Retrieving Counterfactuals Improves Visual In-Context Learning

Supplementary Material

A. Implementation Details

A.1. Baseline Implementations

Backbone vision-language models. All baselines and CIRCLES are instantiated with the same backbone vision-language models (VLMs) to ensure a fair comparison. We use Gemma3 [42] with 4B and 12B parameters (denoted as Gemma3-4B and Gemma3-12B) and Qwen2.5-VL [6] with 3B and 7B parameters (denoted as Qwen2.5-VL-3B and Qwen2.5-VL-7B) from the HuggingFace platform. For all models, we run inference with BFloat16 precision using the vLLM library.

Retrieval backbone. For all retrieval-based baselines, including RICES, MUIER, MMICES, and CIRCLES, we use CLIP ViT-g/14 ("laion/CLIP-ViT-g-14-laion2B-s12B-b42K") [36] as the shared image-text encoder. Unless otherwise specified, all visual and textual features are extracted from the CLIP image/text encoder and L2-normalized before similarity computation.

Prompting setup. We use a unified prompting template across all methods for each task. For each instance, the prompt includes the query image and question, followed by in-context demonstrations selected by the respective method. The input query is repeated at the end of the prompt for clarity. All methods and VLMs use a decoding temperature of 0.0 for deterministic generation, and the maximum output length is set to 512 tokens for all experiments. For VQA tasks (OK-VQA and VizWiz), in-context demonstrations consist of the original question-answer pairs. For classification tasks (CUB and Flowers), we use fixed question templates and the corresponding class labels as in-context examples. Specifically, the question template for CUB is "What is the category of the bird in this image?", and for Flowers, "What is the category of the flower in this image?".

Baselines. We implement the following baselines:

- **None** (zero-shot): The backbone VLM receives only the task description and query example, without any in-context demonstrations.
- **Random**: In-context demonstrations are uniformly sampled from the training set for each query, without retrieval.
- **RICES** [3, 48]: For each query, CLIP image embeddings are computed and the top-K nearest neighbors in the visual feature space are retrieved by cosine similarity. Retrieved examples are sorted by similarity and added to the prompt.
- **MUIER** [32]: Following the original paper, we use a multimodal similarity score combining image-image and image-text similarities, with the same CLIP backbone as RICES. The image-text similarity is computed between the query image and each candidate example's question text. Candidates are ranked by the multimodal score, and the top-K are selected as demonstrations.
- **MMICES** [13]: MMICES is implemented as a two-stage selector. First, 1024 candidate demonstrations are retrieved using image-image similarity. Second, these candidates are re-ranked by a text-image similarity score measuring how well each candidate's image matches the query's question. The top-K re-ranked examples are used as in-context demonstrations.

All baselines are evaluated with a fixed in-context budget of $K = 32$ demonstrations per query, unless otherwise specified in ablations. Demonstrations are selected from the training set of each dataset.

A.2. Implementation of CIRCLES

CIRCLES augments standard image retrieval (IR) with composed image retrieval (CIR) to construct richer and more causally informative in-context examples. For each query, CIRCLES first retrieves visually similar neighbors (IR) using CLIP, and then invokes a training-free CIR module to generate counterfactual examples via language-guided attribute interventions. The final in-context set is composed of both IR and CIR examples, constrained by a fixed total budget of 32 examples unless otherwise stated.

Attribute Extraction. For each query, we prompt the backbone VLM to identify the most prominent attributes visible in the image that are relevant to the given question. The VLM outputs a ranked list of attributes, ordered from most to least important for answering the query. This approach does not rely on explicit attribute labels or fixed vocabularies; instead, attribute phrases are generated dynamically by the VLM based on the image and question context.

Standard image retrieval (IR). The IR component in CIRCLES is identical to RICES: CLIP image embeddings are computed for all training images and the query image, and we retrieve the top-K_{IR} neighbors by cosine similarity. For a full in-context budget of 32 examples, we set K_{IR}=16 and allocate the remaining budget to CIR (e.g., K_{CIR}=16).

Composed image retrieval (CIR) with OSrCIR. We implement CIR using OSrCIR [41], a training-free framework that synthesizes captions conditioned on the query image and attribute-manipulation text. For each selected attribute of a query image, we prompt the VLM to generate a counterfactual caption describing the image with the desired attribute change. This composed caption is encoded with CLIP and used to retrieve images from the training set whose visual embeddings best match the description. As detailed in Section 3, candidates are ranked by the sum of image-image and text-text similarity scores, both normalized to the range $[0, 1]$ via L2-normalization. The top K_{CIR} candidates are selected as CIR examples and combined with IR examples to form the final in-context set.

The detailed prompts used in our implementation are provided in Appendix G.

B. Discussions on Efficiency

Compared to retrieval-only in-context learning baselines such as RICES, MUIER, and MMICES, CIRCLES introduces additional computation at inference time due to attribute extraction and composed image retrieval. For each query, we first invoke the VLM once to identify salient attributes, and then issue one VLM call per selected attribute to generate a counterfactual caption describing the desired manipulation. These calls are in addition to the final answer-generation call, which is shared by all methods. In contrast, standard retrieval baselines perform only CLIP-based retrieval followed by a single VLM call for answer generation.

Although CIRCLES introduces additional VLM calls for attribute extraction and composed retrieval, these calls use short prompts, and the dominant cost still comes from the final in-context inference step with 32 demonstrations. To quantify this overhead, Table 3 reports the average number of tokens processed per question for RICES and CIRCLES across datasets. Overall, CIRCLES adds only about 10% token overhead relative to RICES, while providing consistently better performance in the main experiments.

Table 3. Average token usage per question for RICES and CIRCLES. The additional VLM calls in CIRCLES introduce only a modest overhead (around 10%).

Model	Method	CUB	Flowers	OK-VQA	VizWiz
Gemma3 -4B	RICES	11.1k	10.0k	9.4k	10.2k
	CIRCLES	12.2k	11.2k	10.6k	11.4k
Qwen2.5 -VL-3B	RICES	9.9k	15.8k	12.8k	33.3k
	CIRCLES	10.9k	17.2k	14.1k	35.9k

To further compare CIRCLES with baseline methods under the same VLM call budget, we test a compute-matched variant of RICES, denoted RICES*, which performs multiple answer generations ($K = 3$) followed by self-consistency. As shown in Table 4, CIRCLES consistently outperforms RICES* across datasets and model scales. This suggests that the improvements of CIRCLES are not explained solely by additional generations, but by the quality of the attribute-guided retrieved demonstrations.

Table 4. Compute-matched comparison between CIRCLES and RICES*. RICES* uses multiple generations ($K = 3$) with self-consistency under the same inference-time call budget.

Model	Method	CUB	Flowers	OK-VQA	VizWiz
Gemma3 -4B	RICES*	64.73	86.60	26.34	56.17
	CIRCLES	71.97	93.32	31.27	57.61
Gemma3 -12B	RICES*	75.70	96.02	36.64	74.09
	CIRCLES	77.03	97.77	37.75	74.30

Finally, we view CIRCLES as a first, training-free instantiation of counterfactual retrieval rather than an efficiency-optimized endpoint. Our current implementation relies on an LLM to generate counterfactual captions, which are then encoded with CLIP and used for composed retrieval. As composed image retrieval models and multimodal embedding architectures mature, one could instead directly embed joint (image, manipulation text) queries, or precompute attribute-aware embeddings, thereby reducing or even eliminating the need for multiple online LLM calls. Such designs would bring the computational profile of counterfactual retrieval closer to that of standard similarity-based retrieval, while retaining the robustness and causal benefits demonstrated by CIRCLES.

C. Robustness and Sensitivity

Our main experiments use deterministic decoding with temperature 0.0. To assess robustness under stochastic generation, we additionally evaluate RICES and CIRCLES with Gemma3-4B using temperature 1.0 and repeat decoding five times. Table 5 reports the mean and standard deviation across the benchmark datasets.

Table 5. Robustness under stochastic decoding for Gemma3-4B. We set temperature = 1.0, repeat decoding five times, and report mean $\pm$ standard deviation.

Method	CUB	Flowers	OK-VQA	VizWiz
RICES	64.42 $\pm$0.32	86.42 $\pm$0.24	26.08 $\pm$0.23	55.69 $\pm$0.07
CIRCLES	70.30 $\pm$0.28	92.87 $\pm$0.13	30.54 $\pm$0.21	57.04 $\pm$0.23

CIRCLES consistently outperforms RICES under stochastic decoding on all four datasets. Moreover, the

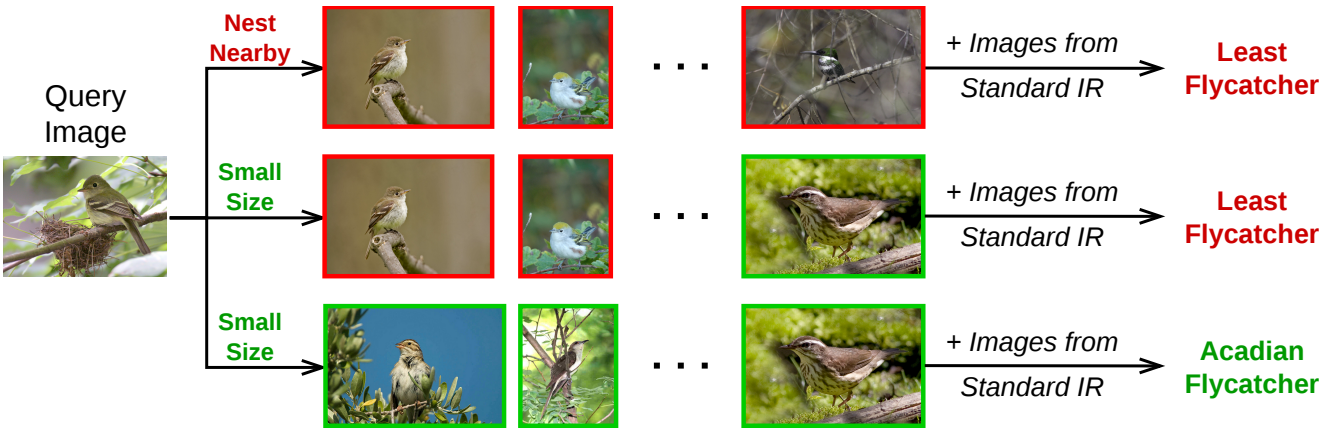

Figure 7. Qualitative examples of CIRCLES. Top: failure due to a non-discriminative extracted attribute. Middle: failure due to weak composed retrieval. Bottom: success case with better matched composed examples. Red denotes unhelpful retrieved examples or incorrect predictions, and green denotes helpful retrieved examples or the correct prediction.

standard deviations are small relative to the performance gaps between the two methods, suggesting that the gains from CIRCLES are stable across repeated stochastic runs rather than driven by favorable sampling variation.

Figure 7 shows two representative failure modes of CIRCLES and one successful case. In the top row, the extracted attribute (*nest nearby*) is visually plausible but not discriminative, so the composed examples provide little useful signal beyond standard IR. In the middle row, the extracted attribute (*small size*) is relevant, but the composed retrieval is only weakly aligned with the target, limiting its benefit for prediction. In the bottom row, the same attribute yields better matched composed examples and leads to the correct answer. These cases suggest that the effectiveness of CIRCLES depends on both reliable attribute extraction and high-quality attribute-conditioned retrieval, which we view as promising directions for future improvement.

D. Effect of Attribute Quality and Counterfactual Retrieval

We first investigate how the quality of attribute information affects the performance of CIRCLES. On CUB, in addition to the attributes extracted by the backbone VLM, we have access to class-specific attribute frequency tables provided by the dataset. For each attribute, we compute a discriminativeness score as the gap between its frequency within a given class and its highest frequency across all other classes, and rank attributes for each class by this score. Then, for a query image, we form a sample-specific attribute list by taking the top-ranked attributes for its class and pruning those that are not annotated as present in the image. This provides a set of clean, image-level attribute descriptions that we feed to CIRCLES in place of VLM-extracted attributes. As shown in Table 6, using these ground-truth attributes yields consistent improvements over VLM-extracted attributes for both Gemma3-4B and Gemma3-12B, confirming that accurate attribute identification is beneficial for composed image retrieval and, consequently, for in-context learning. At the same time, the gains are relatively modest, suggesting that the backbone VLM is already able to recover attributes that are close to the dataset oracle.

Table 6. Effect of replacing VLM-extracted attributes with ground-truth attribute annotations on CUB.

Attribute Source	Gemma3-4B		Gemma3-12B	
	Acc	F1	Acc	F1
LLM	71.97	72.39	77.03	76.90
Dataset	72.44	72.82	77.65	77.46

Next, we explore whether it is the explicit use of attributes or the counterfactual retrieval itself that drives the gains of CIRCLES. To disentangle these factors, we compare three settings on all four benchmarks: (1) standard image retrieval only (IR), which corresponds to the RICES-style baseline; (2) IR with attribute information but without CIR (IR+Attr), where we retrieve images using IR and append the extracted attributes as additional textual context without composing counterfactual queries; and (3) IR with full CIR (IR+CIR), which is our proposed CIRCLES setting where attributes are used to generate counterfactual captions and retrieve manipulated visual examples.

The results for Gemma3-4B are summarized in Table 7. We observe that providing attribute information alone (IR+Attr) substantially improves performance over IR on fine-grained classification tasks such as CUB and Flowers, where attribute recognition is central to the task. In contrast, on open-ended VQA benchmarks like OK-VQA

Table 7. Ablation of CIR versus attribute-only prompting on Gemma3-4B. IR denotes standard image retrieval; Attr denotes adding textual attributes without CIR.

Setting	CUB		Flowers		OK-VQA		VizWiz	
	Acc	F1	Acc	F1	EM	F1	EM	F1
IR	65.40	67.62	86.70	87.43	26.65	32.72	56.08	70.40
IR + Attr	71.99	71.91	92.91	93.53	26.16	32.63	53.83	68.82
IR + CIR	**71.97**	**72.39**	**93.32**	**93.49**	**31.27**	**36.89**	**57.61**	**71.35**

and VizWiz, simply adding attributes without counterfactual retrieval slightly degrades performance, likely because the added textual descriptions can bias the model toward spurious cues without offering new visual evidence. In all cases, enabling CIR on top of IR and attributes (IR+CIR) yields the best performance, with clear gains over both IR and IR+Attr across all datasets. This suggests that attributes are most effective when they are grounded through counterfactual visual examples, which help the model better interpret and utilize fine-grained attribute cues in diverse downstream tasks.

E. More Ablations on CIRCLES Designs

While Section 4.5 ablates several design choices within the CIR module itself, here we further probe the overall CIRCLES framework by examining (i) different implementations of the underlying image retrieval (IR) component and (ii) the relative contributions of IR and CIR.

Alternative IR similarity functions. In the main experiments, we implement IR using image-image similarity in CLIP space, following the RICES design. To understand whether richer similarity measures can further help CIRCLES, we consider two additional variants inspired by prior work. First, we augment image-image similarity with image-text similarity as in MUIER, where we also match the query image against the question text associated with each candidate example. Second, we combine image-image similarity with the text-text similarity used in our CIR implementation to capture task similarity between questions. Since CUB and Flowers are classification tasks with an identical question template for all test images, text-based similarity is uninformative there. We therefore restrict these IR ablations to OK-VQA and VizWiz.

Results in Table 8 show that simply adding image-text similarity does not improve performance, and in fact slightly degrades results on OK-VQA, mirroring the small gap between RICES and MUIER observed in Table 1. In contrast, incorporating text-text similarity on top of image-image similarity yields small but consistent gains on both datasets, especially on VizWiz. This suggests that, for open-ended VQA, capturing task similarity at the text level is beneficial, while the simple image-image retrieval used in our main CIRCLES configuration is already a strong baseline.

Table 8. CIRCLES with different implementations of the image retrieval component (Gemma3-4B).

Similarity	OK-VQA		VizWiz	
	EM	F1	EM	F1
image-image	31.27	36.89	57.61	71.35
image-image + image-text	30.14	35.67	57.81	71.63
image-image + text-text	31.37	36.74	59.09	72.64

IR versus CIR versus IR+CIR. We also disentangle the effects of IR and CIR by comparing three variants: (1) *IR only*, which corresponds to a standard retrieval-based ICL setup using only retrieved images and their labels; (2) *CIR only*, where we discard the original retrieved examples and retain only the counterfactual examples produced by CIR; and (3) *IR + CIR* (full CIRCLES), which includes both the original retrieved examples and their counterfactual counterparts in the in-context prompt. The results on Gemma3-4B are summarized in Table 9.

On fine-grained classification benchmarks (CUB and Flowers), CIR only lags far behind IR only, indicating that counterfactual examples alone are not sufficient to capture the subtle visual prototypes needed for class recognition. In contrast, on OK-VQA and VizWiz, CIR only achieves performance comparable to IR only, slightly improving EM and F1 on OK-VQA while being close on VizWiz. Across all four datasets, however, the combined IR + CIR variant consistently yields the best performance, sometimes by a large margin (e.g., +6-7 accuracy points over IR only on CUB and Flowers). These trends highlight that IR and CIR play complementary roles: IR provides realistic, prototypical examples that anchor the model's understanding of the task, while CIR introduces targeted counterfactual variations that clarify the role of key attributes and reduce spurious correlations. Together, they enable CIRCLES to leverage both factual and counterfactual experience for more robust in-context learning.

Table 9. Ablation of IR and CIR in CIRCLES, tested on Gemma3-4B.

Setting	CUB		Flowers		OK-VQA		VizWiz	
	Acc	F1	Acc	F1	EM	F1	EM	F1
IR only	65.40	67.62	86.70	87.43	26.65	32.72	56.08	70.40
CIR only	25.16	26.14	60.24	65.85	29.81	35.52	54.06	68.29
IR + CIR	**71.97**	**72.39**	**93.32**	**93.49**	**31.27**	**36.89**	**57.61**	**71.35**

F. Additional Results on Benchmarks and Model Scaling

Results on ScienceQA. To further examine the generality of CIRCLES beyond image classification and visual question answering tasks, we additionally evaluate CIRCLES on ScienceQA [30], a more challenging multimodal reasoning benchmark. Following the same in-context learning protocol used in the main paper, we construct the demonstration pool from the ScienceQA validation split and report performance on the test split. Table 10 shows that CIRCLES consistently outperforms all compared example selection baselines on ScienceQA. The gains are modest but consistent across both model sizes.

Table 10. Test accuracy (%) on ScienceQA. In-context examples are retrieved from the validation split. The experiments are run on the Qwen2.5-VL model family.

Size	Random	RICES	MUIER	CIRCLES
3B	78.33	80.12	80.66	**80.71**
7B	85.18	88.05	87.90	**88.35**

Scaling to Larger Models: Gemma3-27B. To examine whether the gains of CIRCLES persist at a larger model scale, we additionally evaluate Gemma3-27B, a larger model in the Gemma3 family, on the same four benchmarks used in the main paper. Table 11 shows that CIRCLES continues to outperform RICES across all datasets, indicating that the benefit of attribute-conditioned, contrastive retrieval is not limited to smaller or mid-sized backbones.

Table 11. Performance comparison on Gemma3-27B.

Method	CUB	Flowers	OKVQA	VizWiz
RICES	69.76	93.98	38.66	67.24
CIRCLES	**72.21**	**97.51**	**39.14**	**68.72**

G. Prompt Templates

We provide the detailed prompt templates used for each dataset in this section. Figure 8 shows the template used to extract key visual attributes that are relevant for answering the question. Given an input image and question, the VLM is instructed to list a small set of concise attributes, which are then used as the basis for constructing counterfactual manipulations. Figure 9 contains the template used to generate counterfactual captions with targeted attribute changes. The system prompt in Figure 9 follows the one used in OSrCIR [41].

Figures 10–12 show the templates used for VLM inference under different in-context learning settings: Figure 10 corresponds to the setting without in-context examples ("None" in Table 1); Figure 11 illustrates the baseline prompt used for in-context learning methods such as RICES; and Figure 12 presents the full CIRCLES prompt, which augments the standard retrieved demonstrations with the counterfactual examples produced by CIR. Across all of these templates, the value of `{{Task Type}}` is set to "Image Classification" for CUB and Flowers, and to "Visual Question Answering" for OK-VQA and VizWiz, so that the VLM is explicitly informed of the task format.

For the classification datasets (CUB and Flowers), we additionally enforce a closed set of answer options to align with the standard evaluation protocol. Concretely, for these datasets, we append the sentence "You need to choose one of the following options: `{{Options}}`" immediately after the first sentence describing the task type, where `{{Options}}` is replaced by the list of candidate class names for the given example. For OK-VQA and VizWiz, which are evaluated in an open-ended manner, we do not provide such options and instead allow the model to freely generate answers conditioned on the image, question, and in-context demonstrations.

Prompt template for attribute extraction

Identify the key attributes of the following image that are most relevant to answering the question.

`{{Image}}`
Question: `{{Question}}`

Please list the top `{{num_attributes}}` key attributes as short phrases in a section named '### Attributes', one per line, ordered from most to least important.

Figure 8. Prompt template for attribute extraction.

Prompt template for generating counterfactual caption

`{{System Prompt by OSrCIR}}`

`{{Image}}`

Manipulation Text: Change the attribute `{{Attribute}}` to a different plausible value. Ensure the modified caption is concise and contains no more than 77 tokens.

Figure 9. Prompt template for generating counterfactual caption based on the query image and identified key attribute.

Prompt template for VLM inference used by None

Your task is to perform `{{Task Type}}`.

`{{Image}}`
Question: `{{Question}}`

Please provide your response by directly outputting the answer.

Figure 10. Prompt template for VLM inference used by None (zero-shot learning).

Prompt template for VLM inference used by Random/RICES/MUIER/MMICES

Your task is to perform `{{Task Type}}`.

`{{Image}}`
Question: `{{Question}}`

Here are `{{K_IR}}` in-context examples to help you answer the question:

`{{Retrieved Image}}`
Question: `{{Retrieved Question}}`
Answer: `{{Retrieved Answer}}`

`{{Retrieved Image}}`
.....

Here is the original question again.
`{{Image}}`
Question: `{{Question}}`

Please provide your response by directly outputting the answer.

Figure 11. Prompt template for VLM inference used by Random, RICES, MUIER, and MMICES.

Prompt template for VLM inference used by CIRCLES

Your task is to perform `{{Task Type}}`.

`{{Image}}`
Question: `{{Question}}`

Here are `{{K_IR}}` in-context examples to help you answer the question:

`{{Retrieved Image}}`
Question: `{{Retrieved Question}}`
Answer: `{{Retrieved Answer}}`

`{{Retrieved Image}}`
.....

Examples retrieved based on the target image description after changing `{{Attribute}}` (caption: `{{Caption}}`):

`{{Retrieved Image}}`
Question: `{{Retrieved Question}}`
Answer: `{{Retrieved Answer}}`

`{{Retrieved Image}}`
.....

Here is the original question again.
`{{Image}}`
Question: `{{Question}}`

Please provide your response by directly outputting the answer.

Figure 12. Prompt template for VLM inference used by CIRCLES.