Semantic Coherence-Aware Evidence Filtering for Retrieval-Augmented Visual Question Answering

Sung Geun An* Gyeongseop Lee* Jooyoung Kim*
Seoul National University
{ssunggun2, pepsipower, gracekim15237}@snu.ac.kr

Abstract

Recent advancements in multimodal Visual Question Answering (VQA) have leveraged Retrieval-Augmented Generation (RAG) to enhance answer accuracy by incorporating external images and texts. However, current RAG-based VQA systems typically rank retrieved evidence solely based on relevance to the input query, while ignoring semantic coherence among the selected evidence. This often leads to inconsistent or contradictory inputs that result in hallucinated answers. In this work, we propose a lightweight, plug-and-play ***semantic coherence module*** *that can be integrated into existing RAG-VQA pipelines without fine-tuning the underlying retriever. Our two-stage approach first filters evidence using query-level similarity, then assesses inter-evidence consistency via cross-modal alignment metrics. By removing incoherent image-text pairs prior to generation, our method reduces hallucinations and improves factuality. We demonstrate the effectiveness of our module on real-world VQA datasets, showing improvements in both answer quality and system robustness across diverse scenarios.*

1. Introduction

Recent advances in large language models (LLMs) have substantially expanded the capabilities of Visual Question Answering (VQA) systems, particularly when integrated with Retrieval-Augmented Generation (RAG). In multimodal RAG-based VQA, external knowledge in the form of images and text is retrieved from a corpus to support the generation of accurate and grounded answers. This has enabled applications in high-stakes domains such as sinkhole risk assessment, wildfire response, and urban planning, where reliable multimodal reasoning is critical.

Despite this progress, current multimodal RAG-based VQA systems generally follow a *retrieve-then-answer* pipeline that ranks evidence (e.g., documents or images) based only on their similarity to the input query. These systems typically ignore semantic coherence between the retrieved pieces of evidence. As a result, retrieved contents may be topically relevant to the query but mutually inconsistent, redundant, or even contradictory. This lack of internal consistency among evidence can mislead the generative model and result in hallucinated or erroneous answers, undermining reliability in decision-critical settings.

Prior research has attempted to enhance RAG-based VQA through improved retrievers (e.g., MuRAG [4]) and answer validation techniques (e.g., MAVEx [14]), but these approaches typically assess evidence relevance at an individual level. They fail to explicitly model inter-evidence consistency, which is particularly important when integrating multiple modalities or sources. As such, existing methods do not sufficiently mitigate conflicts among retrieved results, especially across retrieved image-text pairs.

To address this gap, we propose a plug-and-play semantic coherence module that can be integrated into any RAG-VQA pipeline without fine-tuning the underlying retriever or multimodal language language model (MLLM). Our module evaluates semantic coherence across the retrieved evidence (image-text pairs) and removes incoherent items, allowing only semantically consistent evidence to be used in answer generation.

To be specific, our approach involves a two-stage architecture. In the first stage, we compute pairwise semantic similarity between the query and retrieved evidence using multimodal embedding models such as CLIP [10]. In the second stage, we evaluate the internal coherence of the retrieved evidence set via cross-modal alignment metrics (e.g., CLIPScore or BLIP [7] matching), and perform selective filtering. This design enables our module to be lightweight, domain-agnostic, and easily deployable on top of existing RAG-VQA frameworks.

We demonstrate that integrating our semantic coherence module significantly reduces hallucinations and improves answer consistency in RAG-based VQA. Our contributions are summarized as follows:

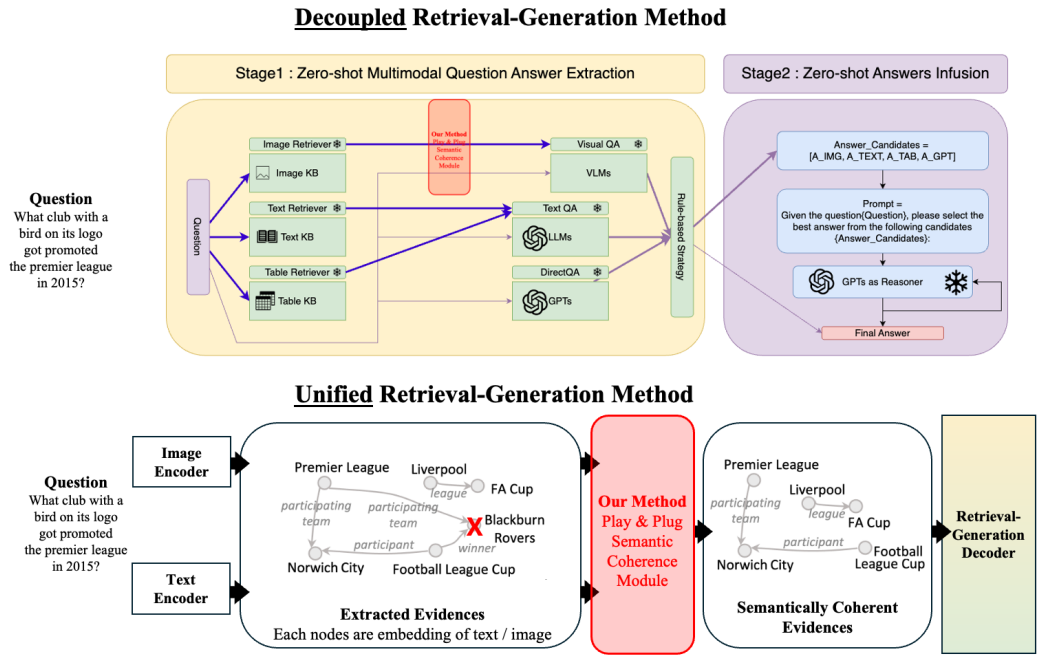

Figure 1. **Overview of two representative RAG-based VQA architectures.** Top: a decoupled pipeline (MoqaGPT) using fixed retrievers and reasoning modules. Bottom: a unified retrieval-generation model (SKURG). Our proposed semantic coherence module (highlighted in red) can be inserted in both cases to filter incoherent evidence.

- We identify and formalize the problem of inter-evidence semantic inconsistency in multimodal RAG-based VQA systems.
- We propose a modular, retriever-agnostic framework for semantic coherence filtering across multimodal evidence.
- We empirically show that our method enhances VQA accuracy and reliability on real-world datasets with MULTIMODALQA (MMQA) dataset [13]

2. Preliminaries

While recent RAG-based VQA models vary widely in architecture, we observe that they often fall into two distinct implementation patterns based on how evidence retrieval and answer generation are coordinated. Specifically, we identify (1) **retrieval-decoupled** architectures, where retrieval and generation are modularized as separate components, and (2) **retrieval-integrated** architectures, where these steps are performed jointly in a unified pipeline. To motivate the general applicability of our proposed semantic coherence module, we briefly describe representative examples of each: MoqaGPT [16] and SKURG [15]. **Figure 1** illustrates the overall structures and shows where our module can be inserted in both designs.

2.1. MoqaGPT: Decoupled Retrieval-Generation

MoqaGPT [16] exemplifies a decoupled architecture that separates retrieval and answer generation into distinct stages. The model first retrieves modality-specific evidence using fixed retrievers for text and images, without mapping them into a unified embedding space. These retrieved evidences are passed to zero-shot vision-language or text-based models to extract answer candidates independently. Finally, a rule-based strategy combined with LLM-based reasoning selects the final answer among these candidates. Our semantic coherence module can be inserted between retrieval and reasoning stages to filter out incoherent evidence, as illustrated in the top part of Figure 1.

2.2. SKURG: Unified Retrieval-Generation

In contrast, SKURG [15] follows a unified pipeline that jointly performs retrieval and generation in an end-to-end manner. It introduces an entity-centered fusion encoder that constructs a knowledge graph from multi-modal inputs. These fused entity representations are used by a unified retrieval-generation decoder, which retrieves and integrates evidences on-the-fly through a pointer mechanism during answer generation. This design tightly couples retrieval and gener-

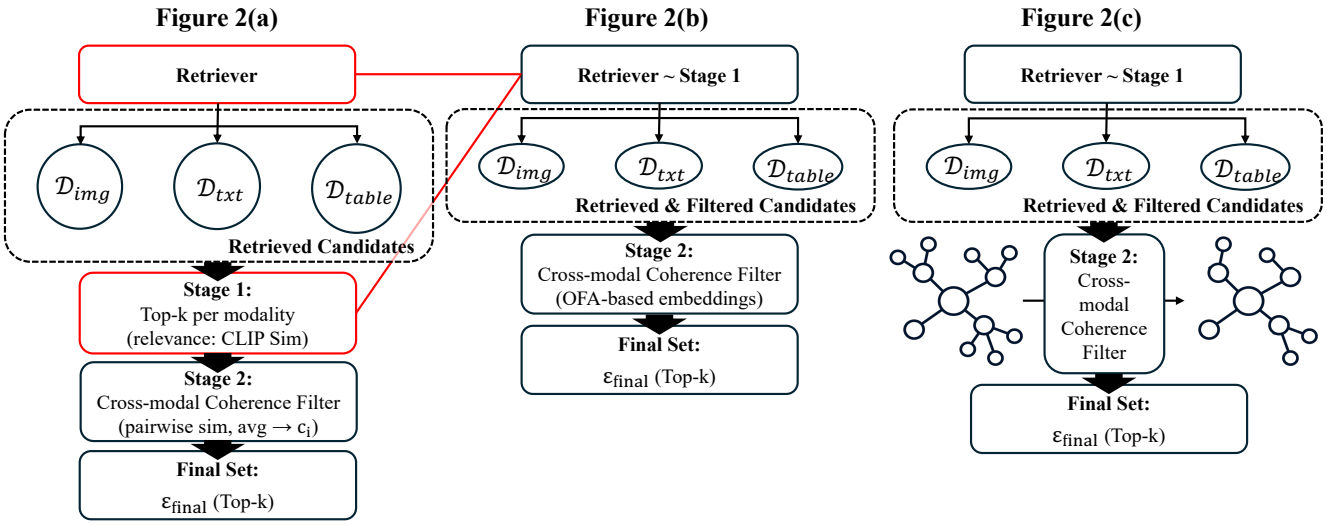

Figure 2. Integration of proposed module in: (a) MoqaGPT, (b) SKURG (embedding-based), (c) SKURG (graph-based).

ation, improving coherence through iterative reasoning across modalities. Although tightly integrated, SKURG still offers insertion points for external filtering modules: our coherence module is applied after entity fusion and before generation to prune conflicting evidences,as shown in the bottom half of Figure 1, thus enhancing semantic alignment while preserving SKURG' s joint architecture.

3. Methodology

In this section, we describe the overall design of our proposed 2-stage *semantic coherence* module and how it is integrated as a plug-and-play module into two representative RAG-based VQA systems (MoqaGPT and SKURG). The core idea is to evaluate and prune multimodal evidence sets for semantic consistency, thereby improving the end-to-end performance and reliability of existing models.

3.1. 2-Stage Coherence Module

Given a question q and multimodal evidence pools-images $\mathcal{D}_{\text{img}}$, text passages $\mathcal{D}_{\text{text}}$, and tables $\mathcal{D}_{\text{table}}$-our goal is to select a final evidence set $\mathcal{E}_{\text{final}}$ of size k that is both highly relevant to q and mutuallly coherent. To achieve this, we introduce a lightweight, two-stage coherence module that sits between retrieval and answer generation.

In the first stage, we compute a relevance score for each candidate d in each modality via a pretrained multimodal model (e.g. CLIP). Concretely, for modality m we calculate

$$r_d = \texttt{Sim}(q, d)$$

and then keep only the top-k items per modality:

$$\mathcal{E} = \texttt{TopK}\{r_d | d \in \mathcal{D}_m\}.$$

We merge these sets into a combined pool $\mathcal{E} = \mathcal{E}_{\text{img}} \cup \mathcal{E}_{\text{text}} \cup \mathcal{E}_{\text{table}}$ which typically has up to $3k$ candidates.

In the second stage, we refine $\mathcal{E}$ by measuring each item's coherence with the rest of the pool. For each candidate $e_i \in \mathcal{E}$, we compute the average pairwise similarity to all other items:

$$c_i = \frac{1}{|\mathcal{E}| - 1} \sum_{j \neq i} \texttt{Sim}(e_i, e_j)$$

Finally, we select the top-k candidates under c_i for form $\mathcal{E}_{\text{final}}$. This two-step process ensures that the retained evidence is not only individually relevant to the query but also semantically consistent as a group.

3.2. Module Description

1. **Stage 1: Modality-wise relevance filtering**
   - Compute similarity scores between the question and each retrieved evidence item per modality (e.g., image, text, table).
   - For each modality, select the top-k most relevant evidence items.
   - If necessary, repeat retrieval until enough items are gathered.
2. **Stage 2: Cross-modal coherence refinement**
   - Construct a merged pool of selected evidence across all modalities.
   - Evaluate each item's coherence with others in the pool using pairwise similarity.
   - Select the final top-k most coherent evidence

items for answer generation.

3.3. Integration in MoqaGPT

Figure 1 shows the insertion point of our semantic coherence module in the MoqaGPT pipeline. MoqaGPT separates evidence retrieval and answer generation into distinct stages, allowing external modules to modify the retrieved evidence before generation.

In Stage 1, the module selects the top-k relevant evidence items from each modality based on similarity to the question. If fewer than k items remain, retrieval is repeated. In Stage 2, the module computes pairwise similarity among the pooled evidence items and selects a subset with the highest coherence.

This procedure, described in Algorithm 1, outputs a filtered evidence set that is then passed to the modality-specific QA models. Each model generates an answer candidate independently. These candidates are subsequently processed by a rule-based strategy that selects a subset of plausible answers based on modality type and confidence heuristics. The selected candidates are then concatenated into a templated prompt and passed to a final LLM-based reasoning module, which produces the final answer.

3.4. Integration in SKURG

Our coherence module is integrated into SKURG pipeline between retrieval stage and knowledge graph construction, before answer generation. Stage 1 filtering is applied in the same manner as described for MoqaGPT, using CLIP-based scoring to select top-k relevant evidence items from each modality.

For Stage 2, we implement two variants of coherence filtering. The first is an embedding-based approach, where each retrieved evidence is encoded using SKURG's internal multimodal encoder (e.g., OFA), and pairwise similarity is computed using pooled representations. A coherent subset is selected based on average similarity scores across modalities.

The second is a graph-based approach that operates over the constructed knowledge graph. In this variant, each entity in evidence items is represented as a node, and edges reflect shared or related entities identified during graph construction. Each entity node is annotated with the set of evidence sources (text or image) in which it appears. Nodes that occur in only one source and are not relationally connected to entities from other sources (i.e., degree 1 without cross-source edges) are pruned to maintain structural coherence.

By selectively retaining evidence both relevant and mutually coherent, our module enhances the consistency and reliability of answers generated by SKURG.

Algorithm 1 in Decoupled Retrieval-Generation

Input: A question q,
Retrieved evidence $\mathcal{D} = \{D_{\text{img}}, D_{\text{text}}, D_{\text{table}}\}$,
Target number of evidence items k
Output: Coherent evidence set $\mathcal{E}_{\text{final}}$

`// Stage1: Modality-wise relevance filtering`
$\mathcal{E} \leftarrow \emptyset$ **foreach** *modality* $m \in \mathcal{D}$ **do**
  foreach *document* $d \in D_m$ **do**
    Compute relevance score $r_d = \texttt{Sim}(q, d)$
  Select top-k items $\mathcal{E}_m$ by r_d $\mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{E}_m$

`// Stage2: Cross-modal coherence refinement`
foreach $e_i \in \mathcal{E}$ **do**
  Compute coherence score
  $c_i = \frac{1}{|\mathcal{E}|-1} \sum_{j \neq i} \texttt{Sim}(e_i, e_j)$
Select top-k items $\mathcal{E}_{\text{final}}$ by c_i
return $\mathcal{E}_{final}$

Algorithm 2 in Unified Retrieval-Generation

Input: A question q,
Retrieved evidence $\mathcal{D} = \{D_{\text{img}}, D_{\text{text}}, D_{\text{table}}\}$,
Target number of evidence items k
Output: Coherent evidence set $\mathcal{E}_{\text{final}}$

`// Stage1: Modality-wise relevance filtering`
$\mathcal{E} \leftarrow \emptyset$ **foreach** *modality* $m \in \mathcal{D}$ **do**
  foreach *document* $d \in D_m$ **do**
    Compute relevance score $r_d = \texttt{Sim}(q, d)$
  Select top-k items $\mathcal{E}_m$ by r_d $\mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{E}_m$

`// Stage2: Cross-modal coherence refinement`
`// Method A: Embedding-based filtering`
foreach $e_i \in \mathcal{E}$ **do**
  Compute coherence score
  $c_i = \frac{1}{|\mathcal{E}|-1} \sum_{j \neq i} \texttt{Sim}(e_i, e_j)$
Select top-k items $\mathcal{E}_{\text{final}}$ by c_i

`// Method B: Graph-based filtering`
Construct knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\text{edge}})$ from entity set $\mathcal{E}$ with source annotations
foreach *entity node* $v_i \in \mathcal{V}$ **do**
  Compute degree $d_i = \deg(v_i)$
  Identify # of distinct modalities associated with v_i
  if $d_i \leq 1$ ***and*** v_i *is not connected to other modalities* **then**
    Prune v_i from $\mathcal{G}$
Select top-k items $\mathcal{E}_{\text{final}}$ by d_i
return $\mathcal{E}_{final}$

4. Experiments

Our experiments evaluate the proposed semantic-check module using a multimodal VQA benchmark that

emphasizes multi-step reasoning across heterogeneous modalities such as text, tables, and images. The dataset is well-suited for assessing whether the filtering of semantically inconsistent evidence can improve the coherence and factual correctness of generated answers.

4.1. Experimental Protocols

4.1.1. Dataset

We selected MMQA as our primary benchmark due to its moderate dataset size, which makes experimentation feasible under limited computational resources. More importantly, MMQA provides a well-structured setting for evaluating semantic coherence across heterogeneous modalities, as it explicitly requires the integration of multiple evidence sources to produce accurate answers. This aligns closely with our goal of assessing the effectiveness of coherence filtering in retrieval-augmented multimodal QA pipelines.

4.1.2. Metrics

We primarily evaluate model performance using Exact Match (EM) and F1 scores. EM measures strict correctness by computing the proportion of predictions that exactly match the ground truth answers. In contrast, the F1 score accounts for partial correctness by measuring the harmonic mean of precision and recall over token overlap between the predicted and reference answers.

We report results under three evaluation categories:

- **Single Modality**: Evaluation is conducted on questions that can be answered using evidence from a single modality (e.g., text, image, or table). This setting isolates the performance of unimodal pipelines and measures their capacity to extract modality-specific information.
- **Multi Modality**: This category includes questions that require reasoning over multiple modalities. It evaluates the model's ability to perform cross-modal integration and synthesize heterogeneous evidence.
- **Overall**: Represents aggregate performance over the entire evaluation set, encompassing both single- and multi-modality questions.

These metrics provide insight into how well the model performs under varying reasoning demands, and help quantify the benefit of coherence-aware evidence filtering in both unimodal and cross-modal settings.

4.1.3. Baselines

We select MoqaGPT and SKURG as baselines to evaluate our coherence module. MoqaGPT adopts a retrieval-decoupled design, where modality-specific evidence is retrieved first and candidate answers are generated separately, allowing straightforward insertion of our module between retrieval and reasoning stages.

In contrast, SKURG performs retrieval and generation jointly using an entity-centric fusion encoder. Our module is applied after entity fusion to filter incoherent evidence before decoding. These two setups enable us to assess the module's effectiveness in both modular and tightly integrated pipelines.

4.1.4. Implementation Details

We retrieve up to 10 candidates per modality and apply our filtering module to select the top-k ($k = 5$) coherent items based on

All models are accessed through their official APIs or HuggingFace implementations. Retrieval is performed over a fixed corpus of 10 references per modality, and the filtering module selects the top-k evidence items before answer generation. Coherence scores are computed using pairwise cosine similarity among projected embeddings.

MoqaGPT We build our module on the MoqaGPT pipeline using a set of pre-trained models for each modality. For text retrieval, we employ `all-mpnet-base-v2`[11], a transformer-based sentence encoder that retrieves relevant passages based on semantic similarity. Image retrieval is conducted using `CLIP` (ViT-B/32)[10]. To handle structured data, we use `ADA-002`[9], an OpenAI embedding model optimized for retrieving table content.

For evidence scoring and selection, our module supports multiple similarity metrics, including L1 distance, Euclidean distance, and CLIP-based semantic similarity.

For answer generation, the retrieved text and table evidence are combined with the question and processed by `gpt-3.5-turbo`[2]. For image-based questions, image-question pairs are input to `BLIP-2`[8]. Finally, all candidate answers from each modality are scored and aggregated by a reasoning module based on `gpt-3.5-turbo`.

SKURG For encoding inputs, SKURG uses BART-base encoder for text and tables, and OFA-base encoder for images. Entities are extracted from evidence content using a pre-trained ELMo-based NER model, and are linked across modalities to construct a unified knowledge graph for each question. Implementation details in Stage 1 follows that of MoqaGPT.

For Stage 2 (coherence filtering), we support two strategies. In the embedding-based variant, similarity is computed using pooled representations from SKURG's internal encoders (e.g., OFA for image, BART for text). In the graph-based variant, the knowledge graph is implemented as a dictionary where each key represents an entity node and the corresponding value lists the entities it is connected to.

Scoring Strategy	Single Modality		Multi Modality		Overall	
	F1	EM	F1	EM	F1	EM
MoqaGPT Baseline	47.78	40.40	31.17	26.38	40.64	34.37
MoqaGPT Baseline(w/o image modality)	46.48	38.39	29.61	25.33	39.22	32.77
Stage 1 (L1 Similarity)	47.28	40.19	30.14	25.71	39.91	33.96
Stage 1 (Euclidean Similarity)	47.98	40.62	30.89	26.38	40.63	34.49
Stage 1 (CLIP Similarity)	**48.10**	**40.76**	30.65	26.00	40.59	34.41
Stage 2 (L1 + Pairwise Filtering)	47.28	39.97	30.71	26.38	40.15	34.13
Stage 2 (Euclidean + Pairwise)	**48.05**	**40.55**	**31.46**	**26.67**	**40.92**	**34.58**
Stage 2 (CLIP + Pairwise)	47.72	40.33	30.56	25.71	40.34	34.04
SKURG Baseline	65.91	62.78	55.81	51.14	61.29	58.80
Query-Level (CLIP Similarity)	65.59	62.41	55.49	50.83	60.96	58.43
Coherence-Aware (Embedding-based)	65.47	62.63	55.76	**51.20**	**61.31**	**58.85**
Coherence-Aware (Graph-based)	**66.23**	**63.05**	**56.07**	**51.48**	**61.58**	**59.11**

Table 1. Comparison of evidence scoring strategies on the MultiModalQA benchmark. Query-level methods rank references by query-to-document similarity, while coherence-aware methods further refine selection via cross-modal consistency.

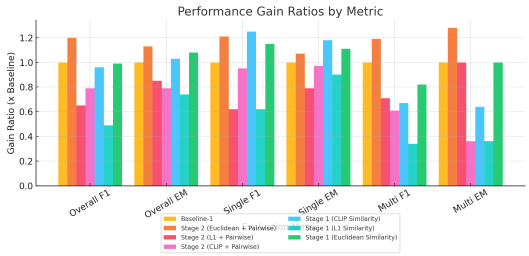

Figure 3. Performance comparison across modality settings in MoqaGPT.

We also observe that the overall impact of the image modality remains limited. The difference between the text-only and text-image baselines is small, and further applying coherence filtering to the image branch does not yield substantial improvement. This is reflected in the minimal performance delta across configurations involving image evidence, as shown in the lower part of Figure 3. The gain ratio helps clarify these marginal effects by quantifying the relative benefit of adding or filtering each modality.

4.2. Model Performance

4.2.1. Overall Performance

We evaluate the impact of the semantic coherence module under different modality configurations within both the MoqaGPT and SKURG frameworks. As shown in Table 1, two-stage filtering consistently yields small but measurable improvements across F1 and EM metrics. The improvements are more notable in the multi-modality setting, where the integration of heterogeneous evidence sources increases the likelihood of semantic noise. This suggests that coherence filtering may help stabilize generation performance in such conditions.

In MoqaGPT, the performance gain from coherence filtering is most evident when multiple modalities are involved. With Euclidean similarity and full two-stage refinement, F1 improves from 29.61 to 31.46, and EM from 25.33 to 26.67. These gains, while modest in absolute terms, reflect improved evidence alignment under cross-modal reasoning (Figure 3).

Meanwhile, the baseline SKURG model achieved an overall F1/EM score of 61.29/58.80. When using only query-level CLIP similarity for evidence ranking, performance slightly dropped (-0.33 in F1, −0.37 in EM), suggesting the need for additional coherence filtering. The embedding-based coherence-aware variant produced results nearly identical to the baseline, showing minimal variation across all modality groups. Notably, the graph-based coherence strategy led to consistent performance improvements across all settings, improving overall F1 and EM by +0.29 and +0.31 respectively. These findings align with those observed in the MoqaGPT setting, reinforcing the effectiveness of coherence-aware filtering regardless of the underlying retrieval-generation architecture. While the integration of our plug-and-play module itself contributed to performance gains, strategies like the graph-based variant—which more closely align with the original model's reasoning paradigm—proved more effective at preserving semantic coherence among retrieved evidences.

4.3. Case Study

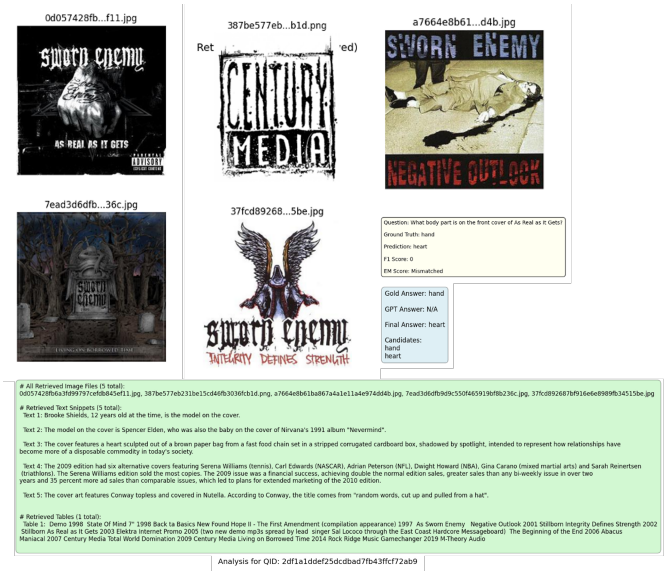

Figure 4. Visual comparison of evidence filtering strategies across modalities.

MoqaGPT To illustrate the effect of our coherence-aware evidence filtering module, we present a representative example from the MMQA dev set. The question requires synthesizing evidence from multiple modalities to generate an accurate answer.

Baseline (No Filtering). In the original pipeline, the retrieved evidence includes several partially relevant or off-topic items. These items exhibit inconsistencies in content and modality focus, leading to distractive or conflicting information. As a result, the generated answer is vague or factually incorrect. For example, as illustrated in Figure 4, the question *"What body part is on the front cover of As Real as It Gets?"*
with ground truth **hand** yields the following unfiltered evidence set:

$$\{I_i\}_{i=1}^{5}, \qquad \{T_j\}_{j=1}^{5}, \qquad \{\text{Tab}_1\}$$

where

- I_1 is the actual front-cover image of *As Real as It Gets*,
- $I_2 - I_5$ are other album covers or logos (off-topic),
- $T_1 - T_5$ are text snippets about unrelated album art or media facts (off-topic), and
- Tab_1 is table listing release dates (partially related by album title but containing no body-part information).

In particular, snippet T_3 contains the pharase "... *the cover features a heart sculpted* ...," which introduces a spurious cue. This misleading token "heart" correlates strongly with the model's incorrect prediction $\hat{y} = \texttt{heart}$, despite the true answer being "hand".

With Coherence Filtering. After applying our two-stage filtering module, the selected evidence set becomes noticeably more focused and semantically consistent. The retrieved text and image content reinforce each other, and redundant or contradictory information is removed. This leads to a more precise and contextually grounded answer.

SKURG We also present a representative failure case of SKURG where our coherence-aware module successfully corrects the prediction. In Figure 5, the question asks: "Which Title(s), in Filmography of Ben Piazza, has the left half of a woman's face on its poster?" Given the multimodal context, SKURG incorrectly selects *The Hanging Tree*, which also stars Ben Piazza and features a woman's portrait in partial profile, yet is unrelated to the question's implied subject. In contrast, our model correctly identifies *Tell Me That You Love Me, Junie Moon*, a film in which Liza Minnelli plays a woman with facial disfigurements. The movie's promotional materials prominently depict the left half of her face, aligning directly with the question. Our model's success is attributed to effective pruning based on cross-evidence coherence. In the constructed multimodal knowledge graph, we identify shared entity hubs (e.g.,Junie Moon, Liza Minnelli, and Facial disfigurement) which form a dense subgraph around Image Evidence 4 and the relevant texts. In contrast, Image Evidence 1 and its linked table evidence remain structurally isolated. This particular case shows that our module removes this low-coherence cluster, allowing the generator to focus on semantically aligned signals.

5. Related Works

5.1. Multimodal Visual Question Answering (VQA)

Visual Question Answering (VQA) refers to the task of answering natural language questions based on visual inputs such as images or videos. Recent advances in VQA have expanded beyond pure vision-language reasoning to incorporate retrieval-augmented or grounded generation. Early multimodal VQA models relied on parametric learning with paired image-question datasets, but recent work incresingly integrates external knowledge through retrieved evidence. Representative benchmark datasets such as MULTIMODALQA [5] and WebQA [3] provide diverse, multi-hop questions that require both textual and visual understanding across multiple sources.

5.2. Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) frameworks combine non-parametric retrieval with genera-

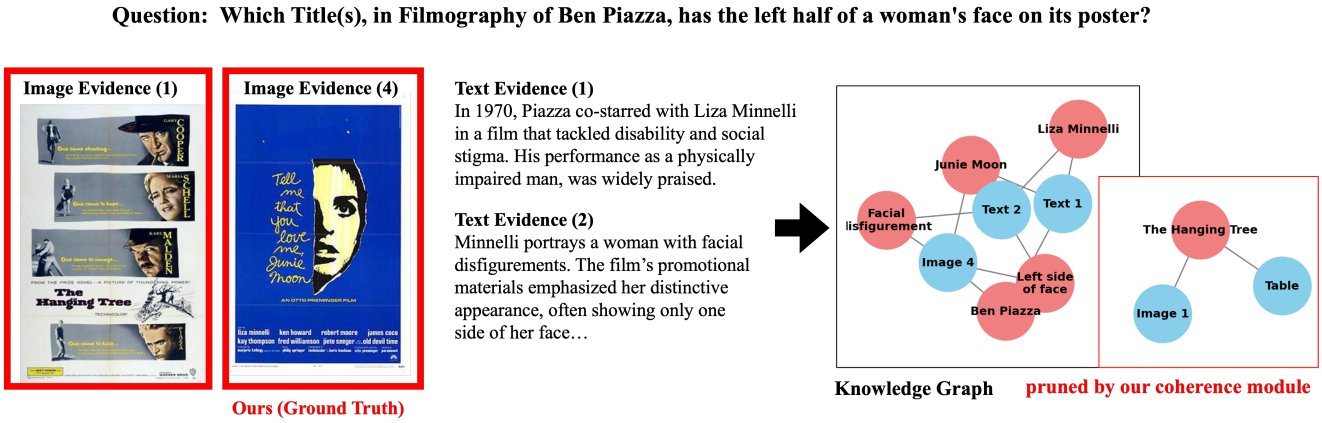

Figure 5. Qualitative example where SKURG fails while our coherence module succeeds. Our method selects semantically consistent evidence (Text 1-2, Image 4), leading to the correct answer, while SKURG is misled by visually similar but unrelated content.

tive models, enabling open-domain or visual question answering by incorporating relevant external knowledge at inference time. Early works such as RAG [6] focused on purely textual retrieval, while later models like MuRAG [4], M2RAG [4], and FilterRAG [12] extended this framework to more complex multi-hop or noisy retrieval settings.

In the multimodal domain, models such as MAVEx [14], SKURG [15], RAMQA [1], and MoqaGPT [16] incorporate cross-modal retrieval (e.g., from web images or captions) to enhance answer grounding. However, these systems predominantly focus on retrieving and encoding each evidence item independently, often relying on dense retrieval or pointer-based decoding, without verifying semantic consistency across multiple retrieved elements.

While FilterRAG [12] attempts to mitigate retrieval noise through learned relevance estimation, it requires retriever modification and task-specific tuning. In contrast, our work introduces a retriever-agnostic coherence filtering module that can be seamlessly inserted into both modular (e.g., MOQAGPT) and unified (e.g., SKURG) pipelines. This plug-and-play design improves reasoning reliability by filtering out inconsistent or contradictory evidence prior to generation.

6. Conclusion

Retrieval-augmented VQA pipelines typically rank evidence only by query relevance; as a result, mutually inconsistent, redundant, or contradictory items often reach the generator and increase hallucination risk. To mitigate this issue we proposed a *lightweight, plug-and-play two-stage coherence module*. Stage 1 performs *modality-wise relevance filtering*, selecting the top-k items per modality by query-evidence similarity. Stage 2 then conducts *cross-modal coherence refinement*, scoring the pooled items by average pairwise similarity and retaining the most self-consistent subset. Because the module operates on frozen embeddings, it can be inserted between retrieval and generation in existing pipelines such as MoqaGPT and SKURG. Experiments on the MMQA benchmark show *small but consistent* improvements.

This study (i) formalises the inter-evidence inconsistency problem in multimodal RAG-VQA, (ii) offers a retriever-agnostic two-stage filtering solution, and (iii) demonstrates its seamless integration into off-the-shelf systems with measurable accuracy and reliability benefits.

Future work will extend coherence filtering to video-based QA, explore richer cross-evidence metrics such as scene-graph overlap and temporal alignment, and investigate end-to-end training that jointly optimises retrieval, coherence enforcement, and answer generation within a unified framework.

References

[1] Yang Bai, Christan Earl Grant, and Daisy Zhe Wang. Ramqa: A unified framework for retrieval-augmented multi-modal question answering. *arXiv preprint arXiv:2501.13297*, 2025. 9

[2] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla and... Dhariwal, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. GPT-3 paper. 6

[3] Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16495–16504, 2022. 8

[4] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William W Cohen. Murag: Multimodal retrieval-augmented generator for open question answering over images and text. *arXiv preprint arXiv:2210.02928*, 2022. 2, 9

[5] Deepak Gupta, Surabhi Kumari, Asif Ekbal, and Pushpak Bhattacharyya. Mmqa: A multi-domain multi-lingual question-answering framework for english and hindi. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018. 8

[6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020. 9

[7] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 2

[8] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Proceedings of Machine Learning Research*, 202:19730–19742, 2023. ViT-G encoder + T5 decoder VLM. 6

[9] OpenAI. text-embedding-ada-002: Openai embedding model. `https://openai.com`, 2023. Table retrieval. 6

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2, 6

[11] Nils Reimers and Iryna Gurevych. Making monolingual sentence embeddings multilingual using knowledge distillation. *EMNLP*, 2020. all-mpnet-base-v2. 6

[12] SM Sarwar. Filterrag: Zero-shot informed retrieval-augmented generation to mitigate hallucinations in vqa. *arXiv preprint arXiv:2502.18536*, 2025. 9

[13] Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. Multimodalqa: Complex question answering over text, tables and images. *arXiv preprint arXiv:2104.06039*, 2021. 3

[14] Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. Multi-modal answer validation for knowledge-based vqa. In *Proceedings of the AAAI conference on artificial intelligence*, pages 2712–2721, 2022. 2, 9

[15] Qian Yang, Qian Chen, Wen Wang, Baotian Hu, and Min Zhang. Enhancing multi-modal multi-hop question answering via structured knowledge and unified retrieval-generation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5223–5234, 2023. 3, 9

[16] Le Zhang, Yihong Wu, Fengran Mo, Jian-Yun Nie, and Aishwarya Agrawal. Moqagpt: Zero-shot multi-modal open-domain question answering with large language model. *arXiv preprint arXiv:2310.13265*, 2023. 3, 9