RMIR: A Benchmark Dataset for Reasoning-Intensive Multimodal Image Retrieval

Yijiang Li[*,†]
University of California, San Diego
yijiangli@ucsd.edu

Kunal Kotian[*]
Amazon
kkotian@amazon.com

Ali Marjaninejad[*]
Amazon
alimarj@amazon.com

Meir Friedenberg[*]
Amazon
meirf@amazon.com

Kaushik Pavani[†]
pavanikaushik@gmail.com

Sunny Dasgupta
Amazon
sunnyd@amazon.com

Abstract

Current multimodal image retrieval benchmarks focus on relatively simple queries where target images are either described directly or by simple composition with an input image. When retrieval requires complex reasoning to determine the target image, the task becomes significantly more challenging, yet standardized benchmarks for this setting do not exist. To fill this gap, we introduce RMIR*, a benchmark dataset of* 1,634 *queries requiring reasoning across three categories: functional (object affordances), temporal (time-based relationships), and causal (cause-effect reasoning). Each query combines visual and textual inputs that demand robust visual understanding together with logical inference, beyond surface-level matching, to identify correct target images. In addition to the dataset itself, we present a pipeline to generate the dataset, which can be used to generate additional reasoning-intensive retrieval data at scale. Evaluation of state-of-the-art models on* RMIR *reveals significant performance gaps, with the best model achieving only* 46.53% *recall@*20 *averaged across reasoning categories. Our evaluation also shows that generative embedding models with explicit reasoning substantially outperform discriminative approaches, with reasoning-aware training proving more impactful than model scale. Our systematic analysis exposes fundamental limitations in current multimodal retrieval systems and establishes* RMIR *as a challenging testbed for developing multimodal, reasoning-capable retrieval models. Our dataset and code are available at* https://github.com/amazon-science/rmir

[*]Equal contribution
[†]Work done while at Amazon

1. Introduction

Image retrieval has traditionally focused on finding an image based on a unimodal specification, whether a similar image or a textual description. As applications have grown more sophisticated, the field has evolved toward *conditioned* or *composed* image retrieval [1, 2], where systems must find target images based on both a reference image and a textual specification of how to modify the image (see Figure 1 for examples of what each of these tasks might look like). Only recently, though, has the community begun to explore the yet harder task of *Reasoning-Intensive* Multimodal Image Retrieval [3, 4]. In this setting, retrieval systems are passed an image and a text query about the image that requires *complex, potentially multi-step* reasoning to answer, and are tasked with finding images that answer the query. As we see increased use of retrieval-augmented systems in high-complexity domains, being able to solve these reasoning-intensive retrieval tasks will become increasingly critical. As of yet, though, there do not exist standard datasets for training or evaluation of retrieval models designed for this task.

There are a number of complex desiderata for such a dataset:

- **Complexity:** Queries in the dataset must not be answerable based on surface features alone, but must instead require 1-2 steps of logical reasoning. Similarly, queries must have truly multimodal inputs, not be answerable based on just the input image or query text. At the same time, we are targeting a complexity level still solvable by the average person, without requiring expertise in a particular domain.
- **Correctness:** We must ensure the correctness of the content of the dataset, *i.e.*, any image we mark as positive for a query must correctly answer the query about the input image.

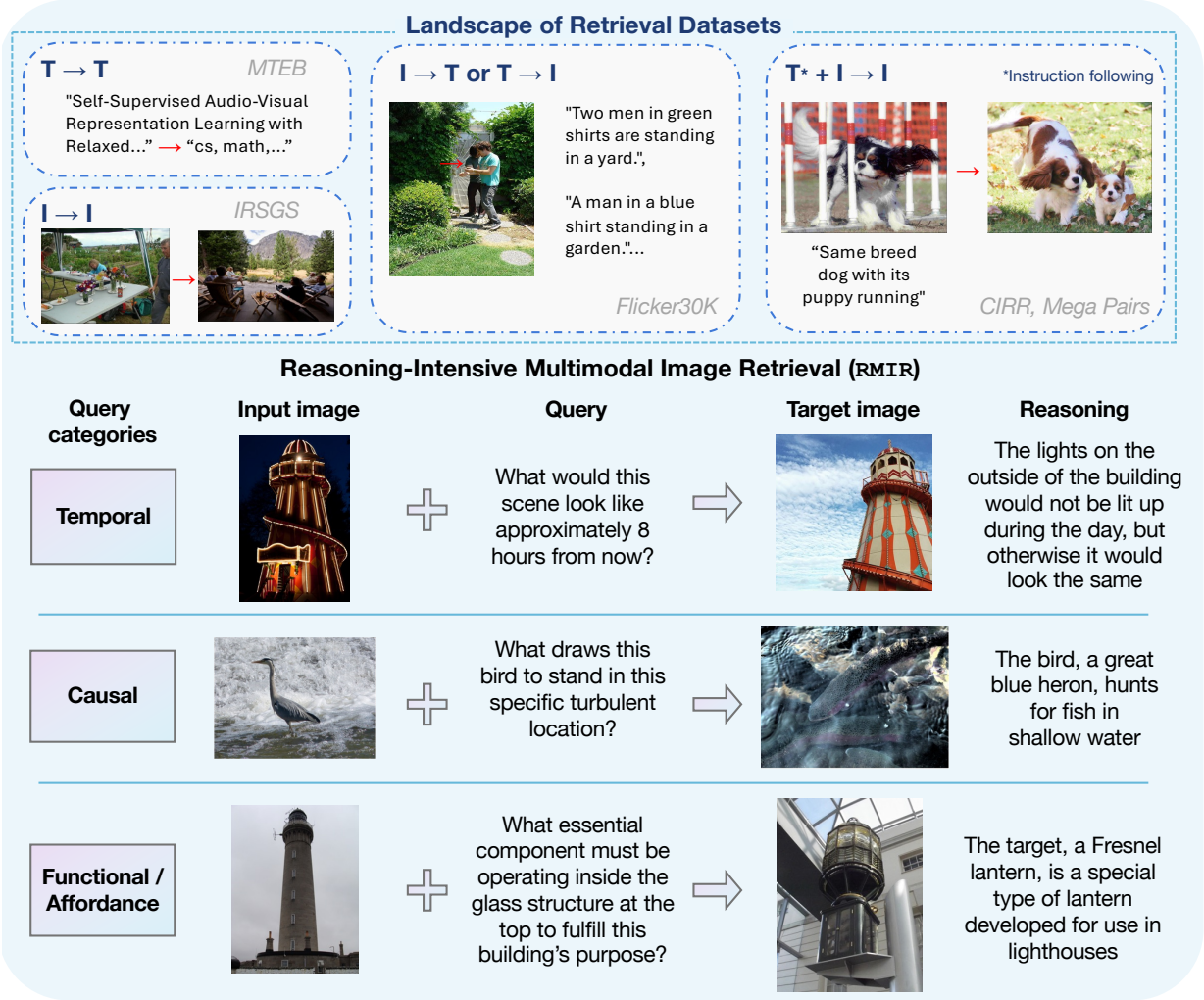

Figure 1. **The landscape of retrieval datasets.** Previous retrieval datasets have progressed over time from simple unimodal retrieval to conditioned retrieval based on image and text. In RMIR, we introduce queries that are not only multi-modal, but also require complex reasoning across the inputs. Our instances come from three reasoning paradigms: *Functional* (object utility), *Causal* (cause-effect relationships), and *Temporal* (changes over time).

- **Retrieval completeness:** On the test dataset used for model evaluation purposes, we must be confident that we have identified all true positives in the pool of candidate images, released as part of the dataset, from which the retriever is meant to select answers. This ensures that evaluated models are not incorrectly penalized for a valid answer which we had not previously identified.
- **Cost-effectiveness:** Data generation needs to be scalable, in order to be able to create data for training of future models for this task.

These desiderata may conflict, reflecting inherent trade-offs among the objectives. For instance, increasing the complexity of queries may make it more costly to evaluate the ground-truth, thus decreasing the scalability of the data-generating pipeline and increasing the likelihood of including incorrect answers. Similarly, increasing the size of the candidate pool from which to retrieve may make it easier to construct complex queries, but at the risk of introducing false negatives.

Taking these considerations into account, our paper has two main contributions: (1) we release a new benchmark for reasoning-intensive multimodal image retrieval, RMIR, as well as data that can be used for training purposes, and (2) we introduce the novel pipeline we developed for generation of this type of data at arbitrary scale. We release code for this pipeline and the curated RMIR dataset at https://github.com/amazon-science/rmir.

We conduct comprehensive evaluation on RMIR across multiple state-of-the-art (SOTA) multimodal re-

Dataset	Task Category	Modality	Size	Reasoning	Expert Curation	Scalable
Semantic Matching Benchmarks						
M-BEIR [5]	Universal Retrieval	Mixed	190K	×	×	✓
MMEB [6]	Embedding Evaluation	Mixed	36K	×	×	✓
FashionIQ [7]	Composed Image Retrieval	I+T→I	24K	×	×	×
CIRR [8]	Composed Image Retrieval	I+T→I	36K	×	×	×
InfoSeek [9]	VQA	I+T→T	1.36M	×	×	✓
Reasoning-Intensive Benchmarks						
BRIGHT [10]	Document Retrieval	T→T	1.4K	✓	✓	×
Cola [11]	Compositional Retrieval	T→I	1.2K	✓	×	×
Dyn-VQA [12]	Retrieval-Augmented (Multi-hop) VQA	I+T→I+T	1.5K	✓	✓	×
MRMR [13]	Expert-level Reasoning Retrieval	I+T→I+T	1.4K	✓	✓	×
MR^2-Bench [14]	Expert-level Reasoning Retrieval	I+T→Mixed	1.3K	✓	✓	×
RMIR (Ours)	**Reasoning Retrieval**	**I+T→I**	**1.6K**	✓	×	✓

Table 1. Overview of multimodal retrieval benchmarks. "Task Category" summarizes the nature of the tasks in each benchmark. "Modality" denotes input→output modalities using T (text) and I (image), *e.g.*, I+T→I+T for image+text-to-image+text. "Size" indicates the number of queries available for evaluation in each benchmark. "Reasoning" indicates whether queries require reasoning. "Expert Curation" indicates whether the queries were manually curated by domain experts. "Scalable" indicates whether the dataset generation method is scalable.

trieval models. Our results reveal considerable performance gaps, underscoring the challenging nature of our benchmark and the need for improved reasoning capabilities in current retrieval systems. The systematic nature of our curation approach provides a foundation for developing retrieval models with enhanced reasoning capabilities.

2. Related Work

Table 1 outlines the characteristics of several notable multimodal retrieval benchmarks. Existing multimodal retrieval benchmarks focus predominantly on surface-level alignment between images and text [15, 16], requiring retrieval of images that are semantically similar to an input text or image query. In composed image retrieval (CIR) tasks, queries are fundamentally multimodal, with textual queries specifying a modification of the input image; retrievers are then tasked with finding the modified image described by the text query and input image [1, 2, 7, 8]. Compositional retrieval benchmarks such as Cola [11] target tasks requiring retrievers to bind each attribute in a complex composite expression to the right objects in an image.

Large-scale benchmarks like M-BEIR [5] and MMEB [6] provide comprehensive evaluation by combining a wide range of smaller datasets across diverse domains, but emphasize breadth over reasoning depth. In the realm of visual question answering (VQA), massive, automatically-generated datasets like InfoSeek [9] evaluate a model's ability to ground visual entities and retrieve fine-grained factual knowledge from external sources. However, they primarily focus on knowledge lookup and entity-centric reasoning, with limited emphasis on multi-step multimodal visual reasoning.

More recently, a few retrieval datasets requiring more complex reasoning have started to be introduced. BRIGHT [10] consists of reasoning-intensive retrieval instances across a range of expert domains (*e.g.*, economics, psychology, mathematics, and coding), but in a text-only document retrieval setting. While Dyn-VQA [12] explores dynamic knowledge and temporal reasoning within an end-to-end multimodal RAG framework, its primary evaluation rests on the final generated answer.

Most relevant to this work are two other concurrent works, MRMR [13] and MR^2-Bench [14], which also introduce multi-modal retrieval benchmarks requiring complex reasoning. There are two main distinguishing factors between these two papers and ours. The first is that whereas our dataset focuses on "day-to-day" or "non-expert" reasoning (*e.g.*, "what tool would you need to achieve this task?"), the others focus on a mixture of domain-specific expert-level reasoning, technical figures, and visual puzzles. In this regard, our datasets are complementary, as it would make sense to evaluate future reasoning models on both types of domains. The second difference is that, whereas the other two papers release a benchmark, we release a pipeline for generating data, in addition to a train and test set. Here too there is a tradeoff; our pipeline enables scalable generation of data for training and evaluation, but the other two papers can ensure a high quality bar by human review of every query.

3. RMIR Dataset

We present an automated pipeline for constructing RMIR, our dataset for reasoning-intensive multimodal image retrieval. Our approach directly addresses the four desiderata from Section 1: *complexity* through multi-step reasoning requirements, *correctness* via multi-judge verification, *completeness* through systematic false negative detection, and *cost-effectiveness* via automated dataset generation (as opposed to manual annotation). We describe the pipeline architecture in Section 3.1 and resulting dataset's characteristics in Section 3.2.

3.1. Automated Dataset Generation Pipeline

Our pipeline transforms any existing dataset of images into reasoning-intensive retrieval tasks through 7 stages (Figure 2), each designed to satisfy specific desiderata while maintaining balance across competing constraints.

Prerequisite seed images dataset: The pipeline requires a seed images dataset to provide both the source images for query generation and the retrieval pool for evaluation. We first aggregate a total of 2,018,175 images from Visual Genome [17] and Open Images v7 [18] and embed them using SigLIP 2 So400m [19], an embedding model pretrained to capture high-level visual semantics in a combined image–text space. We then cluster the images into 50,000 clusters using Faiss' k-means implementation [20]. Finally, we construct our seed dataset of 500,000 images by randomly sampling 10 images from each cluster. This subsampling strategy balances two competing objectives: reducing image similarity within the dataset to ensure distinct retrieval sets for different queries, while maintaining sufficient diversity to provide high-quality matches for each target image description (further discussed in Section 3.1.7). We use Visual Genome and Open Images v7 since they offer diverse images spanning everyday objects, scenes, and activities ideal for generating queries requiring common-sense reasoning accessible to non-experts.

3.1.1. Query Generation

Our dataset generation pipeline starts with a Query Generation stage, which produces reasoning-intensive query candidates from seed images. For each input image sampled from the seed dataset, we prompt a SOTA MLLM (Claude Sonnet 4.5, temp=0.25) to generate two outputs per reasoning category: (1) a reasoning-intensive text query conditioned on the input image, and (2) a natural language description of a target image that represents the answer to the query. The prompt instructions impose three constraints on this query generation task:

1. *Multimodal dependency*: Target images must depend on both input image and query text, not query text alone.
2. *Reasoning complexity*: Queries must require 1-2 steps of logical inference, and not be solvable by object detection alone.
3. *Target specificity*: Queries must uniquely determine a single target concept, and not admit multiple valid answers.

For each input image, we instruct the model to generate three queries, each targeting a distinct reasoning category:

1. *Temporal reasoning*: Ordering events in time and determining their sequential or simultaneous relationships.
2. *Causal reasoning*: Identifying cause-effect relationships that explain why or how events occur.
3. *Functional reasoning*: Determining what tools or objects are required to accomplish a specific task.

To steer query generation, we provide the model with definitions and few-shot examples for each of the three reasoning categories (full prompts in Appendix A.1). This automated approach ensures cost-effective (no human involvement) scaling of query generation.

3.1.2. Query Filtration

Some of the candidate queries produced in the Query Generation stage do not fully satisfy our design constraints despite explicit prompt instructions. To address this, we introduce a Query Filtration stage that validates each generated (input image, query) pair before proceeding to the computationally expensive Candidate Image Retrieval and Query Triplet Judging stages. Specifically, we prompt an MLLM (Claude Sonnet 4.5, temp=0) to verify whether each candidate query satisfies the three constraints specified during Query Generation, namely, multimodal dependency, reasoning complexity, and target specificity.

We retain only those query candidates where the MLLM confirms all three constraints are met, filtering out $\approx$ 40% of generated queries. This validation step serves dual purposes: it maintains dataset quality by catching constraint violations that slip through generation, and it reduces downstream costs by eliminating invalid queries before the expensive Candidate Image Retrieval and Query Triplet Judging stages. The full prompt for query filtration is provided in Appendix A.2.

3.1.3. Candidate Image Retrieval

Having generated and filtered sets of {input image, query, target image description}, we now retrieve actual candidate images from our seed dataset that match each target description. We employ a dual-index retrieval strategy to maximize the likelihood of finding all suitable target images. First, we create two complementary indices over our 500,000-image seed dataset: (1) a vision-based index created by embedding images directly using SigLIP 2 So400m

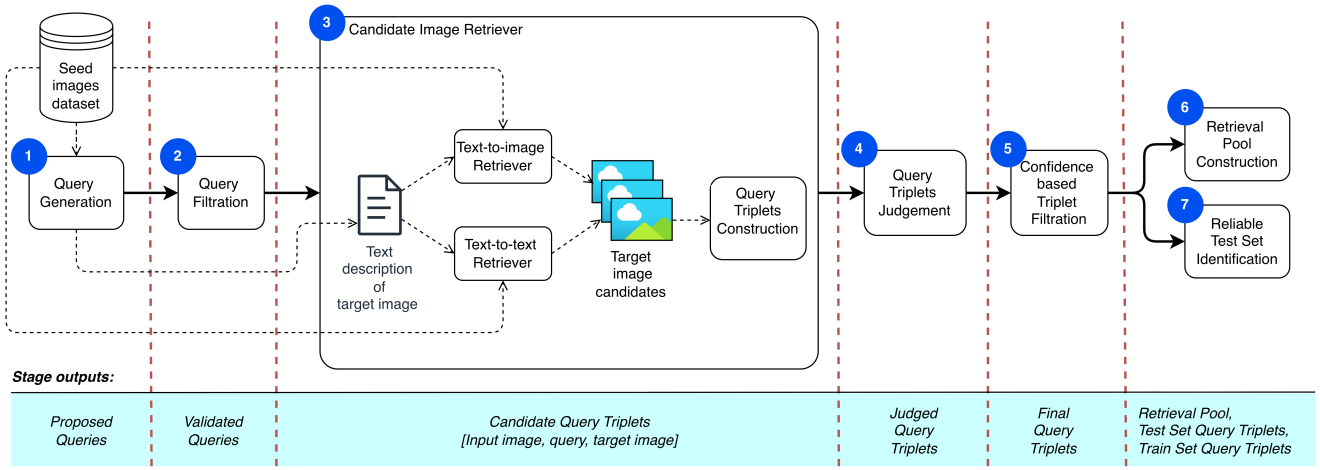

Figure 2. The RMIR multi-stage dataset generation pipeline.

[19], a vision-language encoder, and (2) a text-based index created by embedding the available image descriptions using Qwen3-Embedding-0.6B [21]. This dual approach allows us to leverage semantic similarity to both the candidate images themselves and any available textual labels when matching target descriptions to candidate images.

For each target image description, we embed it into both embedding spaces and retrieve the top K_{img} candidates (K_{img}=16 in practice) from the image-based index and top K_{desc} candidates (K_{desc}=4 in practice) from the text-based index. The exact numbers of candidates to retrieve from each index are hyperparameters we did not tune, but because not all images have text captions we draw a larger proportion of images from the more densely populated image-based index. This yields up to 20 candidate target images per query, each forming a complete triplet with the original input image and query.

3.1.4. Query Triplet Judging

After retrieving candidate target images, we now have complete *query triplets* – an input image, a query about that image, and a target image that correctly answers the associated query about the input image. A set of query triplets forms one of the two main components of a retrieval dataset, together with the *retrieval pool* which we will discuss in Section 3.1.6.

Having a set of query triplets in hand, we validate each query triplet using a panel of 5 MLLM judges (one Claude Sonnet 4.5 instance and four Claude Haiku 4.5 instances, temp=0.85) that evaluate whether the retrieved target image represents a logically consistent answer to the query given the input image. Judges are instructed to accept triplets where the reasoning connecting the input to the target is clear enough that a typical adult could be expected to make the connection within 30 seconds and with 1-2 steps of inference. Judges can accept (True), reject (False), or abstain if critical information is missing or if the content is politically sensitive or sexually explicit. Following the Self-Consistency method [22], we execute the judgment prompt stochastically at temperature 0.85, using the proportion of judges agreeing with the majority decision as a proxy for confidence (e.g. if 4 judges vote True and 1 votes False then the confidence is 0.8). The full prompt, including detailed evaluation criteria and examples, is provided in Appendix A.3. Appendix B.1 details small-scale human validation of judgment quality.

3.1.5. Confidence-Based Query Triplet Filtration

To ensure query triplet correctness, we remove all query triplets where the panel of judges generated a confidence below 1.0, as these represent cases where we cannot be certain whether the target image is a correct answer to the query. However, simply discarding low-confidence triplets is insufficient to maintain retrieval completeness. The challenge arises because target images across all query triplets exist in a shared retrieval pool from which evaluated models must fetch correct images for input queries. If we merely remove a low-confidence query triplet while its target image remains in the pool (due to appearing in other retained triplets), a retriever model may still retrieve that image as a candidate for the original query. Since we lack certainty about whether this triplet is accurate, we could incorrectly penalize the model for retrieving what might actually be a valid target. To address this, for each filtered query triplet, we also remove its target image from the entire seed dataset. This mitigates one source of retrieval incompleteness; we discuss additional measures in Section 3.1.7.

3.1.6. Creation of a Retrieval Pool

The other core component of a retrieval dataset, in addition to the set of query triplets, is the *retrieval pool* from

which the retriever model is expected to select answers to each query. To construct the retrieval pool for our dataset, we take the union of all candidate target images that survived high-confidence filtering across all query triplets. For our current dataset, this approach led to selection of a final retrieval pool of 35,803 images out of the original seed dataset of 500,000.

It is worth noting that the other natural option would have been to use the full seed dataset as the retrieval pool, minus those images we removed in query triplet filtration (Section 3.1.5) due to uncertainty. While this would increase the level of difficulty for retriever models, it would also increase the likelihood of having false negatives in the dataset, which is why we chose to use the union of identified target images for our retrieval pool.

3.1.7. Identification of a Reliable Subset for Evaluation

The pipeline as described above leads to generation of high quality retrieval data requiring complex reasoning about both the query and the input image. But at the same time, there are still a few ways that the dataset can, on occasion, end up containing false negatives. Because of this, we introduce one more filter for the test set that is meant to help alleviate one of these potential scenarios.

Because we only pass K images to the panel of MLLM judges for evaluation, one way false negatives can be introduced is if the $K+1^{th}$ element is a valid answer. The fact that our candidate pool only contains images needed for some query and the fact that our queries are relatively complex both decrease the likelihood of this happening, but it is still a possibility.

Intuitively, though, if we see that the first K elements are all positive matches there is a higher likelihood of the $K+1^{th}$ element also being a match than in a case where only the first element was a match and everything else was negative. More generally, we think that a) the more negatives we see among the candidates we evaluated, and b) the more the positives that exist are concentrated in just the first few elements, the higher the likelihood that we haven't missed additional positives by stopping too early. We formalize this notion with a "Test Set Reliability" (TSR) score, defined as follows:

Let L be a list of retrieved images for a query, with L_1 as the highest ranked element by the retriever and L_K the lowest ranked retrieved element. We define the "negativeness" of an element L_i as the fraction of judges that determined it to be a negative instance, or 0 if less than half did. $\texttt{mean\_negativeness}(L[i:j])$ is then taken to be the average negativeness over elements $L_i, \ldots, L_j$. We then define

$$\text{TSR}(L) = \frac{1}{K}\sum_{i=1}^{K} \texttt{mean\_negativeness}(L[i:K]).$$

This in some sense captures a "tail-negativeness" notion – the more negatives we see towards the 'tail' of a retrieved list, the more confident we are that there aren't other missed negatives, and therefore the higher the TSR score will be.

In practice, we threshold our examples using the K_{img} images from the vision-based index, padding out with artificial negative instances if less than K_{img} were retrieved. We include all queries with TSR greater than some threshold in our test set, and leave the remaining data as suitable for training.

This system for generating data is of course not without flaws – it is, for example, still possible for there to be false negatives in the data if there are alternative valid answers to the query that the MLLM did not identify. That said, there is a tension between fully ensuring data quality and being able to generate data at scale, and we believe we have struck a good balance; at least empirically, most of our data seems to be relatively high quality compared to standard retrieval datasets.

Though the effectiveness of TSR warrants further study in its own right, Appendix B.2 provides some evidence for it based on a lightweight human validation.

3.2. Dataset Composition and Statistics

The RMIR dataset generated using the pipeline in Section 3.1 contains 1,634 curated reasoning queries for evaluation (test set), and an additional 6,687 queries that can be used for retriever training. The test set contains 546, 656, and 432 queries from the functional, temporal, and causal reasoning categories, respectively, with the average text query length for these categories being 13.5, 13.1, 13.0 words, respectively. Our queries' texts have a median Flesch-Kincaid score of 8 (middle-school reading level), indicating the query language itself is not complex. RMIR also includes a shared pool of candidate images for retrieval comprising 35,803 images, used for all queries.

4. Experiments

4.1. Setup

Models tested. We evaluated 11 multimodal models against the RMIR dataset, measuring their retrieval performance across multimodal queries involving functional, temporal, and causal reasoning. These include (i) open source MLLMs with strong visual reasoning capabilities like Qwen2.5-VL [23, 24], Qwen3-VL [25], and Phi-4-Multimodal-Instruct [26], (ii) VLM2Vec [6] and VLM2Vec-V2 [27] universal multimodal embedding models, which are derived from MLLM backbones that are trained over retrieval tasks in the MMEB [6] and and MMEB-V2 [27] datasets respectively via instruction-tuned contrastive training, and (iii) UME-R1 [3] generative multimodal embedding models trained via supervised fine-tuning and

Retriever Models	Functional		Temporal		Causal		Average	
	R@20	R@50	R@20	R@50	R@20	R@50	R@20	R@50
Qwen2.5-VL-3B-Instruct	34.40	46.58	28.35	40.45	31.69	43.79	31.71	43.84
Qwen2.5-VL-7B-Instruct	36.42	48.34	36.76	49.86	37.38	48.70	36.89	48.88
Qwen3-VL-2B-Instruct	8.98	13.49	4.97	8.05	6.13	9.51	6.77	10.46
Qwen3-VL-4B-Instruct	35.94	45.00	30.48	43.28	32.21	42.70	33.00	43.62
Phi-4-Multimodal-Instruct-5.6B	25.46	37.12	25.80	35.42	27.41	37.58	26.33	36.86
VLM2Vec-Qwen2-VL-2B	26.46	38.43	24.91	35.05	30.19	40.36	27.55	38.31
VLM2Vec-Qwen2-VL-7B	42.34	55.91	32.59	46.51	39.64	52.42	38.68	52.03
VLM2Vec-Full (Phi-3.5-V-4.2B)	32.51	46.79	25.82	39.98	31.24	44.18	30.23	43.94
VLM2Vec-V2 (Qwen2-VL-2B)	31.53	43.63	28.33	39.24	32.35	41.65	31.01	41.67
UME-R1-2B (Qwen2-VL-2B)	39.08	55.08	35.53	49.33	38.95	52.90	38.09	52.68
UME-R1-7B (Qwen2-VL-7B)	**51.34**	**64.64**	**40.58**	**56.99**	**46.46**	**59.32**	**46.53**	**60.48**

Table 2. **Performance of various multimodal retrievers on RMIR across different reasoning categories.** Results show Recall@20 and Recall@50 for Functional, Temporal, and Causal reasoning tasks. UME-R1 models are evaluated with queries embedded in generative mode and candidates embedded in discriminative mode. Backbone models are denoted in parentheses when not apparent from the retriever's name.

reinforcement learning to produce reasoning-aware embeddings for image–text inputs, improving retrieval performance over standard discriminative encoders.

Evaluation metrics. For each query, we measure the standard information retrieval metric **recall@K** (**R@K**), defined as the proportion of the query's relevant items caught among the top K retrieved items. We then average the R@K values across all queries in the dataset to compute the dataset-level R@K. Since the queries in RMIR can have up to 20 relevant items, we set K to 20 and 50, reported hereafter as R@20 and R@50.

4.2. Results and Analysis

Table 2 presents the performance of all evaluated models on RMIR. The results demonstrate that reasoning-intensive multimodal retrieval remains a challenging task for all models. The best-performing model, UME-R1-7B, achieves 46.53% R@20 and 60.48% R@50 averaged across all reasoning categories, indicating substantial room for further recall improvement.

Generative embeddings unlock substantial gains through reasoning. The UME-R1 models [3], which leverage generative embeddings by unifying embedding tasks within a generative paradigm, substantially outperform all other approaches. Unlike conventional discriminative embedding models that directly encode inputs and extract representations from the last token's hidden state, UME-R1 first generates intermediate reasoning trajectories and summaries before producing embeddings. This reasoning-driven generation paradigm, enabled through a two-stage training strategy combining supervised fine-tuning with chain-of-thought annotations and reinforcement learning with verifiable rewards, allows the model to leverage the powerful generative reasoning capabilities of MLLMs. UME-R1-7B achieves the highest performance across all reasoning categories, with particularly strong results on functional reasoning (51.34% R@20, 64.64% R@50). Even the smaller UME-R1-2B model (38.09% R@20, 52.68% R@50) outperforms larger MLLMs like Qwen2.5-VL-7B (36.89% R@20, 48.88% R@50), demonstrating that specialized training for reasoning-aware retrieval is more effective than simply scaling model size.

MLLMs show moderate reasoning capabilities. Among the multimodal large language models, Qwen2.5-VL-7B achieves the best performance (36.89% R@20, 48.88% R@50), outperforming its smaller 3B variant by approximately 5 percentage points across both metrics. However, the gap between MLLMs and specialized retrieval models like UME-R1 suggests that strong visual reasoning capabilities alone are insufficient for reasoning-intensive retrieval without explicit training on retrieval objectives and reasoning-aware embedding generation.

Discriminative embedding models struggle without reasoning. The VLM2Vec family of models, despite being trained on diverse retrieval tasks through the MMEB dataset, shows limited effectiveness on RMIR. VLM2Vec-Full achieves only 30.23% R@20 and 43.94% R@50, while VLM2Vec-V2 performs similarly at 31.01% R@20 and 41.67% R@50. These models remain inherently discriminative, directly encoding multimodal inputs without gen-

erating intermediate reasoning steps. This limitation prevents them from benefiting from the reasoning-driven generation paradigm that proves crucial for RMIR. The substantial performance gap between VLM2Vec models and UME-R1 (10.54 and 7.85 percentage points at R@20 for the 2B and 7B models respectively) demonstrates that training on standard retrieval tasks does not adequately prepare models for reasoning-intensive scenarios, highlighting the importance of incorporating explicit reasoning capabilities into the embedding generation process.

Functional reasoning is most tractable. Across all models, functional reasoning queries (which require understanding object affordances and utility) yield the highest performance. This pattern holds consistently across model families, suggesting that reasoning about object functionality may be more directly supported by visual features learned during pretraining.

Temporal reasoning poses unique challenges. Temporal reasoning queries, which require understanding time-based relationships and sequential changes, prove most difficult for all models. Even UME-R1-7B achieves only 40.58% R@20 on temporal queries, 10.76 percentage points lower than its functional reasoning performance. These results indicate that current multimodal models have limited capacity to reason about temporal dynamics from static images.

Model scale matters but is not sufficient. Comparing models within the same family reveals consistent improvements with scale: Qwen2.5-VL-7B outperforms Qwen2.5-VL-3B by 5.18 percentage points at R@20, and UME-R1-7B outperforms UME-R1-2B by 8.44 percentage points. However, the best model achieves only 46.53% R@20, indicating substantial room for improvement on reasoning-intensive retrieval tasks.

5. Future Work

In this section, we identify several opportunities for future work to build upon the RMIR dataset and query generation pipeline.

Training data utility. Although we release 6,687 training queries alongside the test set, we do not conduct experiments training retrieval models on this data. Demonstrating that models fine-tuned on RMIR training data see improved performance on reasoning-intensive retrieval tasks would validate the pipeline's utility beyond evaluation.

Multimodal dependency verification. Our pipeline enforces multimodal dependency through prompt-based steering during query generation (Section 3.1.1) and filtration (Section 3.1.2). Empirically quantifying 'shortcut rates', *i.e.*, the fraction of queries solvable using only the query text or only the query image, via unimodal retrieval baselines, would provide additional confidence in the multimodal nature of the queries and is a natural direction for future work.

Model diversity in the pipeline. Our pipeline relies on a specific set of models at two stages. For candidate image retrieval (Section 3.1.3), we use two specific embedding models (SigLIP 2 and Qwen3-Embedding), which may introduce a bias toward images these models find similar while missing valid candidates they do not surface. Using additional embedding models to retrieve more candidates per query could mitigate this bias, at the cost of increased LLM judging expense. For triplet judging, our MLLM panel uses two models from a single family (Claude Sonnet 4.5 and Claude Haiku 4.5), which may share systematic blind spots. Ensembling judges from diverse model families could further improve judgment accuracy and reduce systematic biases.

Image pool diversity. The current seed dataset draws from Visual Genome and Open Images v7, which, while diverse in everyday scenes and objects, may not fully represent the breadth of visual content encountered in practice. Since our pipeline is dataset-agnostic, scaling to larger and more diverse image sources (*e.g.*, internet-scale datasets) is a natural extension that could yield substantially more training data and broader coverage of reasoning scenarios.

6. Conclusion

We introduce RMIR, a benchmark dataset for reasoning-intensive multimodal image retrieval comprising 1,634 queries across three reasoning categories: functional, temporal, and causal. Unlike existing retrieval benchmarks that focus on semantic matching, RMIR requires models to perform 1-2 steps of logical inference to identify relevant images, addressing a critical gap in evaluating retrieval systems.

Our automated dataset generation pipeline balances the competing demands of complexity, correctness, completeness, and cost-effectiveness through multi-stage filtering and quality control. By leveraging LLM-generated queries and multi-judge validation with MLLMs, we curate high-quality query triplets while maintaining scalability for future dataset expansion.

Comprehensive evaluation of state-of-the-art models reveals that reasoning-intensive retrieval remains challenging, with the best-performing model achieving only 46.53% R@20. Our analysis demonstrates that generative embedding approaches with explicit reasoning capabilities substantially outperform discriminative models, while temporal reasoning poses unique challenges across all model families. These findings establish RMIR as a valuable testbed for developing and evaluating reasoning-capable multimodal retrieval systems.

References

[1] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Image retrieval on real-life images with pre-trained vision-and-language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2125–2134, 2021. 1, 3

[2] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li-Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-an empirical odyssey. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6439–6448, 2019. 1, 3

[3] Zhibin Lan, Liqiang Niu, Fandong Meng, Jie Zhou, and Jinsong Su. Ume-r1: Exploring reasoning-driven generative multimodal embeddings. In *The Fourteenth International Conference on Learning Representations*, 2026. 1, 6, 7

[4] Xuanming Cui, Jianpeng Cheng, Hong-You Chen, Satya Narayan Shukla, Abhijeet Awasthi, Xichen Pan, Chaitanya Ahuja, Shlok Mishra, Taipeng Tian, Qi Guo, Ser-Nam Lim, Aashu Singh, and Xiangjun Fan. Think then embed: Generative context improves multimodal embedding. In *The Fourteenth International Conference on Learning Representations*, 2026. 1

[5] Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. Uniir: Training and benchmarking universal multimodal information retrievers. *arXiv preprint arXiv:2311.17136*, 2023. 3

[6] Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. In *The Thirteenth International Conference on Learning Representations*, 2025. 3, 6

[7] Hui Wu, Yupeng Gao, Xiaoxiao Guo, Ziad Al-Halah, Steven Rennie, Kristen Grauman, and Rogerio Feris. The fashioniq dataset: Retrieving images by combining side information and relative natural language feedback. In *CVPR*, pages 11814–11823, 2021. 3

[8] Zheyuan Liu, Cristian Rodriguez-Opazo, Damien Teney, and Stephen Gould. Cirr: A dataset and benchmark for cross-modal image retrieval. In *ICCV*, pages 2050–2060, 2021. 3

[9] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14948–14968, Singapore, December 2023. Association for Computational Linguistics. 3

[10] Hongjin SU, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Liu Haisu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[11] Arijit Ray, Filip Radenovic, Abhimanyu Dubey, Bryan A Plummer, Ranjay Krishna, and Kate Saenko. Cola: A benchmark for compositional text-to-image retrieval. In *Adv. Neural Inform. Process. Syst.*, 2023. 3

[12] Yangning Li, Yinghui Li, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinran Zheng, Hui Wang, Hai-Tao Zheng, Fei Huang, Jingren Zhou, and Philip Yu. Benchmarking multimodal retrieval augmented generation with dynamic VQA dataset and self-adaptive planning agent. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[13] Siyue Zhang, Yuan Gao, Xiao Zhou, Yilun Zhao, Tingyu Song, Arman Cohan, Anh Tuan Luu, and Chen Zhao. MRMR: A realistic and expert-level multidisciplinary benchmark for reasoning-intensive multimodal retrieval. In *The Fourteenth International Conference on Learning Representations*, 2026. 3

[14] Junjie Zhou, Ze Liu, Lei Xiong, Jin-Ge Yao, Yueze Wang, Shitao Xiao, Fenfen Lin, Miguel Hu Chen, Zhicheng Dou, Siqi Bao, et al. MR^2-bench: Going beyond matching to reasoning in multimodal retrieval. *arXiv preprint arXiv:2509.26378*, 2025. 3

[15] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 3

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 3

[17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73, 2017. 4

[18] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017. 4

[19] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier Hénaff, Jeremiah Harmsen, Andreas Steiner, and Xiaohua Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features, 2025. 4, 5

[20] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024. 4

[21] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025. 5

[22] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. 5

[23] 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. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6

[24] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 6

[25] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. 6

[26] Microsoft: Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuohang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zabir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras, 2025. 6

[27] Rui Meng, Ziyan Jiang, Ye Liu, Mingyi Su, Xinyi Yang, Yuepeng Fu, Can Qin, Zeyuan Chen, Ran Xu, Caiming Xiong, Yingbo Zhou, Wenhu Chen, and Semih Yavuz. Vlm2vec-v2: Advancing multimodal embedding for videos, images, and visual documents. *ArXiv*, abs/2507.04590, 2025. 6

Appendices for RMIR: A Benchmark Dataset for Reasoning-Intensive Multimodal Image Retrieval

A. Prompts

A.1. Query Generation Prompt

```
<purpose>
This task generates high quality data
for a new 'hard' multimodal retrieval
benchmarking dataset which will be
used to evaluate the ability of SOTA
multimodal retrievers to retrieve the
correct target images from a data pool
for a given input image and a query
text.  The generated query triplets
will help drive research on improving
multimodal retrievers' performance when
the link between a query's visual scene
and its textual intent is implicit and
requires intermediate reasoning.
</purpose>

<task_overview>
Using an input image that will be
provided to you, generate triplets
(input_image, query, target_desc) where
the query connects the input image
to the target image through logical
reasoning.  Each triplet should require
1-2 logical steps to solve and be
challenging enough to test multimodal
retrievers' reasoning abilities while
remaining explainable.
</task_overview>

<content_policy>
If the input image depicts content that
is politically sensitive (protests,
police actions, riots, conflicts,
political figures, etc.)  or sexually
explicit, immediately decline the task
by responding only with:  "ABSTAIN DUE
TO SENSITIVE INPUT IMAGE"
</content_policy>

<input_dependence_constraint>
The target image MUST depend on BOTH
the input image AND the query, not the
query alone.
✓ GOOD: Requires interpreting the input
image to understand what is shown,
then answering the query based on that
understanding.
× BAD: Answer determined by query
alone; input image is not needed.
The query should require understanding
WHAT is in the input image to determine
the target image.

Test:
- "Can this query be answered without
the input?" → If yes, revise the
query.
</input_dependence_constraint>

<reasoning_complexity_constraint>
The reasoning that connects the input
image + query to the target image:
- Should be challenging enough to need
1-2 logical inference steps beyond
surface-level observation, i.e., not
solvable by simple object detection
alone
- Should test understanding of
relationships, implications, or context
- Should be explainable in 1-2
sentences
</reasoning_complexity_constraint>

<specificity_constraint>
Query triplets should be generated
such that both conditions below are
met simultaneously:
- The query should be specific enough
that given the input image, there is
only 1 logically correct answer.
- The target image description should
```

be visually distinctive enough that < 5 similar-looking images would exist in a large public datasets like Open Images & Visual Genome.
- The target image description should not have irrelevant details added to superficially pass the constraint above. E.g., If the logical answer to a query is an image of "a person holding an umbrella", don't add superficial details like "red umbrella".

Tests:
- "Are there multiple equally valid target images for this input image+query?" → If yes, add meaningful specificity to the query.
- "Is the target description so generic it would match > 5 images in datasets like Open Images & Visual Genome?" → If yes, add meaningful details to the target's description.
- "Is the target description so specific it's unlikely to exist in standard datasets?" → If yes, simplify the target's description.
</specificity_constraint>

<image_requirements>
Both input and target images are sourced from these standard open source vision datasets: Open Images, Visual Genome. Hence, target image descriptions should:
- Use simple scenes, common objects, everyday photographs, realistic items, or other image types that are found in these datasets.
- Avoid abstract art, rare/unusual subjects, images unlikely to be in standard datasets.
</image_requirements>

<reasoning_categories>
Below are the reasoning categories of interest:

FUNC (FUNCTIONAL/AFFORDANCE)

Understanding what tools or objects are needed to accomplish a specific task or function.

<Example_FUNC_1>
<input_desc>A screw with Phillips head pattern embedded in wood</input_desc>
<query>What tool would fit the pattern shown to remove this?</query>
<target_desc>A Phillips head screwdriver with metal shaft</target_desc>
<reasoning>The cross-shaped indentation in the screw head requires a matching Phillips head tool to engage and turn it.</reasoning>
</Example_FUNC_1>

<Example_FUNC_2>
<input_desc>Image of a car tire with visible nail puncture causing deflation</input_desc>
<query>What tool would seal this type of damage?</query>
<target_desc>A tire repair kit with rubber plugs and insertion tool</target_desc>
<reasoning>A puncture in a tire requires a repair kit with rubber plugs that can be inserted into the hole to create an airtight seal.</reasoning>
</Example_FUNC_2>

<Example_FUNC_3>
<input_desc>Image of raw dough rolled flat on a floured surface</input_desc>
<query>What tool would create the circular shapes needed before baking these?</query>
<target_desc>A round cookie cutter</target_desc>
<reasoning>Flat rolled dough requires a cutting tool to create individual shaped portions for baking.</reasoning>
</Example_FUNC_3>

CAUS (CAUSAL)

Understanding why or how something happens (cause → effect).

<Example_CAUS_1>
<input_desc>Magnifying glass focusing sunlight on paper</input_desc>
<query>What happens at the focal point?</query>

<target_desc>Burnt hole with charred edges in a paper sheet</target_desc>
<reasoning>The magnifying glass converges sunlight onto a tiny hot spot on the paper, making it burn.</reasoning>
</Example_CAUS_1>

<Example_CAUS_2>
<input_desc>Wet laundry hanging on outdoor clothesline in sunlight</input_desc>
<query>What state will these items be in after a sunny afternoon?</query>
<target_desc>Some dry clothes on clothesline</target_desc>
<reasoning>Sunlight and air circulation evaporate the water from wet fabric, leaving them dry.</reasoning>
</Example_CAUS_2>

<Example_CAUS_3>
<input_desc>Cat stepping on computer keyboard</input_desc>
<query>What appears on the monitor screen from this pressure?</query>
<target_desc>Random text characters on a computer screen</target_desc>
<reasoning>The cat's weight depresses multiple keys, inputting unintended random characters into the computer.</reasoning>
</Example_CAUS_3>

TEMP (TEMPORAL)

Understanding and ordering events in time, determining whether one event is before, after, or simultaneous with another.

<Example_TEMP_1>
<input_desc>Tadpole swimming in pond water</input_desc>
<query>What does this creature look like after completing its development?</query>
<target_desc>An adult frog</target_desc>
<reasoning>Tadpoles undergo metamorphosis over weeks, developing legs and losing their tail to become frogs.</reasoning>
</Example_TEMP_1>

<Example_TEMP_2>
<input_desc>Athletes running during a track and field competition</input_desc>
<query>What moment comes immediately before the start of this activity?</query>
<target_desc>Athletes positioned in starting blocks on a running track.</target_desc>
<reasoning>Right before the race begins, athletes are positioned at the starting line in their ready positions, waiting for the starting signal - stationary and prepared but not yet in motion.</reasoning>
</Example_TEMP_2>

<Example_TEMP_3> <input_desc>Tree with pink cherry blossoms in spring</input_desc>
<query>What do these flowering branches produce by late summer?</query>
<target_desc>Red cherries hanging from a branch of a cherry tree</target_desc>
<reasoning>Cherry blossoms develop into fruit over several months after pollination.</reasoning>
</Example_TEMP_3>

</reasoning_categories>

<query_language_style_constraint>
The query itself should be worded in a way that a typical native English speaker can easily understand it. Avoid awkward phrasing.
</query_language_style_constraint>

<Examples>
Below are examples demonstrating incorrect and correct triplet construction:
<constraints_illustration_1>
input image:
###IMAGE OF KETTLE ON STOVE###

× BAD EXAMPLE - Input image is redundant, violating <input_dependence_constraint>:

<query>What happens when the water in the kettle reaches 100°C?</query>
<target_desc>A kettle with steam whistling vigorously from its spout</target_desc>
- Problem: The answer can be derived from the query by itself, and thus the input image is redundant

✓ GOOD EXAMPLE - Input-image-dependent reasoning:
<query>What will happen in the next few minutes?</query>
<target_desc>A kettle with steam whistling vigorously from its spout</target_desc>
- Why it works: Requires identifying the kettle and flame in the input image, then predicting the outcome of continued heating.

The GOOD EXAMPLE above also honors the other two constraints:
<reasoning_complexity_constraint> → Reasoning is needed to understand: (1) the kettle's functionality (contains water for heating), (2) the flame's effect (continuously applies heat to the kettle), and (3) the cause-and-effect relationship between sustained heating and water temperature to deduce that the water will reach boiling point and produce steam that whistles from the spout.

<specificity_constraint> → By specifying "in the next few minutes" the query establishes a temporal constraint that, when combined with the input image of an actively heated kettle, narrows the valid outcomes to the immediate consequence of continued heating (steam whistling). Without this timeframe specification, the query would allow for numerous equally valid answers such as: the kettle eventually cooling down, someone removing it from the stove, the water evaporating completely, or the stove being turned off. The temporal detail limits the valid targets while still requiring reasoning about the heating process.
</constraints_illustration_1>

<constraints_illustration_2>
input image:
###IMAGE OF GREEN LIZARD###
× BAD EXAMPLE - Query is not complex enough, violating <reasoning_complexity_constraint>:
<query>What kind of apple would best match this animals skin color?</query>
<target_desc>Granny Smith apple (green)</target_desc>
- Problem: No need to do complex reasoning. Target and input descriptions even share the same attribute (green color) which makes it a very easy answer.

✓ GOOD EXAMPLE - Reasonable complexity:
<query>What technology would a person use that mirrors this animal's defense mechanism against predators?</query>
<target_desc>Camouflage clothing</target_desc>
- Why it works: Requires understanding the analogy between the camouflage clothing and chameleon's ability to match its skin color to its environment for the same purpose.

The GOOD EXAMPLE above also honors the other two constraints:
<specificity_constraint> → Inclusion of specific terms such as "a person" in the query dramatically reduces the space of valid targets. For example, if the query did not say "a person", the space of valid targets would also include camouflaged military buildings or vehicles. Crucially, note how this extra detail of "a person" is carefully selected to ensure that even after its addition, the query still retains its reasoning complexity.
<input_dependence_constraint> → It is impossible to know what defense mechanism the query refers to without identifying the animal in the input image, thus enforcing input-image-dependence.
</constraints_illustration_2>

<constraints_illustration_3>
input image:

###IMAGE OF PIZZA IN OVEN###
× BAD EXAMPLE – Query is too broad leading to multiple equally valid answers:
<query>What would happen next?</query>
<target_desc>Cooked pizza</target_desc>
– Problem: The query is so broad that for the input image, it can lead to several distinct equally valid answers beyond cooked pizza, such as: people eating pizza, a chef preparing the next pizza order, a host opening a wine bottle, burnt pizza, etc.
✓ GOOD EXAMPLE – Provides key specifics that limits the target class yet not too revealing and therefore still needs complex reasoning:
<query>What would happen if you forgot about this?</query>
<target_desc>Burnt pizza</target_desc>
– Why it works: The query includes an action ("forgetting") that, when combined with the input image, leads to the specific outcome of the pizza getting burnt. Crucially, note how the specific query detail of "forgetting about it" is carefully selected to ensure that even after its addition, the query still retains its reasoning complexity.

The GOOD EXAMPLE above also honors the other two constraints:
<reasoning_complexity_constraint> → Reasoning is needed to understand:
(1) the oven's functionality (produces heat for cooking),
(2) the pizza's requirements (needs specific cooking time), and
(3) the cause-and-effect relationship between these factors to deduce that a longer-than-normal stay in the oven will result in a burnt pizza.

<input_dependence_constraint> → The query makes no direct reference to the item/scene in the input image, without which it is impossible to find the query's target image, thus enforcing input-image-dependence.
</constraints_illustration_3>

</Examples>

<Examples_of_bad_query_triplets>
Below are examples of poor query triplets. Study them to understand exactly why they fail:
<bad_triplet_1>
– input image:
###IMAGE OF PERSON STANDING IN WHAT APPEARS TO BE A FLEA MARKET###
<query>What activity typically happens at this location at the end of the day?</query>
<target_desc>A person packing items into cardboard boxes and loading them into a vehicle.</target_desc>
<critique>The target description only states one possible answer, but beyond it there multiple valid answers: at the end of a flea market day, vendors could be doing several activities – dismantling tents, cleaning up, counting money, etc. The query doesn't uniquely specify one activity, violating <specificity_constraint>.</critique>
</bad_triplet_1>

<bad_triplet_2>
– input image:
###IMAGE OF A BUILDING WITH PILLARS IN FRONT###
<query>What did this structure look like in the years immediately following its construction?</query>
<target_desc>A pristine neoclassical stone building with clean light-colored stone facade.</target_desc>
<critique>Most readers will understand the query as: 'What did this exact building look like when it was newly built?'. So the only correct answer would be an image of the same building in its original state. That is unrealistic for target image retrieval, because Visual Genome and Open Images never contain multiple photos of the same building taken years apart.</critique>
</bad_triplet_2>

</Examples_of_bad_query_triplets>

<validation_checklist>

Before finalizing each triplet, verify:
☐ Query cannot be answered without the input image
☐ Reasoning requires 1–2 inference steps
☐ Only one logically correct target exists given the input image + query
☐ Target description is specific enough for retrieval
☐ Target would realistically exist in Open Images/Visual Genome
</validation_checklist>

<task_instructions>
Using the provided input image, generate one example query triplet for each reasoning category. Each example should:
- Use the same input image across all categories.
- Demonstrate the specific reasoning type for that category following the guidelines in <reasoning_complexity_constraint>.
- Satisfy all the requirements for the generated query and target image description in <input_dependence_constraint>, <specificity_constraint>, and <image_requirements>.
- Be numbered following <output_format> below.

<abstention_guidelines>
You need not forcibly generate all query triplets for every input image. Write "ABSTAIN" in the query, target_desc, and reasoning output fields if:
- The input image is a poor fit for a particular reasoning category, and you cannot generate a high quality query fulfilling all constraints.
- The input image is a screenshot or a digital poster.
</abstention_guidelines>

</task_instructions>

<output_format>
First write a description of the input image:
<input_desc>description of the provided input image</input_desc>
Then, generate each query triplet example following this exact structure:

<Example_[CATEGORY_CODE]_[NUMBER]>
<query>clearly stated question connecting the input image to the target image in < 25 words</query>
<target_desc>precise description with necessary and sufficient visual details to uniquely identify the target image in ≤ 50 words</target_desc>
<reasoning>concise explanation of the logical connection between the input image, the query, and the target image</reasoning>
</Example_[CATEGORY_CODE]_[NUMBER]>

NUMBERING CONVENTION:
- Format: Example_[CATEGORY_CODE]_1. E.g., Example_META_1, Example_FUNC_1
- Always start from 1 for each category
- Do NOT continue numbering from examples provided under any different category
</output_format>

<Example_query_generation>
Here is an example of an input image and the expected query generation output across the reasoning categories of interest.
Input image:
###IMAGE OF A LOCKED PADLOCK###
Expected Output:
<input_desc>A metal padlock securing a slightly rusted black metal latch on a wooden door or gate.</input_desc>

<Example_FUNC_1>
<query>What tool would render the object in the photo useless for its purpose?</query>
<target_desc>A bolt cutter.</target_desc>
<reasoning>A bolt cutter can be used to break open the lock and move forward past the door, rendering the lock useless.</reasoning>
</Example_FUNC_1>

<Example_CAUS_1>
<query>What would someone be unable to

do as a result of the situation shown here?</query>
<target_desc>A person from outside a building entering the building through a door.</target_desc>
<reasoning>The padlock is in its locked state and as a result it blocks entry to the building to outsiders.</reasoning>
</Example_CAUS_1>

<Example_TEMP_1>
<query>What might someone need if they try to use this in a few months without maintaining it?</query>
<target_desc>A spray can/bottle of a lubricant like WD-40.</target_desc>
<reasoning>Metal padlocks exposed to weather elements typically develop rust and corrosion over extended periods. Such a rusted padlock may be opened after lubricating its mechanism.</reasoning>
</Example_TEMP_1>
</Example_query_generation>

Now provide one example query triplet from each reasoning category for the following image: ###INPUT IMAGE GOES HERE###

A.2. Query Filter Prompt

<task_overview>
Your task is to review a candidate query triplet of (input image, query, target_desc) and validate whether it follows a set of predefined constraints. Pairs you validate will later be used to retrieve "target images".
</task_overview>

<input_format>
After all instructions, you will receive a candidate query triplet in the following format:
- input: [input image to which the query is applied]
- query: [a question which links the input image to target_desc through some reasoning]
- target_desc: [description of the target image representing an answer to the query + input image]
</input_format>

Below are the constraints you need to validate the candidate query triplet against.

<input_dependence_constraint>
- The target_desc must be determined jointly by the input image and the query, not by the query alone.
In other words, if we try to find target images using only the query, without considering the input image, we should NOT be able to correctly link the query to the target image described by target_desc.
✓ GOOD triplet: Connecting the query to the target_desc via a valid reasoning path necessitates understanding the input image.
× BAD triplet: target_desc can be directly connected to the query via a valid reasoning path while completely ignoring input image.

Test:
- Can this query be answered without the input? → If yes, it violates this constraint. E.g., if the triplet's query is "what do you need to vacuum clean a carpet?", it violates the input dependence constraint.
</input_dependence_constraint>

<specificity_constraint>
- The query should be specific enough that given the input image, there is only 1 logically correct answer.

Test:
- Are there multiple equally valid target_desc for this input image + query?. If yes, it violates this constraint.
</specificity_constraint>

<reasoning_complexity_constraint>
The reasoning pattern that connects the input image + query to a target image:
- Should be challenging enough to need 1-2 logical inference steps beyond surface-level observation, i.e., not

solvable by simple object detection alone
- Should test understanding of relationships, implications, or context
- Should be explainable in 1-2 sentences

<reasoning_categories>
Below I've described some reasoning patterns that the candidate query triplet may leverage to connect the input image + query to a target image.

FUNC (FUNCTIONAL/AFFORDANCE)

Understanding what tools or objects are needed to accomplish a specific task or function.
<Example_FUNC_1>
<input_desc>A screw with Phillips head pattern embedded in wood</input_desc>
<query>What tool would fit the pattern shown to remove this?</query>
<target_desc>A Phillips head screwdriver with metal shaft</target_desc>
<reasoning>The cross-shaped indentation in the screw head requires a matching Phillips head tool to engage and turn it.</reasoning>
</Example_FUNC_1>

<Example_FUNC_2>
<input_desc>Image of a car tire with visible nail puncture causing deflation</input_desc>
<query>What tool would seal this type of damage?</query>
<target_desc>A tire repair kit with rubber plugs and insertion tool</target_desc>
<reasoning>A puncture in a tire requires a repair kit with rubber plugs that can be inserted into the hole to create an airtight seal.</reasoning>
</Example_FUNC_2>

<Example_FUNC_3>
<input_desc>Image of raw dough rolled flat on a floured surface</input_desc>
<query>What tool would create the circular shapes needed before baking these?</query>
<target_desc>A round cookie cutter</target_desc>
<reasoning>Flat rolled dough requires a cutting tool to create individual shaped portions for baking.</reasoning>
</Example_FUNC_3>

CAUS (CAUSAL)

Understanding why or how something happens (cause → effect).
<Example_CAUS_1>
<input_desc>Magnifying glass focusing sunlight on paper</input_desc>
<query>What happens at the focal point?</query>
<target_desc>Burnt hole with charred edges in a paper sheet</target_desc>
<reasoning>The magnifying glass converges sunlight onto a tiny hot spot on the paper, making it burn.</reasoning>
</Example_CAUS_1>

<Example_CAUS_2>
<input_desc>Wet laundry hanging on outdoor clothesline in sunlight</input_desc>
<query>What state will these items be in after a sunny afternoon?</query>
<target_desc>Some dry clothes on clothesline</target_desc>
<reasoning>Sunlight and air circulation evaporate the water from wet fabric, leaving them dry.</reasoning>
</Example_CAUS_2>

<Example_CAUS_3>
<input_desc>Cat stepping on computer keyboard</input_desc>
<query>What appears on the monitor screen from this pressure?</query>
<target_desc>Random text characters on a computer screen</target_desc>
<reasoning>The cat's weight depresses multiple keys, inputting unintended random characters into the computer.</reasoning>
</Example_CAUS_3>

TEMP (TEMPORAL)

Understanding and ordering events in time, determining whether one event is before, after, or simultaneous with another.
<Example_TEMP_1>
<input_desc>Tadpole swimming in pond water</input_desc>
<query>What does this creature look like after completing its development?</query>
<target_desc>An adult frog</target_desc>
<reasoning>Tadpoles undergo metamorphosis over weeks, developing legs and losing their tail to become frogs.</reasoning>
</Example_TEMP_1>

<Example_TEMP_2>
<input_desc>Athletes running during a track and field competition</input_desc>
<query>What moment comes immediately before the start of this activity?</query>
<target_desc>Athletes positioned in starting blocks on a running track.</target_desc>
<reasoning>Right before the race begins, athletes are positioned at the starting line in their ready positions, waiting for the starting signal - stationary and prepared but not yet in motion.</reasoning>
</Example_TEMP_2>

<Example_TEMP_3>
<input_desc>Tree with pink cherry blossoms in spring</input_desc>
<query>What do these flowering branches produce by late summer?</query>
<target_desc>Red cherries hanging from a branch of a cherry tree</target_desc>
<reasoning>Cherry blossoms develop into fruit over several months after pollination.</reasoning>
</Example_TEMP_3>
</reasoning_categories>
</reasoning_complexity_constraint>

<query_language_style_constraint>
The query itself should be worded in a way that a typical native English speaker can easily understand it.
</query_language_style_constraint>

<task_instructions>
1. Describe this image first in tags without letting yourself get biased at all by the query. ⚠make sure to stay UNBIASED by the query because sometimes the query is written for the wrong image.
2. Then, analyze the provided candidate query triplet of (input image, query, target_desc) validating whether it adheres to the constraints in <input_dependence_constraint>, <specificity_constraint>, <reasoning_complexity_constraint>, and <query_language_style_constraint>. Summarize your reasoning in <reasoning> tags, and enter your final assessment in <is_valid> tags. Your final assessment should be:
- yes, if the candidate query triplet reasonably adheres to all constraints.
- no, if the candidate query triplet clearly violates at least one constraint.
</task_instructions>

<Examples_of_bad_query_triplets>
Below are examples of poor query triplets. Study them to understand exactly why they fail:

<bad_triplet_1>
- input image:
###IMAGE OF PERSON STANDING IN WHAT APPEARS TO BE A FLEA MARKET###
newline <query>What activity typically happens at this location at the end of the day?</query>
<target_desc>A person packing items into cardboard boxes and loading them into a vehicle.</target_desc>
<critique>The target description only states one possible answer, but beyond it there multiple valid answers: at the end of a flea market day, vendors could be doing several activities - dismantling tents, cleaning up, counting money,

```
etc.  The query doesn't uniquely
specify one activity, violating
<specificity_constraint>.</critique>
</bad_triplet_1>

<bad_triplet_2>
- input image:
###IMAGE OF A BUILDING WITH PILLARS IN
FRONT###
<query>What did this structure look
like in the years immediately following
its construction?</query>
<target_desc>A pristine neoclassical
stone building with clean light-colored
stone facade.</target_desc>
<critique>Most readers will understand
the query as:  'What did this exact
building look like when it was
newly built?'.  So the only correct
answer would be an image of the same
building in its original state.  That
is unrealistic for target image
retrieval, because Visual Genome and
Open Images never contain multiple
photos of the same building taken years
apart.</critique>
</bad_triplet_2>
</Examples_of_bad_query_triplets>

<output_format>
Generate your output using exactly this
format:
<input_desc>Description of the input
image generated independently without
getting biased by the query or the
target_desc.</input_desc>
<reasoning>Summary of your assessment
of whether or not the query triplet
adheres to all constraints.</reasoning>
<is_valid>Enter ONLY one of:
yes/no</is_valid>
</output_format>

Now, evaluate this candidate query
triplet:
- input image:###INPUT IMAGE GOES
HERE###
- query:  INPUT QUERY GOES HERE
- target_desc:  TARGET DESCRIPTION GOES
HERE
```

A.3. Query Triplet Judge Prompt

```
<task_overview>
For a given candidate query triplet
(input image, query, target image),
your task is to critically evaluate
whether the target image represents
a logically consistent answer to the
input image + query pair.
Note that the connection between the
input+query and the target may require
intermediate reasoning (1-2 inference
steps), not just surface-level
matching.  Apply strict standards
- accept only clear, unambiguous
connections.
The candidate query triplets you'll be
provided with will involve one of these
reasoning categories:

<reasoning_categories>
--------------------------
FUNC (FUNCTIONAL/AFFORDANCE)
--------------------------
Understanding what tools or objects are
needed to accomplish a specific task or
function.
<Example_FUNC_1>
<input_desc>A screw with Phillips head
pattern embedded in wood</input_desc>
<query>What tool would fit the pattern
shown to remove this?</query>
<target_desc>A Phillips
head screwdriver with metal
shaft</target_desc>
<reasoning>The cross-shaped indentation
in the screw head requires a matching
Phillips head tool to engage and turn
it.</reasoning>
</Example_FUNC_1>

<Example_FUNC_2>
<input_desc>Image of a car tire
with visible nail puncture causing
deflation</input_desc>
<query>What tool would seal this type
of damage?</query>
<target_desc>A tire repair kit
with rubber plugs and insertion
tool</target_desc>
<reasoning>A puncture in a tire
requires a repair kit with rubber plugs
that can be inserted into the hole to
create an airtight seal.</reasoning>
```

</Example_FUNC_2>

<Example_FUNC_3>
<input_desc>Image of raw dough rolled flat on a floured surface</input_desc>
<query>What tool would create the circular shapes needed before baking these?</query>
<target_desc>A round cookie cutter</target_desc>
<reasoning>Flat rolled dough requires a cutting tool to create individual shaped portions for baking.</reasoning>
</Example_FUNC_3>

CAUS (CAUSAL)

Understanding why or how something happens (cause → effect).
<Example_CAUS_1>
<input_desc>Magnifying glass focusing sunlight on paper</input_desc>
<query>What happens at the focal point?</query>
<target_desc>Burnt hole with charred edges in a paper sheet</target_desc>
<reasoning>The magnifying glass converges sunlight onto a tiny hot spot on the paper, making it burn.</reasoning>
</Example_CAUS_1>

<Example_CAUS_2>
<input_desc>Wet laundry hanging on outdoor clothesline in sunlight</input_desc>
<query>What state will these items be in after a sunny afternoon?</query>
<target_desc>Some dry clothes on clothesline</target_desc>
<reasoning>Sunlight and air circulation evaporate the water from wet fabric, leaving them dry.</reasoning>
</Example_CAUS_2>

<Example_CAUS_3>
<input_desc>Cat stepping on computer keyboard</input_desc>
<query>What appears on the monitor screen from this pressure?</query>
<target_desc>Random text characters on a computer screen</target_desc>
<reasoning>The cat's weight depresses multiple keys, inputting unintended random characters into the computer.</reasoning>
</Example_CAUS_3>

TEMP (TEMPORAL)

Understanding and ordering events in time, determining whether one event is before, after, or simultaneous with another.
<Example_TEMP_1>
<input_desc>Tadpole swimming in pond water</input_desc>
<query>What does this creature look like after completing its development?</query>
<target_desc>An adult frog</target_desc>
<reasoning>Tadpoles undergo metamorphosis over weeks, developing legs and losing their tail to become frogs.</reasoning>
</Example_TEMP_1>

<Example_TEMP_2>
<input_desc>Athletes running during a track and field competition</input_desc>
<query>What moment comes immediately before the start of this activity?</query>
<target_desc>Athletes positioned in starting blocks on a running track.</target_desc>
<reasoning>Right before the race begins, athletes are positioned at the starting line in their ready positions, waiting for the starting signal - stationary and prepared but not yet in motion.</reasoning>
</Example_TEMP_2>

<Example_TEMP_3>
<input_desc>Tree with pink cherry blossoms in spring</input_desc>
<query>What do these flowering branches produce by late summer?</query>
<target_desc>Red cherries hanging from a branch of a cherry tree</target_desc>
<reasoning>Cherry blossoms develop

into fruit over several months after pollination.</reasoning>
</Example_TEMP_3>
</reasoning_categories>
</task_overview>

<content_policy>
If the input or target images depict content that is politically sensitive (protests, police actions, riots, conflicts, political figures, etc.) or sexually explicit, immediately decline the task by responding only with: "ABSTAIN DUE TO SENSITIVE IMAGE"
</content_policy>

<input_format>
After all instructions, you will receive a candidate query triplet in the following format:
- INPUT_IMAGE: [the input image to which the query is applied]
- QUERY: [the query/question]
- TARGET_IMAGE: [the target image representing an answer to the query + input image]
</input_format>

<evaluation_criteria>
Follow these guidelines to evaluate the logical consistency of the query triplet:
- Accept the triplet and return True if: The target logically follows from applying the query to the input. Accept only if the connection is clear enough that a typical adult would make it within 30 seconds of viewing.
- Reject the triplet and return False if: No reasonable interpretation connects the input+query to the target.
- Return ABSTAIN only if: Critical information needed to evaluate the triplet's connection is missing or unclear.
</evaluation_criteria>

<output_format>
Generate your output using exactly this format:
<target_desc>Description of the target image.</target_desc>
<input_desc>Description of the input image.</input_desc>
<reasoning>Summary of your assessment of whether or not the query triplet is logically consistent.</reasoning>
<consistent>True/False/ABSTAIN</consistent>
</output_format>

<Examples>
Here are some example evaluations of query triplets:
<Example_1>
INPUT_IMAGE: ###IMAGE OF A LOCKED PADLOCK###
QUERY: What do you need to open it?
TARGET_IMAGE: ###IMAGE OF KEYS###
Expected response:
<target_desc>Two metal keys on a ring with red figure keychain.</target_desc>
<input_desc>A metal padlock securing a door or gate.</input_desc>
<reasoning>The input image shows a padlock securing a door or gate. The query asks what is needed to open it. The target image shows keys, which are the appropriate tool needed to unlock a padlock.</reasoning>
<consistent>True</consistent>
</Example_1>

<Example_2>
INPUT_IMAGE: ###IMAGE OF A SWIMMING POOL###
QUERY: What does this look like at 8pm with guests?
TARGET_IMAGE: ###IMAGE OF A SWIMMING POOL AT NIGHT###
Expected response:
<target_desc>Illuminated outdoor swimming pool at night.</target_desc>
<input_desc>An outdoor swimming pool during daytime with city buildings in the background.</input_desc>
<reasoning>The input image shows a pool during the day, and the target image shows a pool at night, which is indeed what the input image would look like at 8pm. However, the query also specifies that the target shows "guests", which it does not. Hence, the target image is not a valid answer to the query + input image.</reasoning>
<consistent>False</consistent>
</Example_2>

<Example_3>
INPUT_IMAGE: ###IMAGE OF A WOMAN WITH HAIR BACK###
QUERY: What might this hairstyle look like after 8 hours of wear?
TARGET_IMAGE: ###IMAGE OF A WOMAN WITH MESSY, FRIZZED HAIR###
Expected response:
<target_desc>Person with red hair in a ponytail, with messy bangs covering their face.</target_desc>
<input_desc>Smiling woman with dark, curly hair pulled back.</input_desc>
<reasoning>On the one hand, the hair in the target image is considerably messier than that in the input image, which is what one would expect to happen after 8 hours. On the other hand, the hair itself looks quite a bit different from the hair in the input image, with the input image having black hair and the target image having red hair. Because reasonable people could easily disagree about the correct interpretation of the query, we need to abstain on this instance.</reasoning>
<consistent>ABSTAIN</consistent>
</Example_3>

<Example_4>
INPUT_IMAGE: ###IMAGE OF AN EAGLE###
QUERY: What serves this role in ocean ecosystems?
TARGET_IMAGE: ###IMAGE OF A SHARK###
Expected response:
<target_desc>A great white shark underwater with its mouth open.</target_desc>
<input_desc>A bald eagle behind a wire fence.</input_desc>
<reasoning>The input image shows a bald eagle, which is an apex predator. The query asks what serves this role (i.e., apex predator) in ocean ecosystems. The target correctly shows a great white shark, which is a marine apex predator.</reasoning>
<consistent>True</consistent>
</Example_4>

<Example_5>
INPUT_IMAGE: ###IMAGE OF A COW IN A FIELD###
QUERY: What naturally appears on this field as a result of this animal's activity here?
TARGET_IMAGE: ###IMAGE OF A FIELD COVERED IN PINE NEEDLES AND SEED PODS###
Expected response:
<target_desc>A forest-floor scene carpeted in dry brown pine needles, scattered leaves, and hundreds of small spiky seed pods, likely sweetgum balls.</target_desc>
<input_desc>A black and white spotted dairy cow grazing on green grass in a pastoral field with a body of water and hills visible in the background.</input_desc>
<reasoning>The query asks what naturally appears on the field as a result of the cow's grazing seen in the input image. The expected target image would show cow manure/dung - brown matter scattered across the pasture as a byproduct of the grazing cow's digestive processes. The target image, though superficially resembling scattered manure, actually shows a forest floor with dry pine needles and spiky seed pods. There is no logical connection between the grazing cow input image and the ground covered with tree seed pods in the target image.</reasoning>
<consistent>False</consistent>
</Example_5>

<Example_6>
INPUT_IMAGE: ###IMAGE OF A PERSON IN FLIGHT SIMULATOR TRAINING###
QUERY: What credential does this person earn after completing extensive training in this environment?
TARGET_IMAGE: ###IMAGE OF A PERSON STANDING BY THE DOOR OF A SMALL AIRPLANE###
Expected response:
<target_desc>A woman standing proudly in front of a small aircraft.</target_desc>
<input_desc>A black and white photograph showing a person wearing a flight helmet sitting in what appears

```
to be a flight simulator or training
cockpit, with control panels and
instruments visible above and around
them.</input_desc>
<reasoning>The input image shows
someone training in an aircraft cockpit
or flight simulator environment.  The
query asks what credential this person
earns after completing extensive
training in this environment.  The
target image shows a woman standing
next to a small aircraft, which could
represent someone who has completed
pilot training.  However, the target
image doesn't clearly show or indicate
any specific credential (like a pilot's
license, certificate, or wings) - it
simply shows someone posing with an
aircraft.  While there's a logical
connection between flight training
and becoming a pilot, crucially
the target image doesn't clearly
represent the 'credential' aspect of
the query.</reasoning>
<consistent>False</consistent>
</Example_6>

<Example_7>
INPUT_IMAGE: ###IMAGE OF A CHILD
STANDING ON A SELF-BALANCING VEHICLE###
QUERY: What would happen if she
suddenly leaned too far forward?
TARGET_IMAGE: ###BLURRY IMAGE OF A SHOE
AND THE BOTTOM OF JEANS###
Expected response:
<target_desc>A blurry, motion-streaked
close-up of someone's foot and lower
leg with blue jeans and sneakers
visible.</target_desc>
<input_desc>A young girl wearing a
black helmet and casual clothing riding
a self-balancing scooter (Segway-style
device) with a handlebar on a paved
outdoor area.</input_desc>
<reasoning>The query asks about
the consequence of the girl on the
scooter in the input image leaning
too far forward.  If this happened,
she would likely lose her balance
and fall forward.  The target image
shows a foot/lower leg in rapid
movement.  While the target image
shows motion that could theoretically
be related to losing balance, it does
NOT definitively show someone falling
FORWARD, which is a crucial detail,
making it inconsistent with the query +
input image.</reasoning>
<consistent>False</consistent>
</Example_7>
</Examples>

Now evaluate this candidate query
triplet:
INPUT_IMAGE: ###INPUT IMAGE GOES
HERE###
QUERY: INPUT QUERY GOES HERE
TARGET_IMAGE: ###TARGET IMAGE GOES
HERE###
```

B. Human Validation

Though a full scale validation of certain components was beyond the scope of this paper, we were able to complete small-scale human validation of two components, discussed below, which lend credence to the validity of our approach.

B.1. Validation of LLM Judgments

To validate LLM-based triplet judgments, we conducted a human audit on 200 query triplets (100 positive, 100 negative LLM judgments) randomly sampled from the test set. Results show 81% agreement with human assessment, consistent across reasoning categories: causal (80%), functional (84%), and temporal (80%), indicating no systematic failure on any reasoning type.

We argue this accuracy suffices for the benchmark's purpose. Despite some noise, the benchmark discriminates reasoning retrieval capabilities: smaller reasoning-driven embedding models (UME-R1-2B) outperform larger MLLMs (Qwen2.5-VL-7B), demonstrating adequate signal-to-noise for meaningful evaluation.

Our pipeline uses a single model family (Claude) due to cost and access constraints; future work could improve accuracy with an ensemble of diverse judges.

More broadly, benchmark construction involves a tradeoff between scalability and annotation certainty; our approach prioritizes scalability, a choice validated by the benchmark's ability to produce meaningful model rankings.

B.2. Validation of TSR Effectiveness

Do high TSR scores *actually* correlate with fewer false negatives? While a definitive answer would require labeling the entire candidate pool of 35K+ images, we ran a

simple experiment that provided surprisingly strong evidence for TSR effectiveness.

Our pipeline retrieves up to 16 candidate images per query. To evaluate TSR effectiveness, we sampled 75 queries where the last retrieved element received a positive LLM judgment and 75 where it received a negative judgment, then manually labeled each to obtain ground truth (yielding 68 positives and 82 negatives). We computed TSR on the 15 preceding elements (padding with synthetic negatives as needed; see **??**) and compared TSR scores between the groups where the held-out 16^{th} element was human-labeled as positive vs. negative.

The positive group (where we *know* there is another positive instance beyond those on which we computed TSR) had an average TSR score **0.27 lower** than the negative group. A two-tailed t-test yielded a **p-value of $2.38 * 10^{-11}$**, strongly rejecting the null hypothesis of identical distributions and supporting TSR as a predictor of additional positives (*i.e.*, false negatives) beyond the top-ranked results.