DEMORERANKER: ENHANCING THE IN-CONTEXT LEARNING CAPABILITY OF MULTI-MODAL LARGE MODELS VIA DEMONSTRATION RERANKING

Anonymous authors
Paper under double-blind review

ABSTRACT

In the deployment of Large Multi-modal Models (LMMs), researchers and practitioners often rely on simplistic strategies for in-context learning (ICL), such as reusing fixed demonstrations across diverse samples or retrieving candidates directly using the CLIP model. These approaches may not ensure that selected demonstrations align optimally with LMM requirements. To bridge this gap, we introduce DemoReranker, a novel framework that fine-tunes a specialized reranker module to improve demonstration selection for LMMs. First, we assess demonstration quality by measuring its influence on model outputs. Second, our reranker incorporates a scoring head atop the CLIP embedding model, evaluating compatibility between test samples and candidate demonstrations. Third, we optimize the reranker using list-wise ranking loss while keeping the CLIP backbone frozen. Extensive experiments across 7 datasets spanning 3 multi-modal tasks confirm that DemoReranker effectively enhances LMM performance in ICL by reranking demonstrations to identify the most suitable candidates.

1 INTRODUCTION

In-context learning (ICL) is a simple yet powerful paradigm where a model learns to predict outcomes on new, unseen tasks by analyzing a small set of input-output examples (few-shot demonstrations). First showcased in GPT-3 Brown et al. (2020) and recognized as an emergent capability primarily found in large-scale pre-trained models Wei et al. (2022), ICL has since attracted significant interest within the artificial intelligence field. Numerous studies have highlighted the remarkable ICL abilities of large language models (LLMs) across diverse natural language processing (NLP) applications. ICL holds particular practical value, as it allows large pre-trained models to be quickly adapted to novel tasks or specific user needs using minimal examples. Crucially, this approach eliminates the requirement for model retraining or redeployment.

Significant advancements in Large Multi-modal Models (LMMs) have spurred considerable interest in their in-context learning (ICL) capabilities Alayrac et al. (2022). State-of-the-art LMMs, including DeepSeek-VL Wu et al. (2024) and Qwen-VL Bai et al. (2023), exhibit strong ICL performance across diverse tasks like visual question answering (VQA), visual classification (VCLS), and visual captioning (VCAP). Despite this success, prevailing strategies for enabling ICL in LMMs typically utilize simplistic techniques—often fixed examples or examples prioritized using pre-trained vision-language embedding models. A key drawback of these strategies is their neglect of the LMM's own feedback concerning the potential contribution of the examples to enhancing its output quality.

To address the aforementioned challenges, we propose a novel framework: Demonstration Reranker (DemoReranker). DemoReranker aims to fine-tune a pre-trained vision-language embedding model using linear probing, enabling it to learn how to re-rank retrieved examples. This process ultimately produces an ordered list of examples tailored to the requirements of Large Vision-Language Models (LMMs) during inference. Specifically, DemoReranker operates as follows: (a) Re-ranking by Example Quality Score: The framework first re-orders examples retrieved by the embedding model based on a proposed example quality score, defined as the BARTScore of the response generated conditioned on the given example. (b) Architecture: The example re-ranker is implemented as a re-ranker module (or re-ranking head) attached atop the frozen CLIP embedding model. (c) Fine-

tuning Objective: This re-ranker module is then fine-tuned using our proposed list-wise ranking loss, while the parameters of the underlying CLIP backbone remain fixed. Through this fine-tuning procedure, our approach effectively transfers the signal captured by the example quality score into the learned parameters of the example re-ranker.

We carried out comprehensive testing on a range of public benchmarks and internal datasets, spanning three unique multimodal domains including Visual Question Answering (VQA), Visual Classification (VCLS), and Visual Captioning (VCAP). Our findings reveal that the DemoReranker framework significantly improves the in-context learning (ICL) performance of large vision-language models (LMMs). Notably, the fine-tuned reranking model developed through DemoReranker can be adapted for use with commercial LMMs like GPT-4o, enhancing their capacity to access pertinent contextual information. The core advancements of this work are highlighted below:

- We introduce DemoReranker, an innovative architecture engineered to elevate the in-context learning efficiency of LMMs.
- Through rigorous analysis across diverse vision-language tasks and datasets, we confirm that DemoReranker achieves substantial improvements in ICL outcomes.

2 Related works

LMMs and In-Context Learning (ICL) Building on the achievements of Large Language Models (LLMs) in natural language understanding, researchers have developed Large Multimodal Models (LMMs) for processing both visual and textual information Du et al. (2022). Prominent examples include BLIP-2 Li et al. (2023a), MiniGPT-4 Zhu et al. (2023), and LLaVA Liu et al. (2024), which utilize adapter-based architectures Houlsby et al. (2019) to bridge visual and linguistic representations during pre-training, thus lowering resource demands. Despite the growing number of LMMs, a key limitation is that many lack support for contextual learning, which necessitates the ability to handle inputs combining images and text in a mixed format Alayrac et al. (2022). The exploration of contextual learning within multimodal systems, particularly for LMMs, is still in its early stages, with few works focusing on core methodologies. For instance, Yang et al. (2024b) studied how contextual learning influences LMM performance in generating image descriptions, while Li et al. (2024) evaluated its effectiveness and introduced methods for selecting relevant examples using pre-trained multimodal encoders such as CLIP Radford et al. (2021). In line with these efforts, our research advances the field by introducing a new framework designed to enhance contextual learning in LMMs.

The field of artificial intelligence has experienced transformative progress in the evolution of Large Language Models (LLMs) over the past few years. As these models grow in complexity and the scale of their training datasets expands, they have begun to exhibit unexpected capabilities, including advanced reasoning, numerical problem-solving, and the ability to execute instructions based on contextual cues Wei et al. (2022). The pioneering work on GPT-3 Brown et al. (2020) demonstrated that sufficiently large models could infer and execute novel tasks from minimal input examples—a concept later formalized as In-Context Learning (ICL). Subsequent studies have further validated the robustness of ICL, showcasing its effectiveness across a wide spectrum of linguistic and cognitive tasks Mosbach et al. (2023). A key component of leveraging ICL successfully hinges on curating high-quality, contextually relevant example sequences Li et al. (2023b). However, much of the existing body of research remains concentrated on text-based natural language processing (NLP) applications and language-centric foundational architectures, underscoring a pressing need to adapt and generalize these insights to multimodal or domain-specific scenarios.

3 Method

We now detail the technical aspects of our framework. During the renranking fine-tuning phase of our method, the dataset for any vision-language task is partitioned into four distinct subsets: a support set ($\mathcal{D}_{supp}$), a training set ($\mathcal{D}_{train}$) for fine-tuning the embedding model $\mathcal{E}$, a validation set ($\mathcal{D}_{dev}$), and a test set ($\mathcal{D}_{test}$) for evaluating the LMM's in-context learning performance.

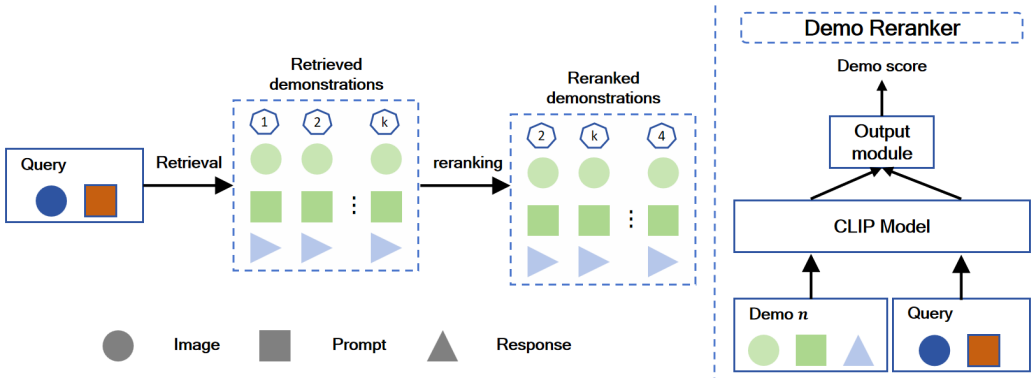


Figure 1: The framework of our DemoReranker.

3.1 In-context learning

Given a pre-trained LMM, denoted as $\mathcal{M}$ (e.g., Deepseek-VL2 Wu et al. (2024)), in-context learning (ICL) requires a multimodal context sequence $\mathcal{Z} = \{z_1, \ldots, z_n\}$. This sequence $\mathcal{Z}$ consists of n exemplars (n-shot), where each exemplar z_i is a tuple comprising an image, a prompt, and its corresponding response: $z_i = (\text{image}_i, \text{prompt}_i, \text{response}_i)$. The context sequence $\mathcal{Z}$ is then concatenated with the test sample $x_{\text{test}} = (\text{image}_{\text{test}}, \text{prompt}_{\text{test}})$ and fed into the LMM $\mathcal{M}$. The model subsequently generates a response $\hat{r}_{\text{test}}$ of length $T_A > 0$ tokens:

$$\hat{r}_{\text{test}} = \mathcal{M}(\mathcal{Z}, x_{\text{test}}) = \{\hat{a}_1, \ldots, \hat{a}_{T_A}\}. \tag{1}$$

3.2 Embedding Strategies

Unlike approaches that retrieve demonstrations solely through images or text Li et al. (2024), our method utilizes the concatenated vector of image embeddings and text embeddings generated by the CLIP model Radford et al. (2021). This strategy is termed Image-Text Embedding (ITE). Each demonstration $z_i = (\text{image}_i, \text{prompt}_i, \text{response}_i)$ is passed to the CLIP model, yielding an image vector v_i^{image} and a text vector v_i^{text}. These vectors are subsequently concatenated to form the joint embedding vector v_i^{clip}. A vector database is constructed using the Faiss toolkit to perform efficient similarity search for retrieval. During inference, the test sample x_{test} is likewise converted into its corresponding CLIP vector $v_{\text{test}}^{\text{clip}}$, which serves as the query input for the vector search.

3.3 Retrieval of demonstration samples

The existing methods Li et al. (2024); Yang et al. (2024b) assumes that an embedding model $\mathcal{E}$ (e.g., CLIP Radford et al. (2021)) is readily available and effective for retrieving the most beneficial examples for a given query. For the current task, we first utilize the base CLIP model to construct a vector database of examples from $\mathcal{D}_{supp}$. For a sample $xq = (\text{image}_q, \text{prompt}_q, \text{response}_q)$ within $\mathcal{D}_{dev}$, we generate its embedding vector using the method described in Section 3.2. We then retrieve $n > 0$ candidate examples $\{z_j\}_{j=1}^{n}$ (corrected index from i to j for consistency). These candidates are ranked based on their embedding similarity scores:

$$r_0(z_j) = \text{sort}\left(\text{sim}(x_q, z_j) \mid \{z_j\}_{j=1}^{n}\right), \tag{2}$$

where $\text{sim}(x_q, z_j)$ denotes the cosine similarity between the CLIP embedding vectors of x_q and z_j, and sort represents the ascending order sorting function.

3.4 Candidate Example Evaluation

The core value of examples for LMMs lies in enhancing the quality of generated responses. Therefore, the following evaluation method is proposed: First, an example z_j assists the LMM in generating a response:

$$\text{response}_j = \mathcal{M}(z_j, \text{image}_q, \text{prompt}_q). \tag{3}$$

The surface-level and semantic similarity between response_j and the ground-truth response response_q is measured using BARTScore Yuan et al. (2021). This similarity score serves as the quality score for z_j:

$$s(z_j) = \text{BARTScore}(\text{response}_q, \text{response}_j). \tag{4}$$

The candidate examples are then ranked based on their quality scores:

$$r(z_j) = \text{rank}(s(z_j) \mid \{s(z_i)\}_{i=1}^{n}) \tag{5}$$

A higher $s(z_j)$ value indicates that the example z_j is more effective in improving the LMM's response quality; consequently, $r(z_j)$ will also be larger (indicating a higher rank).

3.5 Exploratory Experiments and Motivation

Given two distinct rankings of the same set of candidate examples, the ranking correlation can be computed as:

$$\text{Corr}_q = \text{Spearman}\left(\{r(z_j)\}_{j=1}^{n}, \{r_0(z_j)\}_{j=1}^{n}\right), \tag{6}$$

where Spearman denotes Spearman's rank correlation coefficient Dodge (2008). The average correlation score is then given by:

$$\text{Corr}_{\text{avg}} = \frac{\sum_{x_q \in \mathcal{D}_{\text{dev}}} \text{Corr}_q}{|\mathcal{D}_{\text{dev}}|}. \tag{7}$$

Exploratory experiments conducted by us reveal an extremely low correlation (Corr_{avg}) between the rankings generated by the CLIP model and those generated by the LMM. This result suggests that candidate demonstrations retrieved by $\mathcal{E}$ may not adequately meet the requirements of the LMM.

This observation aligns with findings reported in prior studies Li et al. (2023b); Rubin et al. (2021), which indicate that examples retrieved using open-source embedding models are not necessarily optimal for LMM performance. Consequently, it is necessary to construct an example re-ranker. This re-ranker will reorder the $n > 0$ candidate examples, ultimately selecting the top $n_1 > 0$ ($n_1 < n$) most effective examples to enhance prompt efficacy.

3.6 Exploratory Experiments and Motivation

Algorithm 1 DRAFT's demonstration reranker fine-tuning

Input: Reranker $\mathcal{E}_r$ constructed on the embedding model $\mathcal{E}$, LMM $\mathcal{M}$, number of epochs N_1, number of steps per epoch N_2, number of retrieved candidates n. support set $\mathcal{D}_{supp}$, model $\mathcal{E}$'s training set $\mathcal{D}_{train}$, model $\mathcal{E}$'s validation set $\mathcal{D}_{dev}$, test set for the LMM $\mathcal{D}_{test}$.

return A fine-tuned reranker $\mathcal{E}_r$.
Embed each training example in $\mathcal{D}_{supp}$ with $\mathcal{E}$;
$i \leftarrow 0$;
for $i = 1$ to N_1 **do**
  $j \leftarrow 0$;
  for $j = 1$ to N_2 **do**
    Sample an querying example x_q from $\mathcal{D}_{train}$;
    Obtain its demonstration candidates $\{z_k\}_{k=1}^{n}$ from $\mathcal{D}_{supp}$;
    Re-rank $\{z_k\}_{k=1}^{n}$ using Eq 5;
    Calculate $\mathcal{L}_r$ using Eq 10;
    Update $\mathcal{E}_r$;
    $j \leftarrow j + 1$;
  end for
  $i \leftarrow i + 1$;
end for

We describe the construction and fine-tuning methodology for our framework's core component: the example re-ranker. The complete procedure is presented in Algorithm 1.

Re-ranker Architecture As illustrated in the right panel of Figure 1, the example re-ranker $\mathcal{R}$ is built upon the CLIP model $\mathcal{E}$. After encoding z_j and x_q into vectors v_j and v_q via $\mathcal{E}$, we concatenate

these representations and feed them into a reranking head module to predict quality scores $\hat{s}(z_j) \in [0, 1]$:

$$\hat{s}(z_j) = \text{RerankHead}\left(\text{Concat}\left([v_j, v_q]\right)\right), \tag{8}$$

where Concat$(\cdot)$ denotes vector concatenation. The reranking head is implemented as a multilayer perceptron with a Sigmoid activation. During fine-tuning, only the RerankHead() parameters are optimized while the CLIP backbone remains frozen. The predicted quality ranking is obtained by:

$$\hat{r}(z_j) = \text{rank}\left(\hat{s}(z_j) \mid \hat{s}(z_i)_{i=1}^{n}\right). \tag{9}$$

Loss Function Our training objective aligns the predicted ranking in Equation 9 with the ground-truth ranking in Equation 5. We propose the following loss function to inject ranking signals:

$$\mathcal{L}_r = \sum {}^{1 \le i,j \le n \ i \ne j} \max\left(0, m(i,j) \cdot \left(\hat{s}(z_j) - \hat{s}(z_i)\right)\right), \tag{10}$$

where the weighting coefficient $m(i, j)$ is defined as:

$$m(i,j) = \max\left(0, r(z_i)^2 - r(z_j)^2\right). \tag{11}$$

This novel loss function in Equation 10 extends the pairwise comparison approach of Wang et al. (2014) by incorporating listwise ranking information through $m(i, j)$, which dynamically adjusts pair weights.

Intuitive Interpretation: Consider example pair (z_i, z_j). When $r(z_j) \ge r(z_i)$, $m(i, j) = 0$, excluding this pair from loss contribution. When $r(z_j) < r(z_i)$, $m(i, j) > 0$. If the re-ranker's predictions satisfy $\hat{s}(z_i) > \hat{s}(z_j)$ (consistent with Equation 4), the loss remains zero. However, if $\hat{s}(z_i) \le \hat{s}(z_j)$, the term $(\hat{s}(z_j) - \hat{s}(z_i)) \ge 0$ incurs positive loss, driving parameter adjustments to correct the score ordering.

Retrieval	**VQA**			**VCLS**		**VCAP**	
Methods	VQAv2	VizWiz	OK-VQA	Plants52	Hateful-Memes	Flicker30K	NoCaps
Null	58.3	27.6	42.3	14.6	58.8	27.7	29.6
Random	68.5	46.2	56.3	31.5	64.7	37.5	40.4
Fixed	68.6	46.7	57.9	32.3	64.5	38.1	40.7
SparseRetrieval	70.0	37.5	55.8	25.7	60.1	33.9	35.3
Dino	71.7	49.6	59.9	35.7	66.6	39.0	39.6
BGE	71.1	41.7	61.2	26.6	60.2	34.3	36.2
CLIPRetrieval	71.9	61.2	63.4	36.5	68.8	39.2	41.7
UDR	72.6	64.3	64.9	38.5	70.3	40.3	42.3
Lever-LLM	73.4	64.4	65.2	39.0	71.3	40.4	42.8
DemoReranker	**76.1**	**66.8**	**67.5**	**41.2**	**74.3**	**41.7**	**45.2**

Table 1: Results on the 7 tasks. Best scores are bolded.

4 EXPERIMENTS

4.1 DATASETS

We conducted experiments on seven multimodal tasks spanning three categories: five open benchmarks and two proprietary tasks contributed by industrial partners. The tasks include: (a) VQAv2 Goyal et al. (2017), (b) VizWiz Gurari et al. (2018), (c) PS-VQA (a proprietary VQA dataset focused on healthy dietary choices), (d) Hateful-Memes Kiela et al. (2020), (e) Plants52 (a proprietary plant classification dataset), (f) Flickr30K Plummer et al. (2015), and (g) NoCaps Agrawal et al. (2019).

For each dataset, we repartitioned the original training/validation/test splits to create distinct subsets required by our workflow: the support set ($\mathcal{D}_{supp}$), training set ($\mathcal{D}_{train}$) and validation set ($\mathcal{D}_{dev}$) for fine-tuning the example retrieval model, and a test set ($\mathcal{D}_{test}$) for evaluating the in-context learning performance of the language model.

4.2 EVALUATION METRICS

Metrics for VQA We adopt the evaluation framework from Alayrac et al. (2022) and compute accuracy for the VQA task using the following formula:

$$\text{Acc}_{a_i} = \min\left(1, \frac{3 \times \sum_{k=0}^{9} \text{match}(a_i, g_k)}{10}\right),$$

where a_i represents the model's predicted response, g_k denotes the k-th reference answer among the top 10 candidates, and the match() function outputs 1 if the predicted and reference answers align, or 0 otherwise.

Metrics for VCLS For VCLS tasks, we consider accuracy, that is, the proportion of correctly predicted class labels.

Metrics for VCAP To assess VCAP results, we report the ROUGE-L score Lin (2004), a metric that quantifies the overlap between generated captions and reference captions in terms of recall, precision, and harmonic mean (F-score). This metric emphasizes both n-gram matching and sequence-level coherence.

4.3 PROMPT TEMPLATES

We now present the prompt templates we use for the different tasks under ICL.

Prompt template for the VQA task If we do not use any demonstrations, the prompt template for the VQA task is:

```
<image>
Question: [question]
Instruction: answer with a short phrase.
Answer:
```

in which <image> is the placeholder for the input image, [question] is the input question.

The prompt template for VQA with a group of demonstrations is:

```
<demo_image>
Question: [demo_question]
Answer: [demo_answer]

<demo_image>
Question: [demo_question]
Answer: [demo_answer]

Read the above demonstrations, and incorporate them when dealing with
    the following query.

<image>
Question: [question]
Instruction: answer with a short phrase.
Answer:
```

in which <demo_image> is the placeholder for the image in the demonstration sample, [demo_question] is the demonstration question, and [demo_answer] is the corresponding ground-truth answer.

The prompt templates for the other two types of tasks are presented in Appendix A.

5 BASELINES

Using the same inference LMM, we evaluate our DemoReranker approach against existing demonstration retrieval methods by comparing their downstream in-context learning (ICL) performance. The compared methods include: (a) NoDemo: Uses no demonstrations. (b) RandomDemo: Randomly selects demonstrations from the supporting set. (c) SparseRetrieval: A widely adopted sparse retriever from prior literature Chen et al. (2017). (d) DINO: Retrieves demonstrations using image

embeddings from the DINO model Caron et al. (2021). (e) BGE: Retrieves demonstrations using text embeddings from the BGE model Chen et al. (2024). (f) CLIPRetrieval: Retrieves demonstrations using the joint image-text embeddings provided by the CLIP model Radford et al. (2021), as explored in Li et al. (2024). (g) UDR Li et al. (2023b): Evaluated solely on text-based tasks, this method trains a classifier to score demonstrations. (h) Lever-LLM Yang et al. (2024a): Employs a small language model as a demonstration selector. This approach differs from ours in both the fine-tuned model architecture and its loss objectives.

Method	VizWiz	Hateful-Memes	Flicker30K
DemoReranker	**66.8**	**74.3**	**41.7**
DemoReranker-1	66.1	73.2	40.3
DemoReranker-2	65.6	72.9	39.4
DemoReranker-3	66.3	73.7	41.5
DemoReranker-4	66.2	73.6	41.2

Table 2: Results of the ablation study on DemoReranker's training strategy.

5.1 Experimental settings

Computing infrastures All experiments utilize NVIDIA V100 GPUs.

LMM models We adopt the DeepSeek-VL2 Tiny model (3B) Wu et al. (2024) as our Large Multimodal Model (LMM) for evaluating the DemoReranker approach.

Decoding During response generation, we apply beam search decoding with a beam size of 3.

ICL Setup for the LMM Model $\mathcal{M}$ Under DemoReranker, each test sample triggers a two-step process: First, 32 candidate demonstrations are retrieved. These are then reranked using our demonstration reranker, from which the top $n_1 = 4$ demonstrations are selected. The chosen demonstrations are ordered by ascending similarity score and prepended to the test sample input.

Settings for embedding and retrieval This work employs the base CLIP model Radford et al. (2021) for image-text embeddings by default. The ITE strategy from Section 3.2 serves as our standard sample embedding approach. Efficient demonstration retrieval is implemented using the Faiss toolkit Douze et al. (2024).

Settings for demonstration reranker The demonstration reranker employs a two-layer Multilayer Perceptron (MLP) utilizing ReLU activation functions. Its output is subsequently processed by a Sigmoid activation layer. We adapted the fine-tuning procedure from the Huggingface Transformers codebase Wolf et al. (2020). In Equation 10, the temperature parameter τ is set to 5.0. Training involves $N_1 = 50$ epochs for the embedding model, each comprising $N_2 = 200$ steps. During fine-tuning, the model retrieves $n = 32$ examples. For optimization, we utilize AdamW Loshchilov & Hutter (2019) with a learning rate of $1e-5$ and a 50-step warmup phase. Remaining hyperparameters are aligned with the Transformers library defaults. Performance on the $\mathcal{D}_{dev}$ set is assessed after every epoch using Equation 7. An early stopping strategy with a patience of 10 epochs is implemented; training halts if the corr_{avg} metric shows no improvement over ten consecutive evaluations.

5.2 Main Results

Table 1 presents the performance comparison of various methods across seven multi-modal tasks. The results indicate that: (a) DemoReranker achieves significantly superior results over baseline methods on the majority of tasks. This demonstrates its enhanced capability for detecting effective demonstrations across diverse multi-modal applications. (b) Specifically, compared to UDR, DemoReranker exhibits stronger overall performance, confirming the efficacy of our training methodology and loss function. Furthermore, it holds distinct advantages over directly employing CLIP as the demonstration retriever Li et al. (2024). This directly demonstrates that our proposed framework can develop an effective demonstration reranker via the fine-tuning procedure, thereby selecting more beneficial demonstrations.

Strategy	Method	VizWiz	Hateful-Memes	Flicker30K
ITE2ITE	DemoReranker	**66.8**	**74.3**	**41.7**
	UDR	64.3	70.3	40.3
TE2TE	DemoReranker	51.3	64.5	37.2
	UDR	50.1	64.4	37.2
IE2IE	DemoReranker	65.7	74.1	41.7
	UDR	63.5	69.9	40.3

Table 3: Results of the ablation study on the embedding strategy.

5.3 Ablation studies and further analysis

Ablation Study on our DemoReranker framework To evaluate our DemoReranker's each component, we consider the following variant of DemoReranker: (a) DemoReranker-1 alters the loss function in Eq 10 to the regression-based mean square error loss Das et al. (2004). That is, the demonstration reranker directly learns to predict the demonstration quality score. (b) DemoReranker-2 drops the weight coefficient $m(i, j)$ from Eq 10. (c) DemoReranker-3 substitutes the weight coefficient $m(i, j)$ to $m(i, j) = \max(0, r(z_i) - r(z_j))$. (4) DemoReranker-4 substitutes the weight coefficient $m(i, j)$ to $m(i, j) = \max(0, r(z_j)^{-1} - r(z_i)^{-1})$. The results are reported in Table 2.

The results in Table 2 show that: (a) The comparison between DemoReranker-1 and DemoReranker demonstrates the directly modeling the demonstration reranker also works well, but performs slightly worse than the loss function in Eq 10. (b) The comparison between DemoReranker-2 and DemoReranker proves the necessity of the weight coefficient $m(i, j)$, which can effectively inject listwise information into the loss function. (c) The comparison among DemoReranker-3, DemoReranker-4 and DemoReranker proves the function form of $m(i, j)$ is important, and the setting in Eq 11 is valid, since it leads to the best performance.

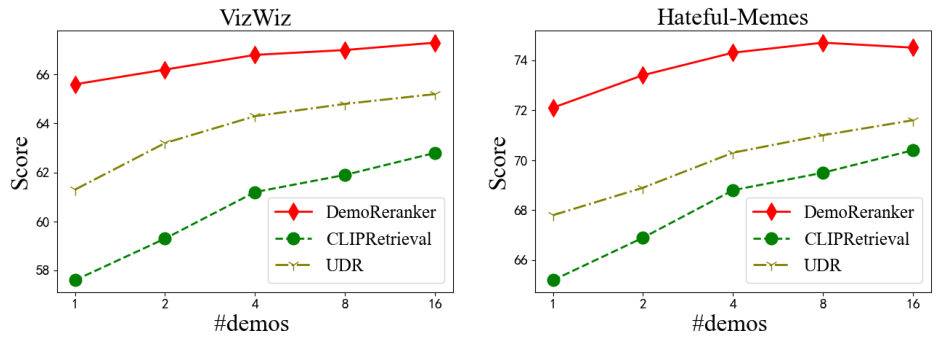
LMM	Method	VizWiz	Hateful-Memes	Flicker30K
GPT-4o	CLIPRetrieval	72.1	76.9	41.1
	UDR	75.6	79.0	42.9
	DemoReranker	**78.1**	**81.9**	**45.8**
Claude 3 Opus	CLIPRetrieval	71.5	76.2	38.2
	UDR	75.3	78.3	41.6
	DemoReranker	**77.8**	**80.7**	**43.2**

Table 4: Experiments on the transfer learning capabilities of DemoReranker. We using the fine-tuned reranker to retrieve demonstrations for GPT-4o and Claude 3 Opus.

Ablation on the embedding strategy This work uses the strategy of concatenating the image and text embeddings (ITE) for sample embedding in the main experiment (Table 1). That is, the demonstration sample embeddings in the Faiss index is obtained via the ITE strategy, and the search input is obtained by embedding the test sample with the ITE strategy. To demonstrate the rationality of this ITE2ITE setting, we conduct an ablation study on different sample embedding strategies. (a) The demonstrations and search input use the strategy of embedding the text input (TE). This setting is denoted as TE2TE. (b) The demonstrations and search input use the strategy of embedding the image input (IE). This setting is denoted as IE2IE.

Table 3 presents the experimental results, demonstrating that: (a) The ITE2ITE strategy outperforms the alternatives. This approach fuses image and text data for demonstration retrieval, leveraging the richest semantic information available from the test sample. (b) Regardless of the embedding strategy used, our DemoReranker method consistently surpasses the UDR approach.

Transferability across Different LMs Note that fine-tuning our DemoReranker methods requires repeated inference on the Large Multimodal Model (LMM) $\mathcal{M}$. This can lead to prohibitive API costs if $\mathcal{M}$ is a commercial model like GPT-4o. Given that different LMMs share similar training mechanisms and are pre-trained on vast internet datasets, their internal representations exhibit significant commonalities. Therefore, in this experiment, we utilize the demo reranker fine-tuned using feedback signals from the DeepSeek-VL2 model – to perform demonstration reranking for powerful commercial LMMs such as GPT-4o or Claude 3 Opus. The results are shown in Table 4.



(a) VizWiz (b) Hateful-Memes

Figure 2: The effects of the number of demonstrations on DemoReranker, UDR, and CLIPRetrieval.

As evidenced by Table 4, the demo reranker, fine-tuned with DeepSeek-VL2 feedback, successfully identifies suitable demonstrations given a test query. This leads to enhanced performance in commercial LMMs like GPT-4o or Claude 3 Opus across different multi-modal tasks. This outcome validates the practical value of the DemoReranker approach for industrial applications: an open-source LMM can be used to train the demo reranker model, which is then deployed to improve the in-context learning of commercial LMMs.

Impact of demonstration quantity In our primary experiments (Table 1), we fixed the demonstration list size at $n_1 = 4$. We now evaluate DemoReranker against CLIPRetrieval and UDR across varying demonstration quantities, with results shown in Figure 2.

DemoReranker consistently surpasses baseline methods regardless of demonstration volume. Two key observations emerge: (a) Demonstration quantity exerts greater influence on the VizWiz generation task than the Hateful-Memes classification task. Specifically, increasing demonstrations yields substantial performance gains for VizWiz but only marginal improvements for Hateful-Memes. (b) Demonstration quality proves more critical than quantity. Notably, DemoReranker using just one or two demonstrations still outperforms UDR with four demonstrations. These findings further demonstrate DemoReranker's robust demonstration retrieval capability.

6 Conclusion

This paper introduces DemoReranker, a unified framework tailored to large vision-language models (LVLMs) for demonstration retrieval and reranking. We train DemoReranker by leveraging LVLM evaluations of candidate demonstrations; this information is incorporated into the reranker through a knowledge distillation loss. Experiments conducted on seven vision-language tasks demonstrate that DemoReranker achieves significant performance gains over baseline retrieval methods. Further analysis validates the efficacy of each core component in our framework and highlights its strong transferability across LVLMs of varying scales (3B to 175B parameters) and different quantities of demonstrations.

References

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8948–8957, 2019.

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.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

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.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*, 2017.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 2318–2335, 2024.

Kalyan Das, Jiming Jiang, and JNK Rao. Mean squared error of empirical predictor. 2004.

Y Dodge. *The concise encyclopedia of statistics*. Springer New York, 2008.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv preprint arXiv:2401.08281*, 2024.

Yifan Du, Zikang Liu, Junyi Li, and Wayne Xin Zhao. A survey of vision-language pre-trained models. *arXiv preprint arXiv:2202.10936*, 2022.

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*, pp. 6904–6913, 2017.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3608–3617, 2018.

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

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in neural information processing systems*, 33:2611–2624, 2020.

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

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

Xiaonan Li, Kai Lv, Hang Yan, Tianya Lin, Wei Zhu, Yuan Ni, Guo Tong Xie, Xiaoling Wang, and Xipeng Qiu. Unified demonstration retriever for in-context learning. *ArXiv*, abs/2305.04320, 2023b. URL `https://api.semanticscholar.org/CorpusID:258557751`.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. *arXiv preprint arXiv:2305.16938*, 2023.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

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*, pp. 8748–8763. PMLR, 2021.

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

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1386–1393, 2014.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

Xu Yang, Yingzhe Peng, Haoxuan Ma, Shuo Xu, Chi Zhang, Yucheng Han, and Hanwang Zhang. Lever lm: configuring in-context sequence to lever large vision language models. *Advances in Neural Information Processing Systems*, 37:100341–100368, 2024a.

Xu Yang, Yongliang Wu, Mingzhuo Yang, Haokun Chen, and Xin Geng. Exploring diverse in-context configurations for image captioning. *Advances in Neural Information Processing Systems*, 36, 2024b.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: evaluating generated text as text generation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 27263–27277, 2021.

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.

A PROMPT TEMPLATES

Prompt template for the image captioning task If we do not use any demonstrations, the prompt template for the image captioning task is:

```
<image>
Instruction: write a concise caption for the image.
Response:
```

in which <image> is the placeholder for the input image.

The prompt template for VQA with a group of demonstrations is:

```
<demo_image>
Response: [demo_caption]

<demo_image>
Response: [demo_caption]

Read the above demonstrations, and incorporate them when dealing with
    the following query.
<image>
Instruction: write a concise caption for the image.
Response:
```

in which <demo_image> is the placeholder for the image in the demonstration sample, [demo_caption] is the ground-truth caption.

Prompt template for the image classification task If we do not use any demonstrations, the prompt template for the image classification task is:

```
<image>
Instruction: assign one of the following labels to the input image.
[label_list]
Response:
```

in which <image> is the placeholder for the input image, and the [label_list] is the collection of label names specified in the given classification task.

The prompt template for VQA with a group of demonstrations is:

```
<demo_image>
Response: [demo_label]

<demo_image>
Response: [demo_label]

Read the above demonstrations, and incorporate them when dealing with
    the following query.
<image>
Instruction: assign one of the following labels to the input image.
[label_list]
Response:
```

in which demo_image is the placeholder for the image in the demonstration sample, [demo_label] is the ground-truth caption.