Cache-of-Thought: Master-Apprentice Framework for Cost-Effective Vision Language Model Inference

Anonymous ACL submission

Abstract

Vision Language Models (VLMs) have achieved remarkable success in a wide range of vision applications of increasing complexity and scales, yet choosing the right VLM model size involves a trade-off between response quality and cost. While smaller VLMs are cheaper to run, they typically produce responses only marginally better than random guessing on benchmarks such as MMMU.

In this paper, we propose *Cache of Thought (CoT)*, a master–apprentice framework for collaborative inference between large and small VLMs. CoT manages high-quality query results from large VLMs (*master*) in a cache, which are then selected via a novel multi-modal retrieval and in-context learning to aid the performance of small VLMs (*apprentice*). We extensively evaluate CoT on various widely-recognized and challenging general VQA benchmarks, and show that CoT increases overall VQA performance by up to 7.7% under the same budget, and specifically boosts the performance of apprentice VLMs by up to 36.6%.

1 Introduction

Recent Vision Language Models (VLMs) (OpenAI, 2024b; Anthropic, 2024; Gemini, 2024) have shown tremendous promise in a wide range of real-world applications, such as autonomous driving (Zhou et al., 2024; Yuan et al., 2024), robotics (Brohan et al., 2023; Duan et al., 2024; Gao et al., 2024a), personalized virtual assistants (Nguyen et al., 2024), search engines (Zhang et al., 2024) and recommendation (Liu et al., 2024). However, the ever-growing size of these recent VLMs has made at-scale deployment and operation challenging due to high consumption of cloud computing resource, high latency, and expensive API calls.

In response to this, there has been research focusing on developing smaller VLMs for on-device capabilities or cheap inference, such as MobileVLM-1.7B (Chu et al., 2023, 2024), GPT-4o-mini (OpenAI, 2024a) and Qwen-VL 7B (Wang et al., 2024) etc. Unfortunately, one could not simply replace the larger VLM with a smaller one and expect the result to be within some practical tolerance, as the performance gap between large and small VLMs is still too *huge* (Figure 1): compared to the impressive performance of large VLMs, some smaller VLMs offer only marginal improvements over random guessing on challenging benchmarks such as MMMU (Yue et al., 2024).

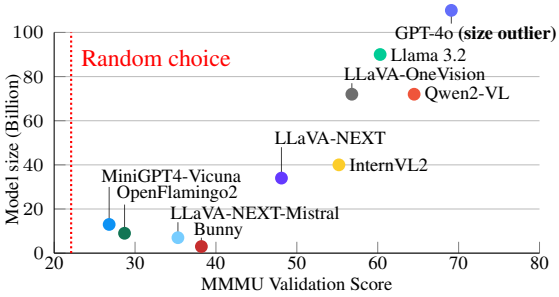


Figure 1: Model Performance Comparison on MMMU(Val) as of Nov. 30, 2024(Yue et al., 2024).

In this paper, we seek a middle ground between smaller and larger models through collaboration between models of different sizes. To enhance on-the-fly performance of smaller models for a cost-effective system, we propose *Cache of Thought (CoT)*, a master-apprentice framework that enables small (*apprentice*) VLMs to generate responses of significant closer quality versus large (*master*) VLMs via a novel design of dynamic *Cache*. The Cache stores historical high-quality answer responses generated by master VLMs, which serves as a guidance for apprentice VLM query answering. Inspired by the principle of well-established case-based reasoning (Aamodt and Plaza, 1994), CoT performs this guidance from master to apprentice via a specialized form of Retrieval-Augmented Generation (RAG): when CoT selects apprentice VLMs for question answering, it retrieves the most

similar historical queries and responses in the cache to provide to the apprentice VLM as in-context examples. Through in-context learning, apprentice can benefit from the accumulated cached history of master, and become more capable of handling new queries. Notably, since the cache is allowed to grow, it is expected that the quality and relevancy of in-context samples should grow as well, leading to further improved capabilities of the apprentice VLM and the overall system. From the practicality perspective, CoT's in-context learning incurs negligible cost: it prepends the retrieved queries to the inference prompt of apprentice VLMs without introducing any extra training workloads, additional data annotations, nor recomputation expenses. CoT is also complementary to other VLM serving strategies and highly adaptable: as VLM routing and selection strategies mature, CoT can be easily integrated, continuing to offer significant benefits to smaller VLMs.

To the best of our knowledge, CoT is the first framework that demonstrates VLM in-context learning, combined with multimodal retrieval, can be applied effectively to real-world VQA tasks with lengthy question prompts (e.g., MMMU (Yue et al., 2024)). In comparison, previous works explored VLM in-context learning under more constrained conditions, such as short textual image descriptions or questions of several words (e.g. "what is in the image") (Alayrac et al., 2022; Jiang et al., 2024), purely visual contexts (Zhang et al., 2023), or synthetic datasets with short, simple questions (Zong et al., 2024). These works often relied on straightforward top-k image embeddings (or even random selection) for selecting in-context examples and utilized ground-truth (rather than model-generated) responses. Prior to this work, it was unclear which retrieval methods were most appropriate for dual-modality similarity, especially for complex VQA questions and in-context learning scenarios. CoT proposes and systematically evaluates various retrieval techniques (e.g., CLIP-based text+image embeddings, keyword extraction, keyword clustering), and maintains a dynamic cache of master VLM-generated responses, thereby expanding the scope and effectiveness of in-context learning for general, real-world VQA tasks without using ground truths.

In summary, we contribute (1) CoT, a general master-apprentice framework for multi-VLM inference with a cache, (2) an effective multi-modal retrieval and in-context learning approach that boosts apprentice VLM performance on the fly without training, and (3) extensive experiments demonstrating the effectiveness of CoT's multi-modal in-context learning and desirability of CoT's cost-response quality trade-off for general VQA.

2 Related Work

RAG and Multi-modal RAG. RAG was originally proposed for language tasks (Lewis et al., 2020). In RAG, a retriever is designed to extract relevant document chunks by similarity, and then these chunks are prepended to question prompts. RAG can reduce hallucinations, support knowledge-intensive tasks (Gao et al., 2024b), and improve capability of small models (Mialon et al., 2023). Recent research has expanded RAG's capabilities beyond text. Multimodal RAG (Chen et al., 2024; Lin et al., 2024a,b; Hu et al., 2022; Yasunaga et al., 2023) retrieves world knowledge from relevant multimodal documents and improves knowledge-seeking or fact-based VQA tasks (Wang et al., 2018; Marino et al., 2019; Chen et al., 2023; Mensink et al., 2023). CoT differs from multi-modal RAG in that its cache is dynamic, and all stored responses are generated by a large master VLM rather than factual documents.

In-context Learning and Multi-modal In-context Learning. In-context learning emerged with large auto-regressive models like GPT-3 (Brown et al., 2020), which adapt to new tasks by observing a few in-context demonstrations. This success motivated research into VLMs. Flamingo (Alayrac et al., 2022) demonstrated significant gains in VQA by providing random demonstrations in context, while Zhang et al. (2023) extended these benefits to tasks like image segmentation and classification. More recently, Zong et al. (2024) validated in-context learning for VLMs ranging from 4B to 70B parameters using synthetic VQA. CoT makes unique contribution in multi-modal in-context learning, with we detail in Section 4.2.

3 Cache-of-Thought Framework

CoT (Figure 3) is a query serving framework for VLM queries that interleaves large (*master*) and small (*apprentice*) VLM calls for significant performance boost and cost savings: query results from master VLM calls are cached, which are then used to enhance apprentice VLM calls via multi-modal retrieval and in-context learning.

Master VLM. CoT's master VLM (often with several hundred billion parameters and serves with

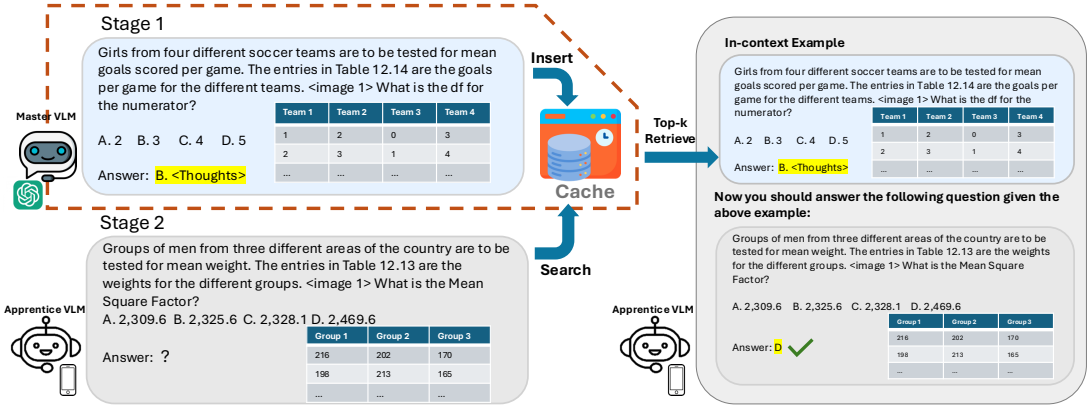


Figure 2: Apprentice VLM answers new query with help of past similar cases answered by the master VLM: *Multi-modal Retrieval and In-context learning*. Images and prompt examples cited from MMMU (Yue et al., 2024)

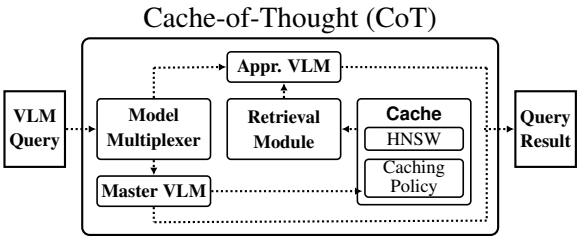


Figure 3: The Cache-of-Thought (CoT) framework.

per-token charge), under the assumption that it produces high-quality answers, acts as the generator of the QA-pairs stored in the cache.

Apprentice VLM. CoT's apprentice VLM (often with less than 7 billion parameters) is used to answer queries via augmentation with various in-context examples fetched from the cache.

Model Multiplexer. The Model Multiplexer routes incoming VLM queries by choosing one of two serving methods: use the master VLM to directly answer the query, or use the apprentice VLM to answer the query augmented with in-context examples fetched from the Cache (described shortly). The multiplexer balances cost and quality by controlling how often CoT calls the master and apprentice VLMs: frequently calling the master VLM increases the rate of populating examples into the cache (hence more effective in-context learning) but incurs a higher cost, and vice versa.

Cache. CoT's cache stores high-quality QA pairs from the master VLM. When the multiplexer routes a new query to the master VLM, the question asked, image prompt, and resulting answer (QA-pair) is stored into the cache, which then computes its dual-modality embedding from the question and image and inserts it into a HNSW index (Malkov and Yashunin, 2020) to facilitate accurate retrieval as an in-context example. CoT's cache can also be preloaded with QA-pairs to efficiently warm-start on query serving, and can incorporate common eviction policies (e.g., LRU or LFU) for more resource-constrained settings.[1]

Retrieval Module. The Retrieval Module retrieves relevant in-context examples (i.e., QA-pairs) from the Cache to augment queries routed to the apprentice VLM by the Multiplexer. It performs an efficient ANN vector search with the dual-modality embedding of the incoming query in the Cache's HNSW index, retrieving examples similar to the query in terms of both the question asked and image prompt.

4 Methodology

This section describes CoT's methodology. Section 4.1 overviews CoT's formulation, Section 4.2 describes CoT's multi-modal in-context learning, and Section 4.3 covers how CoT retrieves relevant examples for in-context learning.

4.1 Formulation

This section formulates CoT's notations. Without loss of generality, CoT deploys two VLM instances with frozen weights, the master VLM f and apprentice VLM g, where deployment cost of g is significantly lower than F. CoT's workload is a set of (finite) queries $\mathcal{Q} \subset \mathbb{R}^d$: At each round t, the system receives a query from the workload $(x_t, q_t) \in \mathcal{Q}$ consisting of the visual input x_t and the textual question q_t (about the content of x_t e.g., 'how many ducks are there in the picture?').

For each query (x_t, q_t) from round t, CoT's model multiplexer J determines which VLM to use (the master f or apprentice g) with the mul-

[1]We defer concrete explorations on CoT's eviction policy to future work when real-world dual modality user query benchmarks are available.

tiplexer function: $J : \mathcal{Q} \mapsto \{f, g\}$, which routes the query according to $J((x_t, q_t))$. Then, the selected VLM will serve the query by generating an answer from an answer space $\mathcal{A}$ with the respective function $f : \mathcal{Q} \mapsto \mathcal{A}$ or $g : \mathcal{Q} \mapsto \mathcal{A}$, e.g., $g((x_t, q_t)) = a_t \in \mathcal{A}$. x_t, q_t, and the answer from the master/apprentice VLM a_t will then comprise the *QA-pair* $P_t = \{(x_i, q_i), a_i\}$.

CoT's cache, denoted as $\mathcal{L}_t \subset \mathcal{Q}$, will store $|\mathcal{L}_t| \leq L$ QA-pairs P_t generated with g, i.e., P_t s.t. $J((x_t, q_t)) = g$, where L is the cache capacity. CoT inserts into $\mathcal{L}_t$ whenever g is invoked; however, due to the bounded size L, insertions performed when $|\mathcal{L}_t| = L$ will trigger evictions. Initially, $\mathcal{L}_0$ can either be empty (cold start) or a prepopulated set of QA-pairs (warm start) from the master VLM g. Hence, the cache $\mathcal{L}_t$ serves as an external knowledge base to support the apprentice VLM f during inference via multi-modal retrieval and in-context learning (Section 4.2).

For retrieval from the Cache $\mathcal{L}_t$, CoT uses a *retrieval mechanism* R to retrieve the top-k relevant QA-pairs $\mathcal{L}_k$ from the cache: $R(\mathcal{L}_{t_i}, (x_{t_j}, q_{t_j})) \mapsto \mathcal{L}_k \subseteq \mathcal{L}_{t_i}, j > i$, which we describe in Section 4.3.

4.2 Multi-modal In-context learning

CoT performs multi-modal in-context learning by transferring knowledge from the cache to the apprentice VLM. Given the pre-trained apprentice VLM f with frozen parameters θ and a query $(x_t, q_t) \in \mathcal{Q}$, CoT retrieves a set of QA-pairs from the cache $M = \{(x_i, q_i), a_i\}$ from $\mathcal{L}_t$ used to aid inference. The apprentice VLM f then generates an answer a_t in one forward pass: $a_t = f_\theta((x_t, q_t), M)$.

Flexibility for Dynamic Systems. Compared to finetuning apprentice VLM instances with Lora using cached content (Hu et al., 2021), CoT's in-context learning offers several advantages in dynamic systems: (1) CoT pre-computes QA-pairs and stores them in the cache for reuse, incurring no extra data annotation and/or recomputation expenses during retrieval; hence, CoT's performance will steadily improve with minimal costs as it answers more high-quality queries with the master VLM. (2) CoT's self-improvement via in-context learning does not require weight updates for its VLMs, hence there are no extra training overheads or delays (e.g., waiting for model convergence).

Prompt Construction. For each query routed to the apprentice VLM, CoT performs in-context learning by retrieving one or more demonstration examples to prepend to the inference prompt. CoT utilizes special tokens for prompt formatting, interleaving support QA-pairs in the form of (image,text): "{support image}, Question: {support question}, Answer: {*ground truth answer*}, {query image}, Question: {query question}, Answer:" In CoT, {*ground truth answer*} is replaced by stored VLM responses from the cache $\mathcal{L}_t$. Beyond this core prompt structure, additional components—such as n-shots and system prompts required by the VLM's input format—are incorporated in the final prompt. For effective prompt construction, CoT explicitly instructs the master VLM to "include reasoning steps" when using it to serve queries to obtain complete responses (to prepend to prompts sent to the apprentice VLM) rather than just short answers or multiple-choice selections. This allows the apprentice VLM to better leverage the master VLM's reasoning capabilities. Full prompts details can be found in the Appendix A.

4.3 Multi-modal Retrieval

In this section, we describe CoT's multimodal retrieval of in-context examples from its Cache, on which performance on downstream tasks is highly sensitive to (Zhang et al., 2023). Much to our surprise, existing works on VLM in-context learning present ungrounded design choices without further experiments in the following aspects:

Choice of Retrieval. Recent works (Zong et al., 2024; Jiang et al., 2024) randomly select in-context examples for all test instances. Flamingo (Alayrac et al., 2022) attempts retrieval-based in-context example selection, which retrieves top-k similar samples using frozen pre-trained vision encoder embeddings from CLIP (Radford et al., 2021). While Flamingo verifies that its image-driven retrieval is better than random selection in test cases with short, few-word questions, it remains unclear whether this method can be extend to general VQA systems where questions can be long and complicated, as seen in benchmarks like MMMU (Yue et al., 2024).

Choice of Support Set. All of the aforementioned studies randomly partition the dataset into a support set and test set offline. However, in-context performance can heavily depend on the distribution of support examples. A setting where the support set is dynamic (i.e., continuously growing as in CoT's case) has not been explored.

Assumption of Ground-Truth Availability. Previous studies often use in-context examples with human-annotated ground truth answers, which can

be inpractical for real-world, online data streams.

In contrast, CoT targets a more challenging and general VQA setting, and makes significant contributions to multi-modal retrieval for in-context learning in terms of the aforementioned 3 aspects: (1) CoT proposes a retrieval method that leverages both CLIP image and text embeddings, supplemented by keyword-based techniques for long or complex queries. CoT also introduces a hierarchical keyword clustering approach tailored to the query distribution (Section 4.3.1). (2) CoT's in-context example is performed over a dynamically updating (as opposed to static in previous works) knowledge base $\mathcal{L}_t$, where new items (that align more closely with recent queries) are continuously added to the cache. (3) CoT directly utilizes responses from the master VLM as in-context examples rather than the human-labeled ground truth answers strongly assumed by all prior works. This allows CoT to incorporate more diverse examples as long as they have been seen by the master VLM, rather than relying on human annotators.

4.3.1 Dual-Modality Dense Retrieval

We begin with a simple dense-retrieval method that uses a unified embedding vector to identify the top-k matches. A natural starting point is CLIP (Radford et al., 2021), which has proven effective in multimodal retrieval (Yasunaga et al., 2023; Vendrow et al., 2024) and is pretrained to align image and text embeddings on internet-scale (image, text) pairs. Recognizing CLIP as a well aligned dual-modality representation, we use its image and text encoders to separately embed each image and text, then average these embeddings to form a unified vector that is stored in our cache index and use for query embeddings.

However, CLIP embeddings also have limitations. They are pretrained primarily on short textual descriptions of images (e.g., "a photo of a cat") and support a relatively small text context window—often fewer than 20 tokens effectively (Zhang et al., 2025). This limitation makes CLIP less ideal for lengthy, sentence-like questions. To mitigate this issue, we break down long questions before encoding them with CLIP. Specifically, we employ a small auxiliary LLM, denoted as $Extractor$ to extract keywords from the long sentence-like query. Details of how we prompt the $Extractor$ can be found in the Appendix.

In some cases, the question text alone may not be sufficiently informative for retrieval (e.g. a question like "what is in the image"); Fortunately, CoT's cache also stores responses from the master VLM, allowing us to extract text embeddings from these richer responses instead of relying solely on the original question. For new queries, we can even make a single pass with the apprentice VLM to generate a text answer response, extract keywords from that response, then feed them into CLIP text encoder to enhance retrieval. Because these auxiliary or apprentice LLM calls are much cheaper than invoking the master VLM, we can use them to achieve significantly better retrieval results without noticeably increasing inference cost. This strategy, which combines embeddings from both the text response and the corresponding image, significantly improves the robustness of our dense-retrieval process, particularly in VQA scenarios where the question alone lacks sufficient context.

4.3.2 Multi-stage Hashtag Retrieval

In real-world scenarios, the streaming data distribution $\mathcal{Q}$ is highly complex and diverse, varying across different contexts. When the cache is large, directly performing dense retrieval from the entire knowledge base $\mathcal{L}_t$ may fail to locate the relevant contexts due to the limited expressiveness of CLIP embeddings even if they are carefully designed (Section 4.3.1). A straightforward approach to improve retrieval accuracy is to restrict retrieval to a specific knowledge domain. However, without a predefined distribution for $\mathcal{Q}$, categorizing streaming data under fixed domains is often difficult.

To enable more granular unsupervised retrieval, we design a two-stage hashtag tree where Level 1 hashtags $\mathcal{H}_1$ capture high-level, task-oriented representations, and Level 2 hashtags $\mathcal{H}_2$ provide finer, concept-specific details. Each data entry in $\mathcal{L}_t$ is assigned at least one Level 1 hashtag and one Level 2 hashtag. We use $h_{i,j}$ to denote the jth hashtag at Level i, where $\mathcal{H}_i = \{h_{i,j}\}, i = 1, 2$.

Initially, both the hashtag tree and the cache $\mathcal{L}_t$ are empty. The master VLM g is tasked with inferring a_t given each new-coming data (x_t, q_t) during the cold start stage, after which the Level 2 hashtag $h_{2,t}$ is obtained as: $h_{2,t} = CLIPTextEncoder(Extractor(a_t))$. Once $|\mathcal{H}_2| > \tau$ or the cold start ends, we use K-Means to cluster $\mathcal{H}_2$ into K groups $\{C_k\}_{k=1}^{K}$, where each cluster is assigned a Level 1 hashtag $h_{1,k}$, computed as the mean embedding of its corresponding Level 2 hashtags.

Given new data t', when the master VLM g is se-

lected for inference, the generated answer is stored in $\mathcal{L}_t$ and assigned to the closest Level 1 hashtag h_{1,j^*} based on Euclidean distance. The Level 1 hashtag is then updated as the weighted average of its current Level 2 hashtag children. Otherwise, if the apprentice VLM f is selected to infer on t', $\mathcal{R}$ first retrieves all examples from $\mathcal{L}_t$ that shares the same Level 1 hashtag as d_t. It then selects top-k results as the k-shot in-context examples for f.

Table 1: Qwen7B MMMU Score vs. Retrieval Strategy

Method			Cache Dev Test Val	Cache Dev Test Val
No in-context			24.66	34.42
Hierarchical Cluster based			**39.04**	**39.44**
GPT-4o			58.90	64.87
Cache	Query	Cache size		
Image	Image+ Text	Full	35.62	39.20
Image	Image+ Text	Half	34.25	38.74
Image+ Text	Image+ Text	Full	34.90	38.39
Image	Image	Full	37.67	37.92

5 Experiment

This section describes our experimental evaluation of CoT. We design our experiments as follows: (1) We evaluate the performance gains on general VQA tasks achieved by CoT's applying of in-context learning to apprentice VLMs via multi-modal retrieval. (2) We evaluate the sensitivity of CoT to different system configurations such as cache size and choice of retrieval methods. (3) We evaluate CoT in both static and dynamic configurations (explained shortly). (4) We study CoT's trade-offs between performance and cost advantage (metric defined by (Ding et al., 2024)) by adjusting the ratio of master and apprentice VLM usage.

Specifically, for (3), in the static configuration, We pre-construct $\mathcal{L}_t$ using offline data, which remains unchanged in each round t. We also fix the model multiplexer J to only select the apprentice VLM. These settings allow us to evaluate how much the apprentice VLM potentially benefits from CoT under a static cache. In the dynamic configuration, CoT processes a continuous data stream and uses a dynamic cache $\mathcal{L}_t$ (explained shortly) which may start either empty or containing offline data. The model multiplexer randomly selects between the apprentice and master VLM according to a specified probability (e.g., a 10% apprentice VLM rate means that it is selected for query serving 10% of the time). After each round, CoT inserts into the cache with the new QA-pair (when the master VLM is used) and may perform eviction (when the cache is at capacity) to maintain cache effectiveness for downstream tasks. This setup allows us to evaluate both CoT's performance-cost trade-offs and capability to adapt over time.

5.1 Experiment Datasets

MMMU (Yue et al., 2024) is a widely recognized and challenging multi-disciplinary VLM benchmark that organizes subjects into a hierarchical structure. It evaluates VLMs on both breadth and depth of subject knowledge, as well as on their expert-level reasoning and understanding. MMMU covers complex VQA tasks across six major fields, ranging from art to science and businesses. The dataset consists of 11.5K questions, divided into development (*dev*), validation (*val*), and test sets (the test set does not include ground truth answers). To ensure fair evaluation of in-context learning, we filter out instances containing multiple images, as most off-the-shelf VLMs are less capable of handling multi-image questions, which could interfere with their in-context learning capabilities. After filtering, the sizes of the dev, val and test set are 146, 857, 9702 respectively.

VL-ICL (Zong et al., 2024) is a pioneering synthetic benchmark designed to measure the broader capabilities and limitations of multi-modal in-context learning. We employ this dataset to examine whether CoT retains the benefits of in-context learning even when only responses generated by the master VLM instead of ground-truth annotations are used. We specifically adopt the TextOCR and Clevr subtasks, as the remaining subtasks lack meaningful question prompts in general language settings. The data set is divided into two subsets of 200 and 800 samples, which we refer to as dev and val, respectively, in this paper.

5.2 Models

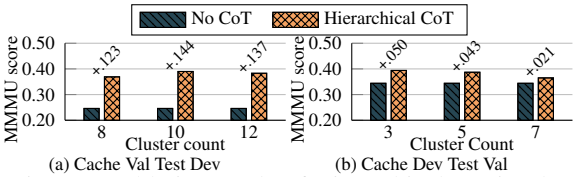
We choose between the open-source Qwen-VL-2 7B (Wang et al., 2024) model as a representative of recent VLMs, and open-flamingo 3B and 9B (Alayrac et al., 2022) (from an earlier generation of VLMs) as CoT's apprentice VLM in various experiments, to evaluate whether CoT can benefit older VLMs and newer, well-instruction tuned VLMs alike. We run all apprentice models with 4 A100 GPUs of 40GB. We use GPT4-o (OpenAI, 2024b) as CoT's master VLM.

Table 2: OpenFlamingo w. Cache Val Test Dev Setting

Model	OpenFlamingo-3B (CLIP ViT-L/14, MPT-1B-Dolly)					OpenFlamingo-9B	
Cache size	N/A	Full Cache Size		Half Cache Size		Full Cache Size	
N-shot / Dataset	0-shot	1-shot	2-shot	1-shot	2-shot	0-shot	1-shot
mmmu	15.75	17.12		17.12		20.55	25.34
clevr	7.50	20.50	28.00	22.50	26.50		
textocr	0.00	6.50	5.50	6.50	5.50		

Table 3: OpenFlamingo w. Cache Dev Test Val Setting

Model	OpenFlamingo-3B (CLIP ViT-L/14, MPT-1B-Dolly)					OpenFlamingo-9B	
Cache size	N/A	Full Cache Size		Half Cache Size		Full Cache Size	
N-shot / Dataset	0-shot	1-shot	2-shot	1-shot	2-shot	0-shot	1-shot
mmmu	22.75	23.92		24.04		24.97	26.25
clevr	6.88	21.50	21.75	19.38	19.38		
textocr	0.00	5.00	3.38	4.63	2.13		



(a) Cache Val Test Dev (b) Cache Dev Test Val

Figure 4: Ablation Study of Hierarchical Retrieval.

5.3 Evaluation

We use accuracy as the evaluation metric. We note that the main purpose of our evaluations is to present a holistic picture of how CoT would work for end users, thus the absolute performance of the models are not the priority. For both multiple-choice answers in MMMU and open responses in VL-ICL, we allow the VLMs to produce intermediate steps before generating the final answer. We prompt the VLMs to produce the final answer in a specific format, and use a rule-based parser to determine the VLMs' choice. More specifically, for multipe-choice answers, we extract the VLMs' choice, and for open responses, we check whether there exists an exact match of the label in the VLMs' response. If the parser fails to determine the VLMs' choice, we assign a `None` value instead, since in reality, an randomly generated choice is misleading if not detrimental.

6 Analysis

6.1 Static Performance

This section preliminarily validates the effectiveness of CoT's multi-modal in-context learning in the static cache setting. We present our findings in Table 1, which reports Qwen-7B's performance on the challenging MMMU dataset when enhanced by CoT's various multi-modal retrieval strategies. Tables 2 and 3 report OpenFlamingo's performance on both MMMU and VL-ICL benchmarks when similarly enhanced by CoT. We experiment with two caching strategies on these datasets: storing a larger validation set (Val) while testing on a smaller development set (Dev), and vice versa.

More Benefits with Larger Cache. As shown in Table 1, caching a larger split while testing on a smaller split leads to greater performance gains when applying in-context learning on smaller models. A similar trend is observed when we manually restrict the cache size to half its original capacity.

Hierarchical Retrieval Outperforms Dense Retrieval, but at a Cost. Our experiments aim to determine which retrieval method performs better and is more promising for in-context example selection. As shown in Table 1, the hierarchical retrieval method slightly outperforms the dual-modality dense retrieval approach. However, this marginal advantage can only be achieved after fine-tuning a sensitive hyperparameter (cluster number). Figure 4 details more related ablations.

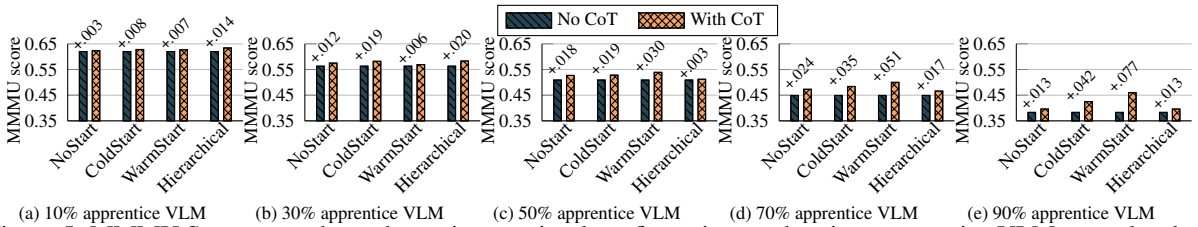
Best Dense Retrieval Strategy. We find that caching image embeddings, and querying with the average of image and text embeddings encoded from keywords extracted from VLM responses is the most effective dense retrieval strategy. Interestingly, (1) pure image retrieval, as reported in the OpenFlamingo paper (Alayrac et al., 2022), also performs reasonably well, and (2) when the test set is relatively small (potentially introducing bias), pure image retrieval can even outperform other dense retrieval methods. In Table 1, all presented text-based dense retrieval results utilize our LLM keyword extractor (Section 4.3.1), as the CLIP embedding context window is often too small to encompass complete VLM responses.

Based on these findings, we choose to evaluate our hierarchical retrieval method and the best-performing dense retrieval strategy in our following more important dynamic experiments (Section 6.2).

Openflamingo-3B and 9B Receive Significant Benefits from CoT. We present results for OpenFlamingo combined with the best dense retriever in Tables 2 and 3. CoT achieves high performance gains across the OpenFlamingo series, on both datasets, varying cache capacities, and the number of in-context examples. In MMMU, OpenFlamingo struggles to accommodate two-shot GPT-4o responses due to shorter context length, which limits the effectiveness of in-context learning.

6.2 Dynamic Performance

This section studies CoT's trade-offs between performance and cost advantage in the dynamic cache setting with various cache sizes. Figure 5 reports



(a) 10% apprentice VLM (b) 30% apprentice VLM (c) 50% apprentice VLM (d) 70% apprentice VLM (e) 90% apprentice VLM

Figure 5: MMMU Score w. cache under various retrieval configurations and various apprentice VLM usage levels.

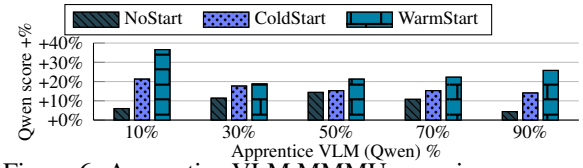


Figure 6: Apprentice VLM MMMU score increases vs. Cache and Usage

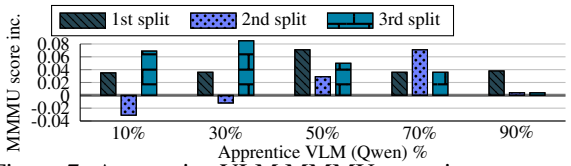


Figure 7: Apprentice VLM MMMU score increases vs. Stage and Usage

the MMMU score versus cache and retrieval configurations (within each sub-plot) and percentage of apprentice VLM usage (between each sub-plot). For *NoStart*, *ColdStart* and *WarmStart*, we initialize CoT's cache as an empty set, the dev set, and the test set, respectively. The *Hierarchical* setting uses the hierarchical retrieval method with CoT's cache startting with the dev set. Figure 6 presents the performance gains of CoT's apprentice VLM versus different cache initialization methods. Figure 7 shows the MMMU score increase of CoT's apprentice VLM on the first, second, and third (temporal, equal-sized) splits of the validation set.

Disparity Between Higher and Lower Use of Apprentice VLMs. The less frequently apprentice VLMs are used, the better the overall performance. However, CoT mitigates this performance loss: as shown in Figure 5, with *WarmStart* and CoT enabled, the MMMU score with 90% apprentice VLM usage matches the performance without CoT with only 70% apprentice VLM usage.

Performance gains from CoT in the dynamic setting. CoT consistently provides performance gains in all settings, with higher gains in settings with higher ratio of apprentice VLM usage. Most notably in Figure 5's WarmStart setting, the MMMU score gain increased from 0.007 to 0.77 as the apprentice VLM usage increased sfrom 10% to 90%.

Hierarchical vs. Dense Retrieval. Unlike in static settings, hierarchical cluster-based retrieval underperforms vs. dense retrieval, primarily due to its hyperparameters remaining fixed while the cache content evolves. We defer exploration of dynamic hyperparameter tuning to future work.

Apprentice VLM performance gains. CoT consistently achieves performance increases for the apprentice VLM no matter how much it has been used. As shown in Figure 6, when the usage of apprentice VLM is between 30% and 70%, the performance gains are between 10% and 23%.

Apprentice VLM performance gain vs. cache usage. As CoT's cache grows as its multiplexer routes queries to the master VLM, the effectiveness of CoT's in-context learning will improve over time: as seen in Figure 7), at 10%-30% apprentice VLM usage, the MMMU score increased between the first and third dataset splits, showing that the apprentice VLM was able to perform better as more entries were added to the CoT's cache.

7 Conclusion

We proposed CoT, a VLM query serving framework that achieves an effective cost-quality tradeoff for general VQA. CoT interleaves large (master) and small (apprentice) VLM usage for query serving, caching high-quality responses generated from master VLMs to significantly boost the performance of apprentice VLMs via a novel multimodal retrieval and in-context learning technique. CoT features a multiplexer to balance master VLM calls for cached example generation, and apprentice VLM calls, which uses a dual-modality dense retrieval mechanism to fetch the most relevant examples from the cache for in-context learning. We evaluate CoT on various challenging VQA benchmarks and show that CoT's techniques can increase overall VQA performance by up to 7.7% under the same budget constraint, and specifically boosts the performance of apprentice VLMs by up to 36.6%.

In the future, we plan to explore generalizing CoT's techniques to models handling other modalities (e.g., code, audio) where similar accuracy-cost trade-offs are present.

8 Limitations

First, we conduct extensive experiments in a setup featuring one master model and one apprentice model, and CoT also has potential to extend this approach to a multi-level master-apprentice framework (e.g., incorporating a 7B model, a 72B model, a 405B model, and a GPT4-o). Second, while our current focus is on image and text modalities, the framework can be extended to include additional modalities, such as video data, acoustic data, code data or other sensory inputs, given more multimodal large models are ready to use.

References

Agnar Aamodt and Enric Plaza. 1994. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Commun.*, 7(1):39–59.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc.

Anthropic. 2024. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet. Accessed 25 Oct. 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. 2023. Can pre-trained vision and language models answer visual information-seeking questions? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14948–14968, Singapore. Association for Computational Linguistics.

Zhanpeng Chen, Chengjin Xu, Yiyan Qi, and Jian Guo. 2024. Mllm is a strong reranker: Advancing multimodal retrieval-augmented generation via knowledge-enhanced reranking and noise-injected training. *Preprint*, arXiv:2407.21439.

Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. 2023. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*.

Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. 2024. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid LLM: Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*.

Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. 2024. Manipulate-anything: Automating real-world robots using vision-language models. *arXiv preprint arXiv:2406.18915*.

Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. 2024a. Physically grounded vision-language models for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024b. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Team Gemini. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Hexiang (Frank) Hu, Pat Verga, Wenhu Chen, William Weston Cohen, and Xi Chen. 2022. Murag: Multimodal retrieval-augmented generator.

Yixing Jiang, Jeremy Irvin, Ji Hun Wang, Muhammad Ahmed Chaudhry, Jonathan H. Chen, and Andrew Y. Ng. 2024. Many-shot in-context learning in multimodal foundation models. *Preprint*, arXiv:2405.09798.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Weizhe Lin, Jinghong Chen, Jingbiao Mei, Alexandru Coca, and Bill Byrne. 2024a. Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Weizhe Lin, Jingbiao Mei, Jinghong Chen, and Bill Byrne. 2024b. PreFLMR: Scaling up fine-grained late-interaction multi-modal retrievers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5294–5316, Bangkok, Thailand. Association for Computational Linguistics.

Yuqing Liu, Yu Wang, Lichao Sun, and Philip S. Yu. 2024. Rec-gpt4v: Multimodal recommendation with large vision-language models. *Preprint*, arXiv:2402.08670.

Yu A. Malkov and D. A. Yashunin. 2020. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 2019. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Thomas Mensink, Jasper Uijlings, Lluis Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, Andre Araujo, and Vittorio Ferrari. 2023. Encyclopedic VQA: Visual questions about detailed properties of fine-grained categories. In *ICCV*.

Grégoire Mialon, Roberto Dessi, Maria Lomeli, Christoforos Nalmpantis, Ramakanth Pasunuru, Roberta Raileanu, Baptiste Roziere, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Transactions on Machine Learning Research*. Survey Certification.

Thao Nguyen, Haotian Liu, Yuheng Li, Mu Cai, Utkarsh Ojha, and Yong Jae Lee. 2024. Yo'llava: Your personalized language and vision assistant. *Preprint*, arXiv:2406.09400.

OpenAI. 2024a. Gpt-4o mini: advancing cost-efficient intelligence. `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/`. Accessed 25 Oct. 2024.

OpenAI. 2024b. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`. Accessed 25 Oct. 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Edward Vendrow, Omiros Pantazis, Alexander Shepard, Gabriel Brostow, Kate E Jones, Oisin Mac Aodha, Sara Beery, and Grant Van Horn. 2024. Inquire: A natural world text-to-image retrieval benchmark. *NeurIPS*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2018. Fvqa: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2413–2427.

Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Richard James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-Tau Yih. 2023. Retrieval-augmented multimodal language modeling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39755–39769. PMLR.

Jianhao Yuan, Shuyang Sun, Daniel Omeiza, Bo Zhao, Paul Newman, Lars Kunze, and Matthew Gadd. 2024. RAG-Driver: Generalisable Driving Explanations with Retrieval-Augmented In-Context Learning in

Multi-Modal Large Language Model. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. 2025. Long-clip: Unlocking the long-text capability of clip. In *Computer Vision – ECCV 2024*, pages 310–325, Cham. Springer Nature Switzerland.

Yuanhan Zhang, Kaiyang Zhou, and Ziwei Liu. 2023. What makes good examples for visual in-context learning? In *Advances in Neural Information Processing Systems*, volume 36, pages 17773–17794. Curran Associates, Inc.

Zhixin Zhang, Yiyuan Zhang, Xiaohan Ding, and Xiangyu Yue. 2024. Vision search assistant: Empower vision-language models as multimodal search engines. *arXiv preprint arXiv:2410.21220*.

Xingcheng Zhou, Mingyu Liu, Ekim Yurtsever, Bare Luka Zagar, Walter Zimmer, Hu Cao, and Alois C. Knoll. 2024. Vision language models in autonomous driving: A survey and outlook. *IEEE Transactions on Intelligent Vehicles*, pages 1–20.

Yongshuo Zong, Ondrej Bohdal, and Timothy Hospedales. 2024. Vl-icl bench: The devil in the details of multimodal in-context learning. *Preprint*, arXiv:2403.13164.

A Example Appendix

Open-Flamingo Prompts

MMMU In-context Learning Prompt:
<image> <shot-1> <|endofchunk|> ... <image> <shot-n> <|endofchunk|> <question> <image> The options are the following: <option> There is only one option possible. The answer is

MMMU N-shot Cache Sample:
Human: <question> The options are the following: <option>. Please include your reasoning steps, then answer your choice in this format: ANSWER: LETTER CHOICE. The letter choice is strictly in alphabetical order, and there is only one option possible.
Assistant: <gpt4-o response>

MMMU Auxiliary VLM Prompt for Response:
<image> <question> The options are the following: <option> There is only one option possible. The answer is

CLEVR In-context Learning Prompt:
<image> <shot-1> <|endofchunk|> ... <image> <shot-n> <|endofchunk|> <image> How many <question> objects? The answer is

CLEVR N-shot Cache Sample:
Human: How many objects in the image have the <question>? Please include your reasoning steps, then answer your choice in this format: ANSWER: NUMBER.
Assistant: <gpt4-o response>

CLEVR Auxiliary VLM Prompt for Response:
<image> How many <question> objects? The answer is

TextOCR In-context Learning Prompt:
<image> <shot-1> <|endofchunk|> ... <image> <shot-n> <|endofchunk|> <image> What text is shown in the red box? Only answer with the largest text. The answer is

TextOCR N-shot Cache Sample:
Human: What text is shown in the red box? Only answer with the largest text. Please include your reasoning steps, then answer your choice in this format: ANSWER: TEXT.
Assistant: <gpt4-o response>

TextOCR Auxiliary VLM Prompt for Response:
<image> What text is shown in the red box? Only answer with the largest text. The answer is

Qwen Prompts

MMMU In-context Learning Prompt:
This is a chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions. If a question does not make any sense, or is not factually coherent, explain why instead of answering something not correct. If you don't know the answer to a question, please don't share false information. The assistant will have one similar in-context example provided by another powerful assistant:
<shot-1>...<shot-n>

Now you should answer the following question given the image below and you can use GPT4 Assistant's case for reference:
Human:
<question> <option> Please include your reasoning steps, then answer your choice in this format: ANSWER: <LETTER CHOICE>. The letter choice is strictly in the alphabetical order, and there is only one option possible.
Assistant(you):

MMMU N-shot Cache Sample:
Human: <question> <option> Please include your reasoning steps, then answer your choice in this format: ANSWER: <LETTER CHOICE>. The letter choice is strictly in the alphabetical order, and there is only one option possible.
Assistant: <gpt4-o response>

MMMU Auxiliary VLM Prompt for Response:
Now you should answer the following question:
Human:
<question> <option> Please include your reasoning steps, then answer your choice in this format: ANSWER: <LETTER CHOICE>. The letter choice is strictly in the alphabetical order, and there is only one option possible.
Assistant(you):

Llama Auxiliary Keyword Extractor Prompts

MMMU Prompt
You are a helpful chatbot that assists users in generating keywords from conversations. You have been given a piece of text and need to generate keywords from it. Please give me 10 keywords that are present in this question-option context and separate them with commas. Make sure you to only return the keywords and say nothing else. Make sure to exclude the words "question" and "options" as keywords I have the following multiple choice question and its options: <query>

CLEVR and TextOCR Prompt
You are a helpful chatbot that assists users in generating keywords from conversations. You have been given a piece of text and need to generate keywords from it. Please give me 10 keywords that are present in this context and separate them with commas. Make sure you to only return the keywords and say nothing else. Make sure to exclude the word "ANSWER" as keywords I have the following contexts: <query>

GPT4-o Prompts

MMMU Prompt
<question> The options are the following: <option>. Please include your reasoning steps, then answer your choice in this format: ANSWER: <LETTER CHOICE>. The letter choice is strictly in alphabetical order, and there is only one option possible.

CLEVR Prompt
How many objects in the image have the <question> Please include your reasoning steps, then answer your choice in this format: ANSWER: <NUMBER>.

TextOCR Prompt
What text is shown in the red box? Only answer with the largest text. Please include your reasoning

steps, then answer your choice in this format: ANSWER: <TEXT>.