (b), this approach substantially reduces retrieval costs while maintaining comparable ICL performance.

Despite these advances, coreset-based strategies still face several limitations in ICL. First, identifying a coreset typically involves complex selection mechanisms and time-consuming evaluations of all support set samples. Second, applying sample-level coreset selection to image classification is inherently challenging. Unlike textual data, which is often composed of compact discrete tokens, visual inputs are inherently richer and more diverse [11]. For example, images of a 'hen' may vary significantly in pose, background, and other category-irrelevant factors, making it harder to adequately represent the underlying feature space within a fixed small subset. Third, fixed coresets inevitably discard a large portion of the support set. Although these samples are evaluated during selection, the information they contain remains unused in downstream inference, leading to unnecessary information loss.

Given these limitations, we propose a novel coreset construction framework for image classification tasks, termed **Ke**y-based **C**oreset **O**ptimization (KeCO). Instead of relying solely on sample selection, KeCO leverages untapped data to update coreset representations at the feature level, yielding a compact yet informative coreset for effective ICL with LVLMs. Specifically, KeCO consists of three steps: (1) we begin by randomly sampling an initial coreset from the support set. Although simple, this initialization proves effective when followed by our feature-level refinement procedure. We extract visual features from coreset samples as *keys*, which serve as anchors for subsequent selection and update processes. (2) The remaining untapped samples are then used to update the coreset. For each incoming query sample from the untapped set, we retrieve a target sample from the coreset to be updated. We investigate various selection strategies (e.g., diversity-based selection) at this step to examine their impact on coreset optimization. (3) Once a target sample is selected, its key is updated by incorporating information from the query sample via linear interpolation.

After iteratively running steps (2) and (3) to refine the coreset with untapped samples, we obtain a compact coreset that significantly enhances ICL performance. To further demonstrate its practicality, we extend KeCO to a **simulated online scenario**, where samples are no longer available all at once but instead arrive in a streaming fashion. KeCO naturally adapts to this setting with minimal adjustments, enabling LVLMs to continuously benefit from incoming samples under constrained memory resources.

We evaluated KeCO on three datasets spanning both coarse-grained and fine-grained image classification datasets. Compared to the fixed coreset, KeCO improves the ICL performance by 20% for OpenFlamingo-3B (OF-3B) [2] and 10% for IDEFICS-8B (IDE-8B) [16]. Remarkably, KeCO even outperforms the ICL using the **full support set**, which is five times larger than the coreset, by approximately 10% for OF-3B and 4% for IDE-8B. We further demonstrate that target sample selection plays a crucial role in our framework, with diversity-based selection yielding the best results. In addition, KeCO achieves larger gains on fine-grained datasets, highlighting its effectiveness in capturing subtle visual distinctions. In the simulated online scenario, KeCO maintains strong and stable performance. For instance, it enables Qwen2-VL [40] to exceed the baseline by approximately 5% on CUB-200. Our main contributions are summarized as follows:

- We propose a novel coreset optimization framework, **KeCO**, which constructs a compact and effective coreset from a large support set. To the best of our knowledge, this is the first attempt to introduce coreset optimization into the ICL paradigm.
- KeCO leverages untapped data to aggregate category-relevent information into the coreset via feature-level updates. Notably, KeCO achieves strong performance in a simulated online scenario, demonstrating its practical applicability.
- Extensive experiments show that KeCO significantly boosts ICL performance for image classification task. We further identify

that diversity-based selection outperforms other target sample selection strategies, shedding light on better update mechanisms.

2 Related Work

2.1 Large Vision-Language Model

Large Vision-Language Models (LVLMs) [27] are generally built on a vision encoder, a pre-trained Large Language Model (LLM), and an alignment module between the vision encoder and the LLM. The level of performance of these models has started to approach those of LLMs, especially after multimodal instruction tuning [6, 12, 13, 28, 45–47]. Although there are numerous LVLMs available, it is important to note that not all of these models support in-context learning (ICL). For example, mPLUG-Owl [44], BLIP-2 [22] and MiniGPT-4 [51] lack the capabilities for ICL because they have not undergone dedicated few-shot pre-training and cannot handle the input distribution associated with ICL. In contrast, models like Flamingo [1] and IDEFICS [16] are specifically designed to support this task. In this work, we mainly focus on two open-source models with ICL capabilities (OpenFlamingo [2] and IDEFICS [16]) for our main experiments. Additional experiments are also performed on a high-performance commercial model, Qwen2-VL [40].

2.2 Multimodal In-context Learning

ICL enables LLMs to tackle novel tasks by leveraging a few demonstrations from a support set [4, 7]. Building on this success, ICL has been extended to LVLMs [1, 2, 8, 17, 19, 21, 39]. To further enhance ICL in LVLMs, several works have focused on improving the construction strategies of in-context sequences, such as similarity-based selection [25, 30, 32, 33, 43] and diversity-based selection [18, 23]. Despite its effectiveness, it faces challenges in resource-limited scenarios, requiring large storage and significant computational resources for computation [14, 42]. Therefore, in the NLP domain, some works focus on selecting samples from an unlabeled pool for annotation to reduce annotation costs [31, 34], while others attempt to explore a representative coreset for ICL [26, 38]. However, identifying such a coreset requires complex computation strategies and time-consuming evaluations of all samples in the support set. Additionally, a small coreset may not adequately represent the diverse semantics of the dataset in vision domain, and discarded data from the support set remains untapped. To overcome these limitations, we randomly sample a coreset and then integrate feature-level information from the untapped data to enhance the information contained within this coreset.

3 Method

3.1 Preliminaries

The conventional In-context learning (ICL) within Large Vision-Language Model (LVLMs) is to construct an in-context sequence $\mathcal{D}$ by retrieving n samples from a given support set. Specifically, given a support set $\mathcal{S} = \{(\boldsymbol{I}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, where $\boldsymbol{I}_i$ represents an image and $\boldsymbol{y}_i$ is its corresponding label, the in-context sequence is formed as follows:

$$\mathcal{D} = \{(\boldsymbol{I}_1, \boldsymbol{y}_1), (\boldsymbol{I}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{I}_n, \boldsymbol{y}_n), (\hat{\boldsymbol{I}})\}. \tag{1}$$

Here, the first n image-label pairs serve as demonstrations, while $\hat{\boldsymbol{I}}$ is the test image whose label $\hat{y}$ needs to be predicted. The constructed sequence $\mathcal{D}$ is then fed into the LVLM, which generates the predicted label $\hat{y}$ based on the following probability distribution:

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}}\, P(y \mid \mathcal{D}), \tag{2}$$

where Y is the label space.

The selection of in-context examples $\{(\boldsymbol{I}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{I}_n, \boldsymbol{y}_n)\}$ plays a crucial role in the performance of ICL, as it directly affects the contextual understanding and generalization capability of LVLMs. A common approach is similarity-based retrieval, where samples most similar to the test image $\hat{\boldsymbol{I}}$ are selected based on a feature extractor $\phi(\cdot)$, such as CLIP:

$$\text{sim}(\hat{\boldsymbol{I}}, \boldsymbol{I}_i) = \phi(\hat{\boldsymbol{I}})^\top \phi(\boldsymbol{I}_i), \quad \text{where } \boldsymbol{I}_i \in \mathcal{D}. \tag{3}$$

However, this method becomes computationally expensive for a large support set. To improve efficiency, coreset-based methods first select a compact and representative subset $C \subset \mathcal{S}$, then retrieve samples from C instead of the whole support set $\mathcal{S}$. The coreset can be constructed by samples with high diversity or large information gain, ensuring a more representative selection for ICL. However, due to the nature of the feature spaces in NLP and vision domain are fundamentally different, coreset methods in the NLP domain still struggle to successfully apply to the vision domain.

3.2 Key-based Coreset Optimization (KeCO)

To address the aforementioned issues, we propose a straightforward but effective framework to obtain a compact and effective coreset, termed Key-based Coreset Optimization (KeCO), as shown in Figure 2. Given a support set $\mathcal{S}$ of size n, we select a initial subset, referred to as the **Key-based Coreset**, denoted by C, with a size of m, where $m < n$. The untapped samples in $\mathcal{S}$ form the untapped set $\mathcal{S}' = \mathcal{S} - C$. $\mathcal{S}'$ will be used to update C, and the updated coreset is denoted as C'. As shown in Figure 2, this framework consists of three stages:

- **Key-based Coreset Initialization**: A coreset C is first selected from the entire support set $\mathcal{S}$. The visual feature of the samples in C are extracted using a vision encoder, which serves as their corresponding keys for retrieval and updates.
- **Target Sample Selection**: Given a query sample s from untapped set S', a corresponding target sample t from C will be selected using different selection strategies.
- **Key-based Sample Update**: An update strategy is applied to incorporate the information from the untapped query samples into each t, resulting in more informative coreset C'.

In the following subsections, we detail each stage of KeCO.

3.3 Key-based Coreset Initialization

We adopt a simple **Random initialization** strategy to get the initial coreset. Specifically, given a support set $\mathcal{S} = \{(\boldsymbol{I}_i, \boldsymbol{y}_i)\}_{i=1}^{n}$ with j categories, we construct C by randomly selecting m samples while ensuring class balance. That is, for each category $c \in \{1, \ldots, j\}$, we select

$$C = \bigcup_{c=1}^{j} C_c, \quad \text{where} \quad C_c = \{(\boldsymbol{I}_i, \boldsymbol{y}_i) \mid \boldsymbol{y}_i = c,\ (\boldsymbol{I}_i, \boldsymbol{y}_i) \in \mathcal{S}\}, \tag{4}$$

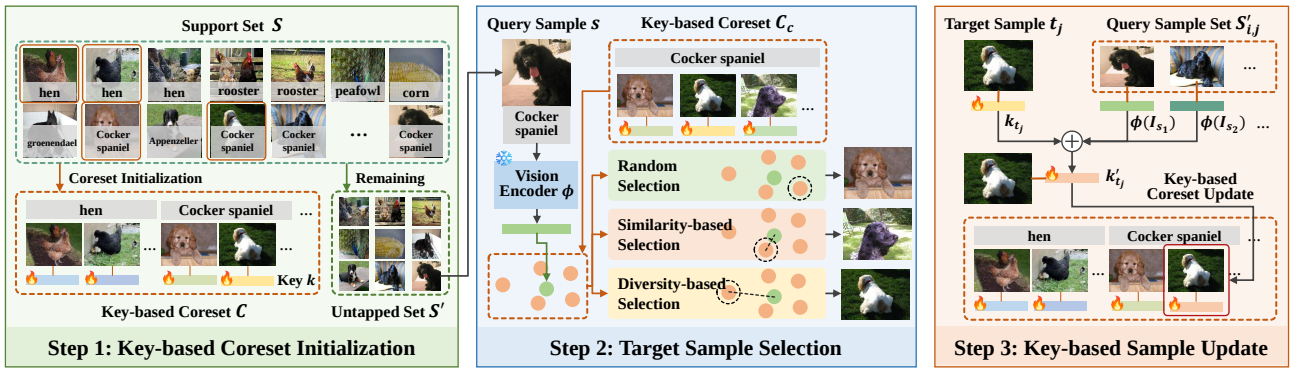

Figure 2: Overview of the KeCO framework: First, a subset is randomly selected from the support set S to initialize the Coreset C, with the remaining samples forming the untapped set S'. A vision encoder is then used to extract visual features for all samples in C, which serve as their *keys*. Next, for each query sample s in the untapped set, its extracted feature will be used to select a corresponding target sample $t \in C$ based on three different strategies (Random Selection, Similarity-based Selection, Diversity Selection). Finally, the key of each t in C will be updated by aggregating the features of all associated query samples.

such that $|C_c| = m/j$, where C_c denotes the subset of C belonging to the category c. For each selected sample, we extract its visual feature $\boldsymbol{k}$ using a vision encoder ϕ, which serves as its **key** and is stored within C. These keys are later used for retrieval and updating in subsequent stages. Thus, each element in C is represented as:

$$C = \{(\boldsymbol{I}_i, \boldsymbol{y}_i, \boldsymbol{k}_i) \mid \boldsymbol{k}_i = \phi(\boldsymbol{I}_i),\ i = 1, \ldots, m\}. \tag{5}$$

Beyond random initialization, we also explore two alternative initialization strategies: **K-center initialization** and **InfoScore initialization**. The K-center approach aims to maximize diversity by selecting samples that best represent the overall distribution of $\mathcal{S}$, while the InfoScore method prioritizes samples with the highest information gain based on LVLM's feedback. Implementation details of these strategies are provided in Appendix A.1.

3.4 Target Sample Selection

After Initialization, we use the untapped set $\mathcal{S}'$ to update C. For each query sample $s = (\boldsymbol{I}_s, c) \in S'$, we need to determine which sample in C could be updated by s. We refer to this sample in C as the **target sample** t. To effectively select the best t for s, we first retrieve all samples in C with the same class label c, denoted as C_c, then explore the following selection strategies to choose the corresponding target sample t from C_c:

- **Random Selection (RS)**: The simplest approach is to randomly select a target sample from C_c. While easy to implement, this method does not prioritize samples that may benefit most from updates.
- **Similarity-based Selection (SS)**: In this strategy, we select the sample that is most similar to s from C_c. To measure similarity, we compute the cosine similarity between feature representations:

$$t = \arg\max_{e \in C_c} \frac{k_e \cdot \phi(\boldsymbol{I}_s)}{\|k_e\| \|\phi(\boldsymbol{I}_s)\|}, \tag{6}$$

  where k_e denotes the key of the sample e in C. This method ensures that the keys in C are updated only by the most similar samples from the newly incoming data.

- **Diversity-based Selection (DS)**: In this strategy, we select the sample that is least similar to s. While this may initially appear counterintuitive, it has been shown to be the most effective strategy as demonstrated and discussed in our Section 4.4:

$$t = \arg\min_{e \in C_c} \frac{k_e \cdot \phi(\boldsymbol{I}_s)}{\|k_e\| \|\phi(\boldsymbol{I}_s)\|}, \tag{7}$$

Once the target sample t is selected, it will undergo the key-based update process described in the next section.

3.5 Key-based Sample Update

We utilize untapped set S' to update the keys in the coreset, dividing it into mini-batches for smooth updates:

$$S' = \bigcup_{i=1}^{(n-m)/b} S'_i, \tag{8}$$

where each batch $S'_i = \{s_{i_1}, s_{i_2}, \ldots, s_{i_b}\}$ contains b samples.

For each sample $s \in S'_i$, we select a corresponding **target sample** $t \in C$ from the coreset, based on the strategy introduced in Section 3.4. Since multiple samples in S'_i may select the same target t_j, we group all such samples together:

$$S'_{i,j} = \{s \in S'_i \mid \text{target}(s) = t_j\}, \tag{9}$$

where target(s) denotes the selected target sample for s. This defines $S'_{i,j}$ as the subset of samples in batch S'_i that share t_j as their target.

Then, we update the key k_{t_j} of t_j using:

$$k'_{t_j} = k_{t_j} - \alpha \cdot \frac{1}{|S'_{i,j}|} \sum_{s \in S'_{i,j}} (k_{t_j} - \phi(\boldsymbol{I}_s)), \tag{10}$$

where $|S'_{i,j}|$ is the number of samples associated with t_j, and α is a update rate that controls the size of the update step, which ranges from 0 to 1. This update strategy encourages each key to move toward the averaged features of its associated samples in a controlled manner, and the update process is repeated for e epochs.

3.6 Simulated Online Scenario

In addition to the general coreset update using a fixed support set, we also explore a scenario that is more aligned with realistic and practical considerations: the **simulated online scenario**, where the untapped samples arrive in a streaming pattern. Specifically, we consider a data stream $\mathcal{S}^{\text{stream}} = \{(\boldsymbol{I}_1, \boldsymbol{y}_1), (\boldsymbol{I}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{I}_n, \boldsymbol{y}_n)\}$ of size n, where each sample arrives sequentially. Our framework adapts naturally to this online scenario with minor modifications to the three core steps: key-based coreset initialization, target sample selection, and key-based sample update.

Online Key-based Corese Initialization. Unlike the general scenario where the coreset C is initialized all at once, the online scenario requires a **Filling-based initialization** due to the sequential data stream. To maintain class balance, we allocate a quota of m/j samples per class for a total coreset size m over j classes. For each incoming sample $(\boldsymbol{I}_s, \boldsymbol{y}_s)$, if class $\boldsymbol{y}_s$ has fewer than m/j samples in C, it is directly added. Otherwise, it updates C via our key-based strategy.

Online Target Sample Selection.

To determine the target sample to update, we adopt the same selection strategies as in the general scenario: RS, SS, and DS, as described in Section 3.4.

Online Key-based Sample Update. After selecting the target sample t from the coreset, the incoming sample $(\boldsymbol{I}_s, \boldsymbol{y}_s)$ is used to update the key of t using the same update rule introduced in Equation (10). Since data arrives one sample at a time, the update is applied immediately upon the arrival of each new sample:

$$k_t' = k_t - \alpha(k_t - \phi(\boldsymbol{I}_s)) = (1-\alpha)k_t + \alpha \cdot \phi(\boldsymbol{I}_s), \tag{11}$$

where $\phi(\boldsymbol{I}_s)$ is the visual representation of the incoming sample and α is the update rate that ranges from 0 to 1. This online update strategy enables the coreset to continually integrate new information without storing the entire data stream.

3.7 Inference and Evaluation

Once we obtain the final coreset C' by updating C via KeCO, we can evaluate a test image $\hat{\boldsymbol{I}}$ by retrieving the top-k most relevant samples from C', based on the similarity between $\phi(\hat{\boldsymbol{I}})$ and all stored keys $\boldsymbol{k}$. These retrieved samples constitute the in-context sequence $\mathcal{D}$, which is concatenated with the test image and fed into the frozen LVLM for prediction, as defined in Equation (2).

4 Experiments

4.1 Models

LVLMs Given the limited number of LVLMs supporting multimodal ICL, we employ OpenFlamingo-3B (OF-3B) [1] and IDEFICS-8B (IDE-8B) [16] to evaluate the ICL performance.

Vision Encoder We choose to use CLIP/ViT-L-14 (CLIP) [35] and google/siglip-so400m-patch14-384 (Siglip) [48] to extract visual feature of sample's image, serving as the key for each sample. They are the corresponding visual language models (VLMs) for OF-3B and IDE-8B, respectively. This is done to align the representation space of LVLMs.

4.2 Baselines

To evaluate the effectiveness of KeCO, we compare our method with two traditional ICL baselines.

Fewshot in Coreset (FS-IC). We evaluate the coreset C that is constructed solely through key-based initialization from the support set S, without any refinement using the untapped samples S'. During ICL inference, demonstrations are selected from this unrefined coreset C to form ICEs for prediction.

Fewshot in Support Set (FS-IS). To evaluate the performance on the entire support set S, demonstrations are selected directly from S to form ICEs for ICL inference, with no updates applied to the data.

4.3 Implementation Details

We utilize three image classification datasets, which include both coarse-grained and fine-grained categories. Specifically, they are CUB-200 [41], Stanford Dogs [15], and ImageNet-100 [37](details of these datasets are provided in Appendix A.2). We evaluate the in-context image classification capability of OF-3B, IDE-8B and Qwen2-VL [40] in different methods. For OF-3B, due to its limited context length, we employ probabilistic inference for it. Specifically, for each class name, we compute the probability of the class name given the image and the prompt, denoted as prob(class name|image, ICE sequences), and select the class name with the highest probability as the prediction. Since class names can consist of multiple tokens (e.g., "tiger shark" consists of two tokens), we average the probabilities of all tokens [4]. For IDE-8B and Qwen2-VL [40], we provide a list of candidate choices in the prompt, and we follow [9] to formulate image classification as a multiple-choice problem. In addition to the correct label, three other options are randomly selected from the classes excluding the correct label. Every in-context example is: "<image> Which of these choices is shown in the image? Choices: A.<class name A>, B.<class name B>, C. <class name C>, D. <class name D> Answer with the letter from the given choices directly."

We use top-1 accuracy as the metric to evaluate the performance due to its clarity and common usage. In general, each category in the coreset C contains 5 or 10 samples (depending on the dataset), and the size n of the support set S is five times the size m of C. Therefore, for CUB-200, $n = 5000$ and $m = 1000$; for Stanford Dogs, $n = 6000$ and $m = 1200$; and for ImageNet-100, $n = 5000$ and $m = 1000$. Since $S' = S - C$, the sizes of S' are respectively 4,000, 4,800 and 4,000. Unless otherwise specified, the update rate α is set to 0.2, the epoch e is set to 10, and the batch size b is set to 1,000 (ablation study on α, e and b are provided in Appendix A.3).

4.4 Results and Key Findings.

Incorporating Information from Untapped Data into the Coreset Improves LVLM's ICL Performance. Table 1 clearly shows the ICL performance comparisons across three datasets and Figure 5 illustrates two cases comparing FS-IC with three KeCO methods (RS, SS, and DS) in selecting demonstrations and making predictions for ICL. The results in Table 1 indicate that OF-3B and IDE-8B, when using most KeCO methods (except KeCO-SS), consistently outperform FS-IC, despite identical coreset sizes. The primary difference lies in the KeCO methods' utilization of keys as carriers for the information of untapped data. For instance, in a

Dataset	CUB-200				Stanford Dogs				ImageNet-100			
Method	OF-3B		IDE-8B		OF-3B		IDE-8B		OF-3B		IDE-8B	
	2-shot	4-shot	2-shot	4-shot	2-shot	4-shot	2-shot	4-shot	2-shot	4-shot	2-shot	4-shot
FS-IC	48.11	34.88	84.65	90.36	49.80	50.22	74.42	82.25	65.26	62.92	90.52	94.98
FS-IS	61.04	58.61	85.62	93.09	61.42	62.55	76.89	87.23	71.88	71.98	90.48	95.50
KeCO-SS	49.65	32.22	84.60	90.58	56.60	51.54	75.59	82.27	70.66	63.60	90.78	95.88
KeCO-RS	72.90	72.66	85.45	94.34	72.20	73.04	79.66	90.45	77.50	79.30	91.14	96.12
KeCO-DS	**74.20**	**76.99**	**87.38**	**94.75**	**73.75**	**75.13**	**79.72**	**91.72**	**78.22**	**80.28**	**91.68**	**96.22**
Online Scenario												
KeCO-SS	46.41	30.88	83.51	88.70	54.63	50.89	73.39	82.33	67.48	62.46	90.22	94.76
KeCO-RS	65.90	58.73	85.69	93.41	67.06	67.52	78.38	88.45	74.94	75.26	90.58	95.78
KeCO-DS	**70.61**	**67.81**	**85.91**	**93.77**	**69.62**	**72.37**	**78.87**	**90.06**	**76.96**	**77.64**	**90.88**	**95.94**

Table 1: Performance accuracy (%) of OF-3B and IDE-8B on CUB-200 (coreset size = 1,000, support set size = 5,000), Stanford Dogs (coreset size = 1,200, support set size = 6,000) and ImageNet-100 (coreset size = 1,000, support set size = 5,000). Results are evaluated under different shot conditions (2-shot and 4-shot) across three KeCO methods and compared against two baselines (FS-IC and FS-IS).

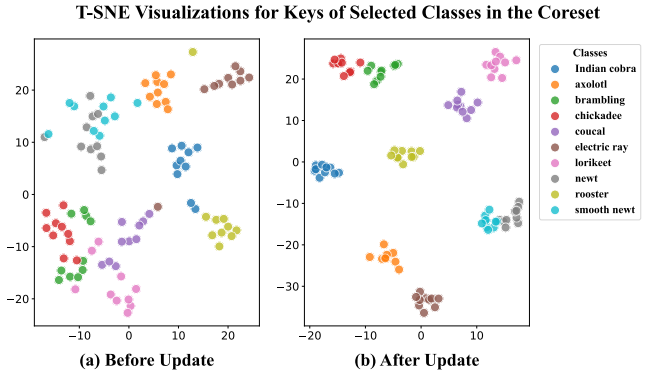

Figure 3: T-SNE visualization of the samples' keys of selected class in (a) the pre-update coreset and (b) the updated coreset using the KeCO-DS method.

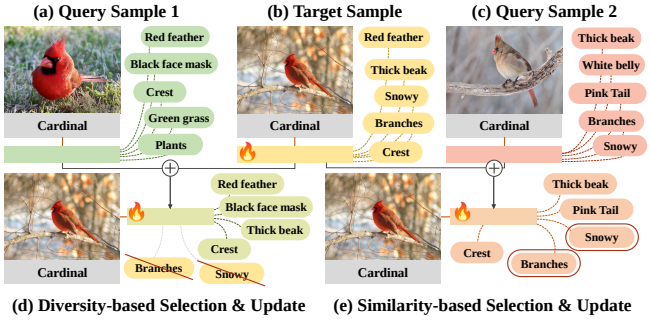

Figure 4: When blending keys from dissimilar samples in (d) Diversity-based selection, the updated key are encouraged to preserves shared, category-relevant information, while suppressing category-irrelevant or misleading attributes. In contrast, when blending similar samples in (e), all information, including misleading information, is retained in the updated key.

4-shot setting compared to FS-IC, KeCO-RS and KeCO-DS enable OF-3B to achieve improvements of 37.78% and 42.11% in the CUB-200, 22.82% and 24.91% in the Stanford Dogs, and 16.38% and 17.36% in the ImageNet-100. Even for the already robust IDE-8B, these methods lead to an increase of 3.98% and 4.39% in the CUB-200, 8.2% and 9.47% in the Stanford Dogs, and 1.14% and 1.24% in the ImageNet-100.

When the coreset keys are updated with untapped data, each update integrates new information into coreset, enhancing the category-relevant information of each sample's key. As illustrated in Figure 3, in the original coreset, the keys or visual features of samples within the same category are quite dispersed. However, after updates with KeCO-DS, these keys become more clustered, making the distinction between different categories more pronounced. This clustering effect facilitates the retrieval of samples belonging to the same category as the test sample, thereby providing more precise and relevant knowledge to the LVLM when conducting ICL.

While a larger support set can accommodate more information, as shown by FS-IS outperforming FS-IC, KeCO-RS and KeCO-DS show superior performance over FS-IS across three datasets, even though the support set size of FS-IS is five times larger than the coreset size of the KeCO methods. For instance, in a 2-shot setting, the ICL performance of OF-3B using the KeCO-DS achieves 74.02%, compared to 61.04% using FS-IS. Similarly, IDE-8B improves from 85.62% with FS-IS to 87.38% with the KeCO-DS on the CUB-200. This further validates the effectiveness of KeCO in enhancing the ICL performance of LVLMs.

The KeCO framework also offers a novel perspective for coreset selection in resource-constrained ICL scenarios, highlighting the potential benefits of optimizing the coreset with untapped data. While current research on retrieval of in-context examples primarily focuses on selecting representative subsets without updates or set the full training dataset as the support set, our findings suggest that even a simple randomly initialized subset can be an effective coreset after updated using untapped data.

Target Sample Selection is Important for KeCO. From the experimental results, it is clear that the selection of target samples plays a crucial role in the effectiveness of KeCO. Both KeCO-RS and KeCO-DS lead to performance improvements, with KeCO-DS exhibiting the best results. As we can see from Equation 11, α is a parameter that ranges from 0 to 1, which determines that k_t' will lie somewhere between k_t and $\phi(I_s)$. Therefore, when a sample least similar to the query sample in untapped set S' is selected for update, both keys are averaged with a weight of α. This effectively maintains the invariant class features while blurring the attribute differences. As illustrated in Figure 4 (d), the invariant part between the keys of the target (b) and query samples (a), such as 'red feather' or 'crest' related to the 'Cardinal' category, are category-relevant information. They exist in both target and query samples, so they will still be preserved in the updated key after merge dissimilar samples in this class. However, category-irrelevant attributes, such as background information ('green grass' vs. 'snowy'), are blended during the update process, because they only appear in one image. This blurs unnecessary distinctions and enables KeCO-DS to focus more effectively on category-relevant features essential for classification.

On the other hand, KeCO-SS performs the worst, sometimes even underperforming FS-IC. This occurs when the query samples in S' select the most similar sample for update, potentially retaining not only useful but also category-irrelevant information, such as background information, in the updated key. As depicted in Figure 4 (e), this is the case when the category-irrelevant information ('snowy' and 'branches') is also preserved in the updated key. Consequently, samples with the same category-irrelevant information but from different categories are more likely to be retrieved as ICE when retrieving from the coreset. This could mislead the LVLMs, negatively affecting their ICL performance.

KeCO Methods Perform Better on Fine-grained Classification Dataset than on Coarse-grained Dataset. In fine-grained classification datasets, the similarity between categories is higher compared to coarse-grained classification, making it more challenging to distinguish between samples of different sub-classes. Table 1 shows that KeCO-RS and KeCO-DS improvements over the baselines are more pronounced on fine-grained classification datasets (CUB-200 and Stanford Dogs) compared to the coarse-grained dataset (ImageNet-100). For instance, for OF-3B, the 2-shot ICL on CUB-200 improves from 61.04% to 74.20%, an increase of 13.16%, whereas on ImageNet-100, it improves from 71.88% to 78.22%, an increase of 6.34%. This discrepancy is due to the fact that the pre-training corpus of LVLMs is unbalanced, with a majority being coarse-grained knowledge [24, 29]. Therefore, LVLMs inherently possess sufficient knowledge about coarse-grained categories but lack knowledge of subordinate-level categories. This can be evidenced from Table 1 where OF-3B and IDE-8B's FS-IC and FS-IS performance on CUB-200 and Stanford Dogs are lower than those on ImageNet-100.

After we optimzie the keys in the coreset, we can retrieve better demonstrations, which supplements useful knowledge when LVLMs perform ICL inference, thereby having a larger impact on fine-grained datasets. However, despite the larger improvement on fine-grained datasets compared to ImageNet, the in-context classification accuracy still does not surpass that on ImageNet, suggesting a need for more balanced pre-training corpora for training LVLMs.

Dataset	FS-IC	FS-IS	Online	Δ
CUB-200	92.95	93.11	98.30	5.35
Stanford Dogs	96.92	97.78	97.93	1.01

Table 2: 2-shot ICL Performance accuracy (%) of Qwen2-VL on CUB-200 and Stanford Dogs. Results are evaluated under KeCO-DS in online scenario and two baselines (FS-IC and FS-IS).

KeCO Methods Retain a Strong Performance in a Simulated Online Scenario. From Table 1, it can be observed that the ICL performance of LVLMs using KeCO-RS and KeCO-DS consistently outperforms two baselines (FS-IC and FS-IS) in a simulated online scenario. For instance, when using KeCO-DS, in the 4-shot setting, the ICL performance of OF-3B on the Stanford Dogs is about 22.15% and 9.82% higher than FS-IC and FS-IS, respectively. Similarly, for IDE-8B, there is an improvement of 7.81% and 2.83%, respectively. In reality, systems usually receive a continuous stream of data, rather than having access to large amounts of data at once that can be reused. Therefore, the improved performance of LVLMs using KeCO-RS and KeCO-DS in the simulated online scenario holds more practical significance, as it is more reflective of real-world situations.

Furthermore, we performed an additional experiment on the high-performance commercial model Qwen2-VL [40], applying KeCO-DS to update coreset in an online scenario. This approach aligns with real-world applications where data typically arrives in a stream, necessitating models that can continuously adapt. As shown in Table 2, in the online scenario, the 2-shot ICL performance on the CUB-200 improved from 92.95% to 98.30%. On the Stanford Dogs, the performance improved by 1%, from an already high baseline of 96.92%, after updating the coreset. This highlights KeCO's efficacy in enhancing the adaptability and responsiveness of high-performance LVLMs like Qwen2-VL to online data with minimal computational overhead.

5 Further Analyses

5.1 Analyses of Different Coreset Initialization

The K-center-greedy (k-center) approach aims to maximize diversity by selecting samples that best represent the overall distribution of the support set. The Infoscore approach prioritizes samples with the highest information gain based on LVLM. After initializing the coreset with these approaches, we used KeCO-RS and KeCO-DS for optimization. As shown in Table 3, the ICL performance of two LVLMs with coresets initialized using the k-center is inferior to those initialized using Infoscore and random methods. This suggests that the k-center-greedy algorithm, despite its popularity in active learning, might not suit ICL. The primary goal in active learning is to diversify the labeled data to enhance the robustness of the trained model [36]. However, in the context of ICL, the model relies on the in-context examples (ICE) to quickly adapt to downstream tasks. Therefore, it is crucial for these ICE to be closely related to

Figure 5: Case study on comparing FS-IC with three KeCO methods (RS, SS and DS) in selecting demonstrations and predicting for in-context learning.

Method	OF-3B			IDE-8B		
	k-center	info	random	k-center	info	random
FS-IC	39.88	47.56	50.22	73.82	81.38	82.25
FS-IS	62.55	62.55	62.55	87.23	87.23	87.23
KeCO-RS	73.26	73.10	73.04	90.64	90.54	90.45
KeCO-DS	75.06	75.22	75.13	91.52	91.38	91.72
Online Scenario						
KeCO-RS	55.07	66.67	67.52	87.51	88.72	88.45
KeCO-DS	58.82	69.81	72.37	89.38	89.64	90.06

Table 3: 4-shot ICL performance accuracy (%) of OF-3B and IDE-8B on Stanford Dogs under different coreset initialization. Results are evaluated across two KeCO methods (KeCO-RS and KeCO-DS) and two baselines (FS-IC and FS-IS).

the test input's label to furnish the model with pertinent knowledge. If the data in the coreset is overly diverse, it may encompass numerous samples with images filled with irrelevant and misleading information, such as images with overly conspicuous background. This could potentially interfere with the LVLM's inference process and diminish its ICL performance.

Furthermore, the ICL performance of LVLMs with a coreset initialized using Infoscore is inferior to that with a coreset initialized randomly. This can be attributed to the inadequate alignment of vision encoder and LLM in LVLMs, typically caused by the LLM's larger scale and the scarcity of high-quality multimodal datasets [10]. As a result, even the seemingly 'useful' samples selected via Infoscore may not function effectively in Multimodal ICL. Moreover, studies such as [3, 49] have demonstrated that Multimodal ICL primarily focuses on text, overshadowing the role played by images. Therefore, the Infoscore obtained by LVLMs is likely to rely more on text information rather than visual information, leading to a deficiency in the evaluation of samples in terms of visual information. Consequently, the Infoscore metric, while applicable in NLP tasks, may not be as effective in more complex multimodal scenarios.

Despite the varying effectiveness of coresets initialized with different methods in the FS-IC setting, the use of the KeCO framework to update coreset leads to substantial improvements, even in the online scenario. For example, with coresets initialized using the k-center-greedy method, OF-3B sees improvements and 35.18% in the common setting and 18.94% in the online scenario after applying KeCO-DS, compared to the results with FS-IC. Similarly, IDE-8B experiences improvements of 15.56% and 17.7% in the common and online settings, respectively. These results further underscore the efficacy of the KeCO framework.

5.2 Analyses of the Size of Additional Data and Coreset

From Table 6 in Appendix A.4, it can be observed that as the amount of untapped data increases, there is an improvement in the FS-IS, KeCO-RS, and KeCO-DS settings. For instance, when the ratio of coreset size to untapped data size increases from 1:2 to 1:6, the performance of OF-3B under the FS-IS setting improves from 58.67% to 63.42%, and IDE-8B under the same setting improves from 75.68% to 77.45%. Similarly, under the KeCO framework, when the ratio changes from 1:2 to 1:6, OF-3B's ICL performance under the KeCO-DS setting improves from 69.99% to 74.58%, and IDE-8B under the same setting improves from 79.02% to 79.92%. The analysis of the coreset size can be found in the Appendix A.4.

6 Conclusion

In our paper, we propose a novel coreset optimization framework, KeCO, which effectively constructs a compact and effective coreset from a large support set. KeCO everages untapped data to aggregate category-relevent information into the coreset via feature-level updates. KeCO achieves larger gains on fine-grained datasets, highlighting its effectiveness in capturing subtle visual distinctions. In addition, our results indicate that diversity-based selection consistently outperforms other strategies, shedding light on better update mechanisms. Our experiments also demonstrate the effectiveness of KeCO in both conventional ICL settings and the proposed simulated online scenario, which better reflects practical use cases.

Acknowledgments

This work is supported by the National Science Foundation of China (62206048), the Natural Science Foundation of Jiangsu Province (BK20220819), and the Fundamental Research Funds for the Central Universities (2242025K30024). This research work is also supported by 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. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems* 35 (2022), 23716–23736.

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

[3] Folco Bertini Baldassini, Mustafa Shukor, Matthieu Cord, Laure Soulier, and Benjamin Piwowarski. 2024. What makes multimodal in-context learning work?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1539–1550.

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

[5] Ting-Yun Chang and Robin Jia. 2022. Data curation alone can stabilize in-context learning. *arXiv preprint arXiv:2212.10378* (2022).

[6] Anurag Das, Xinting Hu, Li Jiang, and Bernt Schiele. 2024. MTA-CLIP: Language-Guided Semantic Segmentation with Mask-Text Alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[7] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234* (2022).

[8] Fu Feng, Yucheng Xie, Xu Yang, Jing Wang, and Xin Geng. 2025. Redefining< creative> in dictionary: Towards an enhanced semantic understanding of creative generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 18444–18454.

[9] Gregor Geigle, Radu Timofte, and Goran Glavaš. 2024. African or european swallow? benchmarking large vision-language models for fine-grained object classification. *arXiv preprint arXiv:2406.14496* (2024).

[10] Hulingxiao He, Geng Li, Zijun Geng, Jinglin Xu, and Yuxin Peng. 2025. Analyzing and Boosting the Power of Fine-Grained Visual Recognition for Multi-modal Large Language Models. *arXiv preprint arXiv:2501.15140* (2025).

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[12] Xinting Hu, Li Jiang, and Bernt Schiele. 2024. Training Vision Transformers for Semi-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[13] Jiaxing Huang, Jingyi Zhang, Kai Jiang, Han Qiu, and Shijian Lu. 2023. Visual instruction tuning towards general-purpose multimodal model: A survey. *arXiv preprint arXiv:2312.16602* (2023).

[14] Yuchu Jiang, Jiale Fu, Chenduo Hao, Xinting Hu, Yingzhe Peng, Xin Geng, and Xu Yang. 2025. Mimic In-Context Learning for Multimodal Tasks. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. 29825–29835.

[15] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. 2011. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, Vol. 2.

[16] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. 2023. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems* 36 (2023), 71683–71702.

[17] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. 2024. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems* 36 (2024).

[18] Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse Demonstrations Improve In-context Compositional Generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1401–1422.

[19] Bo Li, Peiyuan Zhang, Jingkang Yang, Yuanhan Zhang, Fanyi Pu, and Ziwei Liu. 2023. Otterhd: A high-resolution multi-modality model. *arXiv preprint arXiv:2311.04219* (2023).

[20] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2023. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425* (2023).

[21] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. 2023. Otter: A Multi-Modal Model with In-Context Instruction Tuning. *arXiv preprint arXiv:2305.03726* (2023).

[22] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[23] Li Li, Jiawei Peng, Huiyi Chen, Chongyang Gao, and Xu Yang. 2024. How to configure good in-context sequence for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 26710–26720.

[24] Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, et al. 2023. M^3IT: A Large-Scale Dataset towards Multi-Modal Multilingual Instruction Tuning. *arXiv preprint arXiv:2306.04387* (2023).

[25] Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. *arXiv preprint arXiv:2305.04320* (2023).

[26] Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. *arXiv preprint arXiv:2302.13539* (2023).

[27] Zijing Liang, Yanjie Xu, Yifan Hong, Penghui Shang, Qi Wang, Qiang Fu, and Ke Liu. 2024. A Survey of Multimodel Large Language Models. In *Proceedings of the 3rd International Conference on Computer, Artificial Intelligence and Control Engineering*. 405–409.

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems* 36 (2023), 34892–34916.

[29] Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253* (2024).

[30] Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. 2022. What Makes Good In-Context Examples for GPT-3?. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*. 100–114.

[31] Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jiani Zhang, Huzefa Rangwala, Christos Faloutsos, and George Karypis. 2023. Which examples to annotate for in-context learning? towards effective and efficient selection. *arXiv preprint arXiv:2310.20046* (2023).

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

[33] Jane Pan. 2023. *What in-context learning "learns" in-context: Disentangling task recognition and task learning*. Master's thesis. Princeton University.

[34] Jian Qian, Miao Sun, Sifan Zhou, Ziyu Zhao, Ruizhi Hun, and Patrick Chiang. 2024. Sub-SA: Strengthen In-context Learning via Submodular Selective Annotation. *arXiv preprint arXiv:2407.05693* (2024).

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

[36] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)* 54, 9 (2021), 1–40.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115 (2015), 211–252.

[38] Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489* (2017).

[39] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. 2021. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems* 34 (2021), 200–212.

[40] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191* (2024).

[41] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. 2010. Caltech-UCSD birds 200. (2010).

[42] Xu Yang, Yingzhe Peng, Haoxuan Ma, Shuo Xu, Chi Zhang, Yucheng Han, and Hanwang Zhang. 2023. Lever LM: Configuring In-Context Sequence to Lever Large Vision Language Models. *arXiv e-prints* (2023), arXiv–2312.

[43] Xu Yang, Yongliang Wu, Mingzhuo Yang, Haokun Chen, and Xin Geng. 2023. Exploring diverse in-context configurations for image captioning. *Advances in Neural Information Processing Systems* 36 (2023), 40924–40943.

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

arXiv:2304.14178 (2023).

[45] Chenyu You, Nuo Chen, Fenglin Liu, Shen Ge, Xian Wu, and Yuexian Zou. 2022. End-to-end spoken conversational question answering: Task, dataset and model. *arXiv preprint arXiv:2204.14272* (2022).

[46] Chenyu You, Nuo Chen, and Yuexian Zou. 2021. MRD-Net: Multi-Modal Residual Knowledge Distillation for Spoken Question Answering.. In *IJCAI*. 3985–3991.

[47] Chenyu You, Nuo Chen, and Yuexian Zou. 2021. Self-supervised Contrastive Cross-Modality Representation Learning for Spoken Question Answering. In *Findings of the Association for Computational Linguistics: EMNLP*.

[48] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*. 11975–11986.

[49] Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruba Ghosh, Yuchang Su, Ludwig Schmidt, and Serena Yeung-Levy. 2024. Why are visually-grounded language models bad at image classification? *arXiv preprint arXiv:2405.18415* (2024).

[50] Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2023. What makes good examples for visual in-context learning? *Advances in Neural Information Processing Systems* 36 (2023), 17773–17794.

[51] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592* (2023).