EDMA: Entropy-Driven Multimodal Answering

Emanuele Mezzi
Vrije Universiteit Amsterdam
e.mezzi@vu.nl

Gertjan J. Burghouts
TNO
gertjan.burghouts@tno.nl

Fabio Massacci
Vrije Universiteit Amsterdam
University of Trento
fabio.massacci@ieee.org

Mengyuan Zhang
Vrije Universiteit Amsterdam
m.zhang@vu.nl

Abstract

Multimodal question answering (MMQA) is based on integrating heterogeneous data sources, selectively leveraging relevant modalities, and ignoring distractors. Existing approaches based on Multimodal Large Language Models (MLLMs) improve performance by splitting the task into intermediate steps, but do not quantify how each modality contributes to reaching the final answer. We argue that quantifying the effect that each modality has on the answering process improves process traceability and final performance. Thus, we propose Entropy-Driven Multimodal Answering (EDMA), which uses logical entropy to formalise multimodal QA as an iterative process of entropy reduction and to quantify the contribution of each modality in distinguishing correct from incorrect candidate answers. This enables the construction of *quantifiable answering trajectories*, where each step is associated with a measurable entropy reduction. EDMA introduces (i) modality-conditioned partitioning to estimate unimodal entropy reduction, (ii) entropy-driven multimodal fusion to capture complementary information across modalities, and (iii) entropy-based selection of the answering trajectory that minimises logical entropy. Evaluated across standard MMQA benchmarks, EDMA outperforms competitive prompting baselines, with average gains of 12.8 F1 and 4.6 LLM-J points over Planning Prompting and 18.6 F1 and 8.3 LLM-J points over Chain-of-Thought. Moreover, EDMA outperforms self-consistency baselines and is competitive with the multi-agent system baseline. Additionally, we show that higher entropy correlates with more false positives and, through controlled trajectory selection, demonstrate that logical entropy serves as an effective selector for correct reasoning paths, validating entropy level as a reliable proxy for performance. Our code is available at EDMA.

1 Introduction

Multimodal question answering (MMQA) requires the integration of heterogeneous data sources such as text, images, and tables [1, 2, 3]. For example, answering the question *"What is the state in the Contour Airlines city destinations, that has a tall metal arch on a river?"* requires identifying visual evidence of a metal arch in images while simultaneously retrieving structured information from tables to determine the destinations of the airline, and combining the evidence.

Recent Multimodal Large Language Models (MLLMs) have shown strong performance on such tasks by leveraging unified representations across modalities [4, 5], and are currently being employed to solve domain-specific problems [6, 7, 8, 9, 10]. Several techniques have been proposed to

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

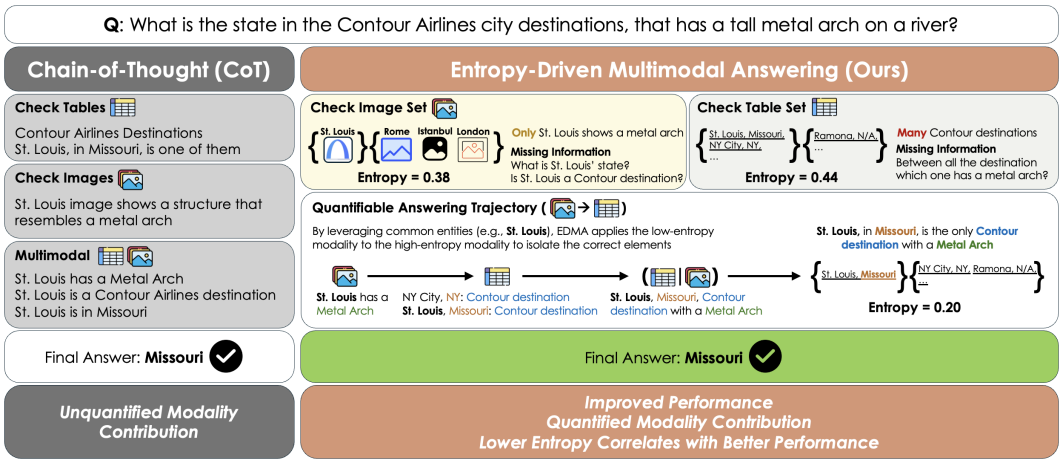


Figure 1: Comparison between CoT and EDMA. While they both structure QA in intermediate steps, EDMA generates quantifiable answering trajectories, allowing for improved performance and quantified modality contribution.

improve multimodal inference, most notably Chain-of-Thought (CoT) prompting [11, 12, 13], which decomposes complex tasks into intermediate steps. Building on this paradigm, extensions such as compositional prompting [14], knowledge-augmented reasoning [15], PEAR-CoT [16], frameworks such as Chain-of-Action (CoA) [17] and Visualization-of-Thought (VoT) [18] have been proposed to address limitations in compositionality, grounding, and spatial awareness. To further improve the inference process, recent work has incorporated uncertainty quantification, enabling models to adaptively seek information or refine intermediate steps [19, 20, 21]. However, while these approaches improve robustness by structuring the inference process in intermediate steps, augmenting contextual information, and guaranteeing higher robustness, they do not provide explicit stepwise quantification of how individual modalities and their fusion contribute to reducing uncertainty over the space of candidate answers. As a result, it remains unclear how different modalities contribute to the inference process, limiting interpretability and controllability in multimodal settings.

To address these limitations, we propose Entropy-Driven Multimodal Answering (EDMA), a methodology that models QA as a process of iterative entropy reduction over candidate answers. As Figure 1 shows, EDMA quantifies the impact of incorporating a specific modality and fusing modalities by measuring how this refines the space of possible answers. Compared to CoT, which splits the answering process in steps without quantifying the contribution of each modality, EDMA builds *quantifiable answering trajectories*, where each step and each modality is associated with a measurable contribution in the reduction of entropy. At the core of EDMA lies logical entropy [22], which measures distinctions between possible outcomes. This formulation is particularly well-suited for multimodal QA, where the objective is to progressively separate correct from incorrect candidate answers based on evidence from different modalities. By framing answering as partition refinement, EDMA treats each piece of evidence as a condition that splits the current answer space into two subsets of answers: those that satisfy the evidence and those that do not. This systematic partitioning provides a natural and interpretable mechanism for tracking exactly how information from each modality contributes to narrowing down the final prediction.

We evaluate EDMA on challenging multimodal QA benchmarks [1, 23] and show that it outperforms best-practice prompting baselines and, on average, state-of-the-art multi-agent systems, while providing traceable uncertainty estimates that guide modality selection and the answering process. Our contributions are as follows:

- EDMA, a methodology that formalizes multimodal QA as modality-conditioned entropy minimization over the space of alternatives.
- A modality-conditioned entropy reduction objective that enables explicit measurement of each modality's contribution.
- We establish the relation between entropy minimization and lower false positives (FPs) levels, thus validating entropy minimization as a proxy for performance.

2 Entropy-Driven Multimodal Answering (EDMA)

2.1 Problem formulation

The problem setting involves one question Q, the set of modalities available to answer the question denoted by M and, for each $m \in M$, the elements associated with that modality denoted by Ω_m, and one Answerer denoted by LLM. To answer each question, the LLM must find the modalities of interest, find the elements within those modalities that contain the necessary information to answer the question, and then combine the information to provide a cross-modal answer, denoted by a^*.

2.2 Overview of EDMA

Figure 2 shows that EDMA is composed of four stages: (a) prediction of the modalities to be used and extraction of answer requirements from the question, (b) generation of unimodal partitions and calculation of unimodal logical entropy, (c) generation of the multimodal partitions with paired multimodal logical entropy, and (d) answer generation by selecting the answering trajectory that minimized the most the logical entropy. The process allows to reduce at each step the number of elements which represent the possible alternatives by checking which ones fit the requirements extracted from the question, returning an answer based on the answering trajectory that minimised the most the logical entropy. In Algorithm 1 in Appendix B, we describe EDMA's answering process.

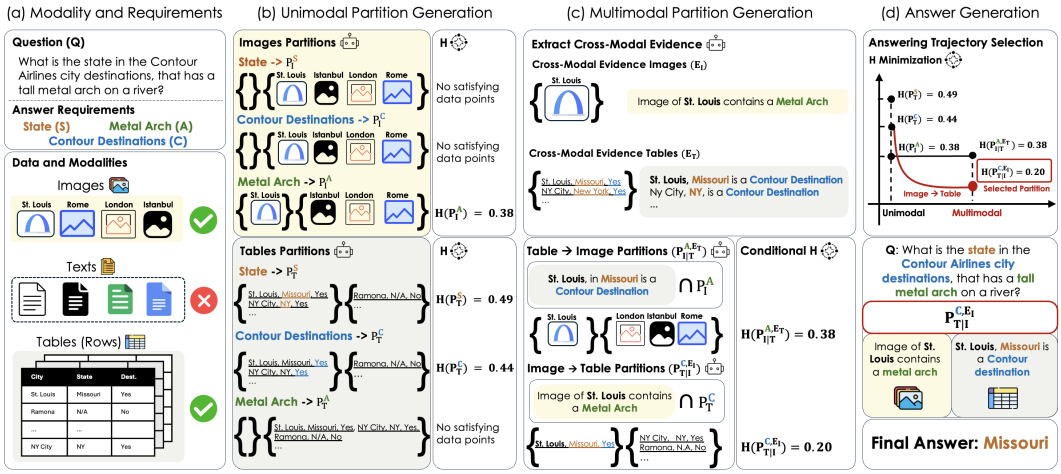


Figure 2: EDMA is composed of four stages: (a) modality prediction and extraction of answer requirements, (b) generation of unimodal partitions and calculation of unimodal logical entropy, (c) generation of multimodal partitions and calculation of multimodal logical entropy, (d) selection of the answering trajectory which minimises the logical entropy and generation of the answer.

2.3 Extraction of Answer Requirements

Supposing we want to answer the question *"What is the state in the Contour Airlines city destinations, that has a tall metal arch on a river?"*, EDMA begins by extracting the set of answer requirements, representing the characteristics that the final answer a^* must satisfy. We denote the answer requirements as R. As shown in Figure 2, the LLM generates R by identifying the key elements of the question. In this specific case, the final answer has to: (i) be a *State*, (ii) be a *destination of Contour Airlines*, and (iii) be characterized by a *Metal Arch*. We denote the extraction of R as $R = LLM(Prompt_R, Q)$, where the LLM receives in input the user instructions $Prompt_R$ and the question Q. In this case, $R = \{S, C, A\}$, where we use S to point to State, C for Contour Destination, and A for Metal Arch.

2.4 Prediction of Starting Modalities

EDMA predicts the subset of modalities $M_s \subseteq M$ representing the modalities with which to start the analysis. This phase does not exclude the possibility of incorporating additional modalities later in

the answering process. The LLM receives the input, the user instructions $Prompt_{M_s}$, the question Q, and for each modality m, the corresponding set of elements Ω_m. It is then tasked with predicting the combination of modalities that is most likely to contain the elements needed to reach the correct answer. This operation can be summarized as $M_s = LLM(Prompt_{M_s}, Q, \{\Omega_m, m \in M\})$. The modalities M_s are then added to M_a, which is the list that keeps track of the modalities used for the analysis. In Figure 2, the predicted modalities are *Table (T)* and *Image (I)*.

2.5 Entropy-Driven Answer Search

To identify the correct answer in a multimodal setting, EDMA models inference as an entropy-driven search over the space of candidate answers. It quantifies how individual modalities and their interactions reduce uncertainty over the answer space. To this end, EDMA first induces partitions over the answer space using unimodal evidence and then refines these partitions through sequential cross-modal conditioning. This process enables systematic measurement of entropy reduction from both unimodal and multimodal evidence.

Quantifying Uncertainty of Answer Partitions We adopt logical entropy [22], which measures uncertainty in terms of distinctions induced by a partition of a candidate set. Let S denote a finite set of candidate elements and let $P_S = \{B_1, \ldots, B_n\}$ be a partition of S into disjoint non-empty blocks. To ensure compatibility across open-source and closed-source models without requiring calibrated probability estimates from the underlying model, we assume a uniform distribution over S, such that each block has probability mass $p_i = \frac{|B_i|}{|S|}$. Logical entropy is defined as $H(P_S) = 1 - \sum_{i=1}^{n} p_i^2$, and can be interpreted as the probability that two independently sampled elements from S belong to different blocks.

For example, consider $|S| = 10$ and a binary partition $P_S = \{S_r, S_{\neg r}\}$ induced by a requirement r. The set S_r contains elements whose information overlaps with r, while $S_{\neg r}$ contains elements that do not. If $|S_r| = 5$ and $|S_{\neg r}| = 5$, then $H(P_S) = 0.50$, corresponding to maximal entropy for a binary partition under a uniform prior. If instead $|S_r| = 1$ and $|S_{\neg r}| = 9$, then $H(P_S) = 1 - (0.1^2 + 0.9^2) = 0.18$, reflecting reduced uncertainty due to the increased concentration of mass.

In our multimodal QA setting, the answering process is modeled as a sequence of partition refinements over S, where information extracted from the elements of each modality introduces distinctions that separate relevant from irrelevant candidates. We restrict attention to valid modality-induced refinements where the predicate-satisfying subset is non-empty, $S_r \neq \emptyset$, ensuring that entropy reduction is grounded in non-vacuous evidence.

Logical entropy thus quantifies the effectiveness of each modality in refining the answer space, allowing to prioritize modalities that maximize entropy reduction, guiding the system toward increasingly discriminative evidence.

Generation of Unimodal Partitions EDMA analyses the elements Ω_m of each modality $m \in M_a$. For each element $\omega \in \Omega_m$, the LLM evaluates, for each answer requirement $r \in R$, whether ω contains information overlapping with r. This operation can be formally expressed as $LLM(Prompt_{Check}, \omega, r)$. Based on this evaluation, EDMA constructs, for each set Ω_m and requirement r, a partition $P_m^r = \{\Omega_m^r, \Omega_m^{\neg r}\}$, where Ω_m^r contains elements whose information overlaps with r, and $\Omega_m^{\neg r}$ contains elements that do not. Under the assumption of uniform distribution over elements in Ω_m, the logical entropy of P_m^r is:

$$H(P_m^r) = 1 - \left(\left(\frac{|\Omega_m^r|}{|\Omega_m|} \right)^2 + \left(\frac{|\Omega_m^{\neg r}|}{|\Omega_m|} \right)^2 \right) \tag{1}$$

This formulation expresses logical entropy as a function of partition cardinalities, enabling estimation without relying on confidence estimates from the LLM. Logical entropy measures the distinction between elements that satisfy a given condition and those that do not. It attains its global minimum of 0 when either $\Omega_m^r = \emptyset$ or $\Omega_m^{\neg r} = \emptyset$. Otherwise, it is strictly positive and is minimized when one subset is highly concentrated (i.e., when a single candidate is clearly distinguished).

In Figure 2, unimodal partitions illustrate how different modalities contribute to distinguishing relevant from irrelevant candidates. The image-based partition P_I^A yields a relatively low entropy

(0.38), indicating that the visual modality already provides a strong separation by isolating the candidate satisfying the requirement A. In contrast, the table-based partitions exhibit higher entropy values (e.g., 0.49 for the S requirement), reflecting a weaker discriminative effect when considered in isolation. Since lower entropy corresponds to sharper distinctions among candidates, this comparison indicates that the image modality is more informative at this stage for narrowing the answer space. EDMA leverages this signal to prioritize modalities that induce stronger reductions in uncertainty, guiding the answering process toward partitions that more effectively constrain the set of plausible answers.

Generation of Multimodal Partitions Once unimodal partitions P_I^A, P_T^S, P_T^C are constructed, EDMA proceeds to generate multimodal partitions. This step leverages information extracted from multiple modalities to refine candidate elements, progressively reducing logical entropy, with the aim of isolating the correct candidate(s), filtering out the alternatives that are inconsistent with cross-modal evidence, which is information extracted from one modality used to reduce the answer space in another modality. In a multimodal setting, the final answer typically resides in a specific modality, referred to as the *target modality*, denoted by m_t. However, the target modality is not known in advance. To identify it, EDMA systematically treats each modality in turn as a candidate target modality m_t, while the remaining modalities $M_c = M_a \setminus \{m_t\}$ serve as conditioning modalities.

For each candidate m_t, EDMA applies information from all conditioning modalities sequentially. For each conditioning modality $m \in M_c$ and the subset $\Omega_m^{r'}$ containing elements that satisfy a requirement $r' \in R$, EDMA extracts the cross-modal evidence set:

$$E_m = \left\{ e \mid e = LLM(Prompt_{Extract}, Q, \omega),\ \omega \in \Omega_m^{r'} \right\} \tag{2}$$

Each piece of evidence represents information from the conditioning modality m and has the same function as the answer requirements. EDMA applies the elements of this set sequentially to iteratively refine the candidates in the target modality m_t that satisfy requirements $r \in R$, where $r \neq r'$. This allows evidence from one modality (e.g., visual) that satisfies one requirement to be used to filter candidates in another modality (e.g., tabular) that satisfy a distinct requirement.

Specifically, for a given requirement $r \in R$, the target subset $\Omega_{m_t}^r$ is updated sequentially for each conditioning modality. At each step, EDMA generates a partition $P_{m_t|m}^{r,E_m} = \{\Omega_{m_t}^{r,E_m}, \Omega_{m_t}^{r,\neg E_m}\}$ of the elements of modality m_t consistent with requirement r, where $\Omega_{m_t}^{r,E_m}$ contains the elements that are also consistent with the evidence E_m from the current conditioning modality, and $\Omega_{m_t}^{r,\neg E_m}$ contains those that are not. Only the subset $\Omega_{m_t}^{r,E_m}$ is carried forward to the next iteration, while $\Omega_{m_t}^{r,\neg E_m}$ is excluded from further conditioning, as it cannot lead to the correct answer. This sequential conditioning progressively narrows the answer space of m_t, increasing the distinction among the remaining candidates, thus decreasing logical entropy.

By evaluating the multimodal logical entropy after each step, EDMA identifies which m_t ultimately yields the partition with the lowest entropy, where distinctions among candidates are maximized, and uncertainty is minimized. Conditional or multimodal logical entropy is computed analogously to the unimodal case:

$$H\left(P_{m_t|m}^{r,E_m}\right) = 1 - \left(\left(\frac{|\Omega_{m_t}^{r,E_m}|}{|\Omega_{m_t}|} \right)^2 + \left(\frac{|\Omega_{m_t}^{r,\neg E_m}|}{|\Omega_{m_t}|} \right)^2 \right) \tag{3}$$

In the example in Figure 2, cross-modal conditioning yields asymmetric effects depending on the direction of information flow. When table-derived constraints (E_T) are applied to the image modality (P_I^A), entropy remains unchanged (0.38), indicating limited additional discriminative power beyond visual cues. In contrast, conditioning the table modality (P_T^C) using visual evidence (E_I) significantly reduces entropy ($0.44 \rightarrow 0.20$), showing that the image provides high-value filtering information that effectively disambiguates table entries, and showing that EDMA does not treat modalities symmetrically, but instead identifies directional entropy reduction, where certain modalities act as stronger selectors of relevant candidates if the derived information is applied in the right order.

2.6 Answer Generation

Trajectory Selection via Entropy Minimization The systematic generation of unimodal and multimodal partitions effectively maps a set of *quantifiable answering trajectories* ($\mathcal{T}$), representing paths in the search process. Each trajectory provides a quantifiable history of how information is fused and how logical entropy changed after the modality fusion. This allows for a modality-level attribution of the logical entropy variation. A trajectory may remain within a unimodal setting ($m \to m$) or bridge modalities to form a multimodal trajectory ($m \to m_t$), where the first modality conditions the second ($m_t|m$). We denote $\Delta H = \frac{H(P^r_{m_t}) - H(P^{r,E_m}_{m_t|m})}{H(P^r_{m_t})}$, the relative change between unimodal entropy and multimodal entropy for modality m_t. If $\Delta H > 0$, modality fusion increases the distinguishability among candidates, thereby reducing uncertainty. If $\Delta H \leq 0$, the conditional modality provides redundant or non-discriminative information for that specific answering path.

In Figure 2, EDMA identifies four answering trajectories, of which two are unimodal and two are multimodal. The unimodal trajectories are represented by P^A_I (initial $H = 0.38$) and P^C_T (initial $H = 0.44$). The Table-Image Trajectory starts with partition P^A_I and ends with partition $P^{A,E_T}_{I|T}$, where table data regarding destinations conditions the image set, resulting in a final entropy of $H = 0.38$ with $\Delta H = 0\%$. In contrast, the Image-Table Trajectory, where visual evidence of the *Metal Arch* conditions the table rows, starts with partition P^C_T and ends with partition $P^{C,E_I}_{T|I}$, reducing logical entropy from $H = 0.44$ to $H = 0.20$, achieving a significant entropy decrease of $\Delta H = 54\%$, a reduction attributed to the image modality's influence on the table modality.

As EDMA prioritizes the trajectory whose final partition yields the lowest logical entropy (t^*), it selects the Image–Table Trajectory, isolating the table entry corresponding to the city satisfying requirement C, which is also associated with an image from which EDMA extracts visual evidence of the arch. The multimodal partition at the endpoint of t^* is chosen as the basis for the final answer.

Final Inference Once the optimal trajectory, modality, and partition are selected, the LLM receives the question Q, the satisfied requirements, the cross-modal evidence E^* (e.g., the context of the arch), and the relevant elements Ω^* that satisfy both the requirements and the cross-modal evidence (e.g., the specific row for St. Louis). Based on this information, the system returns the final answer a^* (e.g., *Missouri*). If the LLM cannot produce an answer (i.e., $a^* =$ None), EDMA selects a new set of modalities $M_s \subseteq M \setminus M_a$, where the newly selected modalities are not contained in the already analysed modality set. EDMA then repeats the process: generating new partitions, recomputing entropy deltas, and attempting to produce the answer with the newly acquired information. The iteration continues until a valid answer is obtained or all modalities are exhausted.

Crucially, by constructing and selecting among quantifiable answering trajectories, EDMA provides a principled mechanism to guide cross-modal inference. Even when SOTA approaches arrive at the same final answer, EDMA makes explicit how that answer is obtained, enabling quantifiable modality-level attribution.

3 Experiments

3.1 Experimental Setup

We evaluate our methodology across a diverse set of SOTA LLMs, including both closed source models, such as GPT-5.2 [24], Kimi-K2.5 [25], Nova-2-Lite-v1 (Nova-2-Lite) [26], and Claude-Sonnet-4.6 (Claude-4.6) [27], and open source models, such as Qwen3-VL-235B-A22B (Qwen3) [28], and Nemotron-Nano-12B-v2 (Nemotron) [29]. Appendix E reports how to access models and the resources needed to run them, while Appendix K reports the prompts to implement EDMA's operations.

Baselines We compare our approach against SOTA multimodal QA system MAMMQA [30] and against several common best-practice prompting strategies: Direct Prompting (DP), where the model generates an answer without explicit intermediate steps, Chain-of-Thought (CoT) [11], which encourages step-by-step generation of intermediate steps within a single generation, and Planning Prompting (PP) [31], where the model first generates a plan and subsequently executes it, CoT-SC

[32] and PP-SC which improve the greedy versions of CoT and PP by generating multiple reasoning chains and returning the majority-vote final answer.

Datasets We evaluate our methodology on MultiModalQA [1] and ManyModalQA [23] datasets which remain, to the best of our knowledge, among the most complex benchmarks for multimodal QA, requiring cross-modal integration across text, tables, and images. They both require deep joint inference over diverse data formats to arrive at a final answer, and thus provide a rigorous testbed for evaluating a model's ability to integrate and reason over heterogeneous information sources.

Evaluation Metrics We evaluate all methods using Exact Match (EM) and F1 score, following standard practice in multimodal QA [1, 33, 34]. EM is a strict binary metric that assigns a score of 1 if the predicted answer exactly matches the ground-truth string and 0 otherwise. The F1 score measures token-level overlap between the predicted and reference answers, balancing precision and recall. Because EM and F1 can be overly penalizing for semantically correct open-ended generations, in our evaluation we also include an LLM-Judge (LLM-J) metric [35].

3.2 EDMA Performance Evaluation

Here we report the performance of EDMA across benchmarks. We first compare EDMA to standard prompting baselines (DP, CoT, and PP) and then we evaluate EDMA against budget-matched baselines.

3.2.1 Comparison between EDMA and Prompting Baselines

Table 1 summarises the performance of EDMA alongside standard prompting baselines across six representative MLLMs. On average, EDMA outperforms Direct Prompting (DP), Chain-of-Thought (CoT), and Planning Prompting (PP) on both datasets across all three metrics. On MultiModalQA, EDMA achieves an average score of 80.1 EM, 81.1 F1, and 79.3 LLM-J, outperforming the strongest prompting baseline (PP) by $+22.7$ EM, $+14.0$ F1, and $+8.5$ LLM-J points. On ManyModalQA, EDMA reaches 83.0 EM, 83.3 F1, and 89.1 LLM-J, improving over PP by $+23.0$ EM, $+11.5$ F1, and $+0.6$ LLM-J. Across both benchmarks, EDMA achieves average gains of $+22.9$ EM, $+12.8$ F1, and $+4.6$ LLM-J over the strongest prompting baseline. The smaller LLM-Judge gain on ManyModalQA (+0.6 vs +8.5 on MultiModalQA) stems from lower benchmark complexity, limiting EDMA's possible improvement space. Nevertheless, EDMA's strong leads in EM (+23.0) and F1 (+11.5) on ManyModalQA confirm that entropy-guided decomposition remains essential for precise entity extraction even on simpler queries.

Table 1: EDMA achieves the highest average EM, F1, and LLM-judge scores on MultiModalQA, ManyModalQA across all evaluated models, outperforming best-practice prompting strategies: Direct Prompting (DP), Chain-of-Thought (CoT), and Planning Prompting (PP).

Method	Model																				
	Claude-4.6			Kimi-K2.5			Qwen3			GPT-5.2			Nemotron			Nova-2-Lite			Avg		
	EM	F1	LLM-J	EM	F1	LLM-J	EM	F1	LLM-J	EM	F1	LLM-J	EM	F1	LLM-J	EM	F1	LLM-J	EM	F1	LLM-J
MultiModalQA [1]																					
EDMA	**82.8**	**83.5**	**88.9**	**80.4**	**81.0**	**83.2**	**79.4**	**80.9**	**82.5**	**80.0**	**80.3**	**80.5**	**79.0**	**80.5**	**75.1**	**79.0**	**80.4**	**65.9**	**80.1**	**81.1**	**79.3**
PP	61.4	73.1	85.6	64.0	74.0	82.0	60.6	69.4	75.4	56.6	68.4	75.5	51.8	60.0	64.6	50.0	57.8	41.8	57.4	67.1	70.8
CoT	61.4	72.3	87.4	63.8	72.7	79.9	61.0	70.7	76.4	46.2	56.8	64.4	46.4	54.2	57.1	43.4	50.2	53.4	53.7	62.8	69.8
DP	60.8	70.5	82.7	39.4	47.2	52.1	60.6	68.9	74.3	58.2	68.8	75.7	42.2	50.0	52.3	51.8	59.6	64.6	52.2	60.8	66.9
ManyModalQA [23]																					
EDMA	**82.0**	**82.4**	90.0	**86.0**	**86.6**	**94.5**	**84.0**	**84.0**	**90.0**	**88.0**	**88.5**	**88.0**	**78.0**	**78.0**	**84.5**	**80.0**	**80.0**	**87.5**	**83.0**	**83.3**	**89.1**
PP	56.0	70.2	90.0	62.0	74.4	93.5	64.0	75.8	89.5	48.0	63.4	86.5	64.0	71.3	84.0	66.0	75.7	**87.5**	60.0	71.8	88.5
DP	70.0	81.5	**93.0**	70.0	80.1	93.5	62.0	71.2	86.0	56.0	67.8	87.5	48.0	56.7	74.5	58.0	66.7	80.5	60.7	70.7	85.8
CoT	56.0	68.2	88.0	62.0	70.4	83.0	64.0	73.3	86.5	42.0	56.2	85.0	40.0	50.2	69.0	60.0	67.9	81.0	54.0	64.4	82.1

In Appendix F and Appendix G, we respectively report detailed analyses regarding EDMA's intermediate operation accuracy and statistical variance for EDMA and baselines results.

3.2.2 Comparison between EDMA and Budget-Matched Baselines

To determine whether EDMA's gains result merely from increased inference computation, we evaluate EDMA on MultiModalQA against budget-matched self-consistency baselines (CoT-SC and PP-SC with $N = 6$ samples) and the multi-agent framework MAMMQA. As shown in Table 5 (Appendix E), EDMA is approximately $6\times$ computationally more expensive than single-pass CoT

and PP, making $N = 6$ a direct budget-matched baseline. Furthermore, EDMA's computational footprint is comparable to MAMMQA, establishing it as an additional budget-matched baseline. We evaluate MAMMQA using exclusively LLM-J, as its verbose answer style would unfairly penalize it under exact-matching metrics like EM and F1.

Table 2: Comparison of EDMA with MAMMQA and self-consistency (SC) baselines ($N = 6$ samples) across EM, F1, and LLM-J metrics.

Method	Model														
	Claude-4.6			Kimi-K2.5			Nemotron			Nova-2-Lite			Avg		
	EM	F1	LLM-J	EM	F1	LLM-J	EM	F1	LLM-J	EM	F1	LLM-J	EM	F1	LLM-J
EDMA	**82.8**	**83.5**	**88.9**	**80.4**	**81.0**	83.2	**79.0**	**80.5**	**75.1**	**79.0**	**80.4**	**65.9**	**80.3**	**81.4**	**78.2**
MAMMQA	–	–	83.3	–	–	**88.4**	–	–	65.7	–	–	65.7	–	–	75.7
CoT-SC	58.5	72.0	84.3	58.5	69.6	76.4	42.5	52.7	61.0	46.5	55.6	63.2	51.5	62.5	71.2
PP-SC	56.5	69.9	82.5	60.5	73.7	80.8	52.5	61.6	60.2	55.5	67.3	59.6	56.3	68.1	70.8

As detailed in Table 2, the effect of self-consistency depends on model capacity. Under PP-SC, Nova-2-Lite gains $+5.5$ EM, $+9.5$ F1, and $+17.8$ LLM-J over single-pass PP, whereas Nemotron improves slightly on EM ($+0.7$) and F1 ($+1.6$) but drops -4.4 LLM-J. In contrast, the larger backbones degrade: Claude-4.6 drops -4.9 EM, -3.2 F1, and -3.1 LLM-J, and Kimi-K2.5 drops -3.5 EM, -0.3 F1, and -1.2 LLM-J. Under CoT-SC, Nova-2-Lite again benefits across all metrics ($+3.1$ EM, $+5.4$ F1, $+9.8$ LLM-J over single-pass CoT), while Nemotron gains $+3.9$ LLM-J but loses -3.9 EM and -1.5 F1. The larger backbones again degrade on all metrics: Claude-4.6 drops -2.9 EM, -0.3 F1, and -3.1 LLM-J, and Kimi-K2.5 drops -5.3 EM, -3.1 F1, and -3.5 LLM-J.

Across the four backbones, SC yields only modest average LLM-J gains over single-pass prompting ($+2.3$ for PP-SC and $+1.8$ for CoT-SC) and does not improve EM. In contrast, EDMA (78.2 LLM-J, 80.3 EM, 81.4 F1) outperforms CoT-SC (71.2 LLM-J) and PP-SC (70.8 LLM-J) by $+7.0$ and $+7.4$ LLM-J, respectively, and MAMMQA (75.7 LLM-J) by $+2.5$ LLM-J on average, although MAMMQA is stronger on Kimi-K2.5 (88.4 vs. 83.2 LLM-J). The importance of EDMA's core mechanisms is further supported by two ablation studies, which isolate the contributions of entropy minimization and cross-modal conditioning (Tables 10–11, Appendix H).

3.3 Qualitative Example

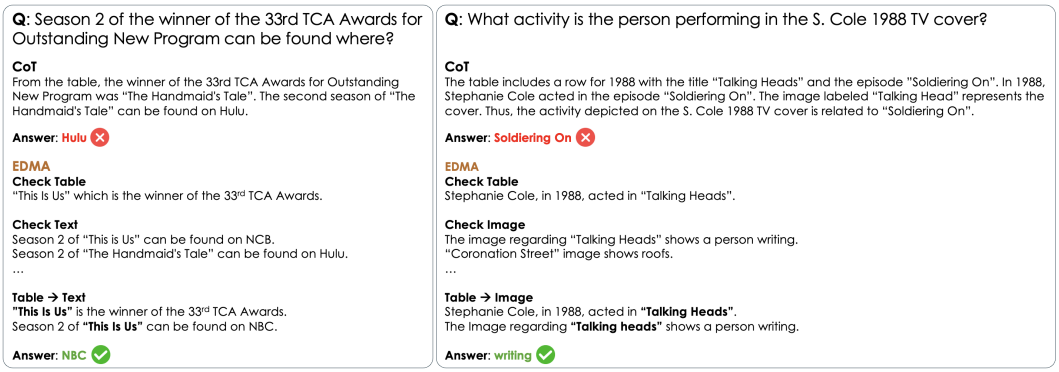


Figure 3: Qualitative comparison of CoT and EDMA. EDMA answers both questions correctly, isolating information in the unimodal setting before bridging modalities in the multimodal step, while CoT returns incorrect answers.

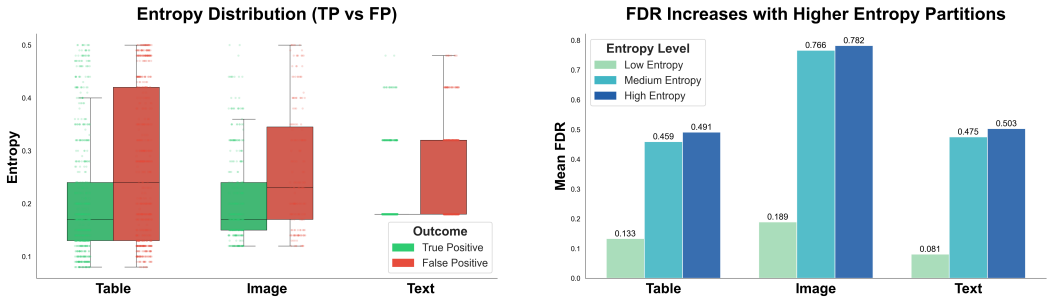
We attribute the performance gains achieved by EDMA to its staged reasoning process. Specifically, EDMA first decomposes the answer space into unimodal partitions before extracting explicit cross-modal evidence to construct combined multimodal partitions. This progressive refinement allows the model to isolate and verify relevant information within individual modalities prior to cross-modal integration and trajectory selection, which may improve the reliability of evidence aggregation and reduce ambiguity in cross-modal alignment. In contrast, standard prompting paradigms such as CoT and PP depend on the model capacity to implicitly manage decomposition and multimodal

integration within a single inference stream, which may limit their ability to consistently isolate and align evidence across modalities.

Figure 3 illustrates this contrast on two multi-hop examples. In the first, CoT misidentifies the winner of the 33rd TCA Award for Outstanding New Program. EDMA instead isolates the tabular context to identify "This Is Us" as the winner, and then queries the secondary modality to retrieve the correct broadcasting network (NBC). In the second example, CoT retrieves the tabular context (the series "Talking Heads") but its prediction is not grounded in the visual modality. EDMA conditions its visual query on the tabular entry and correctly aligns the image, identifying the depicted activity as writing.

3.4 False Positives Correlate with Higher Entropy

Here we analyse the relationship between entropy and prediction errors to assess the effectiveness of EDMA, examining how entropy correlates with the occurrence of FPs across target modalities. Figure 4a shows that across all modalities, FPs are consistently associated with higher entropy values compared to true positives (TPs). This indicates that incorrect predictions tend to occur in regions of higher uncertainty, where the candidate answer space remains insufficiently constrained. In contrast, correct predictions are more likely to arise when entropy is lower, reflecting a more decisive partitioning of the answer space. See Table 14 in Appendix I for data regarding statistical significance.



(a) Across all three target modalities, the mean entropy of FPs is higher than for TPs.

(b) Selecting higher-entropy trajectories is associated with a higher FDR.

Figure 4: Wrong answers are associated with higher entropy values, as shown by a correlational analysis (a) and a controlled trajectory-selection analysis (b).

3.5 Answering Trajectories with Lower Entropy Yield Better Answers

To further show the connection between entropy level and FPs, we perform a controlled intervention analysis to assess whether entropy directly impacts prediction performance. We generate three candidate answers per question by selecting answering trajectories corresponding to the lowest, median, and highest entropy levels. The low-entropy trajectory corresponds to the one selected by EDMA in the main experiments. This design keeps the input question fixed while varying the selected trajectory, and hence its entropy. We evaluate prediction reliability using the False Discovery Rate (FDR), defined as the proportion of FPs among all positive predictions. We compute FDR separately for each entropy group across modalities. Figure 4b shows a consistent trend: low-entropy trajectories yield the lowest FDR, while medium- and high-entropy trajectories exhibit progressively higher error rates. This pattern is consistent across all target modalities, indicating a strong association between higher logical entropy and reduced prediction reliability. By systematically varying the selected trajectory while keeping the input query fixed, we observe a robust degradation in performance as entropy increases. These results support the importance of entropy minimization in EDMA: lower-entropy trajectories are associated with fewer false positives and improved prediction performance. Tables 15–16 in Appendix I, report data regarding statistical significance.

4 Related work

MultimodalQA Multimodal question answering (MMQA) has been introduced as a challenging task requiring the integration of heterogeneous data sources such as text, tables, and images [1]. Subsequent work has focused on improving representation and integration across modalities, including unified architectures such as UniMMQA [2], Solar [33], retrieval-augmented approaches such as UniRAG [34], and conversational extensions like MMConvQA [36]. More recent approaches leverage LLMs and MLLMs to enhance multi-step inference capabilities, for instance by combining language and vision models [37], improving cross-modal alignment through similarity computation [38], or framing information synthesis as a multi-agent process, as in MAMMQA [30].

LLMs in MultimodalQA LLMs have been widely adopted to enhance multimodal QA, often through extensions of Chain-of-Thought (CoT) prompting. Prior work includes Chain-of-Action (CoA) [17], knowledge-augmented CoT (KAM-CoT) [15], duty-distinct CoT (DDCoT) [13], and frameworks for structured CoT such as UniCoTT [39]. These approaches improve inference by decomposing complex tasks into intermediate steps or incorporating external knowledge. Additional adaptations target specific modalities, such as AVQA-CoT for audio-visual QA [40] and methods that leverage textual signals to guide multimodal understanding [41]. Despite these advances, existing CoT-based approaches do not provide explicit mechanisms to quantify the extent to which each modality contributes to the final prediction.

Uncertainty as Guidance Recent work has explored incorporating uncertainty quantification into LLM reasoning to improve robustness and decision-making. Hu et al. [19] propose Uncertainty of Thoughts, which evaluates alternative planning paths based on their expected uncertainty reduction. Yin et al. [20] introduce uncertainty-aware adaptive guidance, an approach that relies on uncertainty signals from the LLM to guide the reasoning process. Liang et al. [21] propose introspective planning to align model uncertainty with task ambiguity, while Bayat et al. [42] leverage uncertainty estimates to guide answer selection. While these approaches treat uncertainty as a global signal for improving reasoning, they do not provide fine-grained, modality-level attribution of uncertainty reduction, limiting their effectiveness in multimodal QA settings.

5 Limitations

EDMA's worst-case complexity is $O(n^3)$ when modalities, requirements, and elements grow together. In practice this translates into many LLM calls per question, costing between 5 and $6.4\times$ more and taking between 20 and $46\times$ longer than the prompting baselines (Appendix E). Our budget-matched baselines match its monetary cost, but not its latency or number of calls, and a full accuracy–cost analysis is left to future work.

Logical entropy is computed under a uniform distribution over elements, which avoids calibrated probabilities but ignores differences in the relevance of individual elements.

Finally, our methodology does not explicitly address scenarios in which different modalities provide conflicting information. One possible extension would be to partition the answer space based on parallel interpretations of the available evidence, treating them as distinct answering trajectories, which would let the model explore potentially conflicting paths and evaluate their outcomes independently.

6 Conclusion

This paper introduces Entropy-Driven Multimodal Answering (EDMA), a novel methodology for multimodal QA based on logical entropy minimization and the construction of quantifiable answering trajectories.

By measuring how each step reduces logical entropy, EDMA quantifies the contribution of every modality, improving traceability. Our analyses show that false positives are associated with higher entropy, and that choosing the lowest-entropy trajectory tends to lead to better answers, making entropy a useful proxy for performance. Overall, on the same backbones, EDMA improves over the baselines it is compared against.

Acknowledgments and Disclosures

This work was partially supported by the *Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO)* under the KIC HEWSTI Project under grant no. KIC1.VE01.20.004.

The authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] 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.

[2] Haohao Luo, Ying Shen, and Yang Deng. Unifying text, tables, and images for multimodal question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9355–9367, 2023.

[3] Negar Foroutan, Angelika Romanou, Matin Ansaripour, Julian Martin Eisenschlos, Karl Aberer, and Rémi Lebret. Wikimixqa: a multimodal benchmark for question answering over tables and charts. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24941–24958, 2025.

[4] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023.

[5] Han Zhang, Zixiang Meng, Meng Luo, Hong Han, Lizi Liao, Erik Cambria, and Hao Fei. Towards multimodal empathetic response generation: A rich text-speech-vision avatar-based benchmark. In *Proceedings of the ACM on Web Conference 2025*, pages 2872–2881, 2025.

[6] Junxian Li, Di Zhang, Xunzhi Wang, Zeying Hao, Jingdi Lei, Qian Tan, Cai Zhou, Wei Liu, Yaotian Yang, Xinrui Xiong, et al. Chemvlm: Exploring the power of multimodal large language models in chemistry area. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 415–423, 2025.

[7] Senqiao Yang, Jiaming Liu, Renrui Zhang, Mingjie Pan, Ziyu Guo, Xiaoqi Li, Zehui Chen, Peng Gao, Hongsheng Li, Yandong Guo, et al. Lidar-llm: Exploring the potential of large language models for 3d lidar understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9247–9255, 2025.

[8] Haotian Xia, Zhengbang Yang, Junbo Zou, Rhys Tracy, Yuqing Wang, Chi Lu, Christopher Lai, Yanjun He, Xun Shao, Zhuoqing Xie, et al. Sportu: A comprehensive sports understanding benchmark for multimodal large language models. *arXiv preprint arXiv:2410.08474*, 2024.

[9] Lei Wang, Yi Hu, Jiabang He, Xing Xu, Ning Liu, Hui Liu, and Heng Tao Shen. T-sciq: Teaching multimodal chain-of-thought reasoning via large language model signals for science question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19162–19170, 2024.

[10] Jiaxing Zhao, Xihan Wei, and Liefeng Bo. R1-omni: Explainable omni-multimodal emotion recognition with reinforcement learning. *arXiv preprint arXiv:2503.05379*, 2025.

[11] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[12] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

[13] Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191, 2023.

[14] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14431, 2024.

[15] Debjyoti Mondal, Suraj Modi, Subhadarshi Panda, Rituraj Singh, and Godawari Sudhakar Rao. Kam-cot: Knowledge augmented multimodal chain-of-thoughts reasoning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 18798–18806, 2024.

[16] Yan Li, Xiangyuan Lan, Haifeng Chen, Ke Lu, and Dongmei Jiang. Multimodal pear chain-of-thought reasoning for multimodal sentiment analysis. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(9):1–23, 2025.

[17] Zhenyu Pan, Haozheng Luo, Manling Li, and Han Liu. Chain-of-action: Faithful and multimodal question answering through large language models. *arXiv preprint arXiv:2403.17359*, 2024.

[18] Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. Mind's eye of llms: visualization-of-thought elicits spatial reasoning in large language models. *Advances in Neural Information Processing Systems*, 37:90277–90317, 2024.

[19] Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei W Koh, and Bryan Hooi. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in llms. *Advances in Neural Information Processing Systems*, 37:24181–24215, 2024.

[20] Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Junqi Dai, Qinyuan Cheng, Xuan-Jing Huang, and Xipeng Qiu. Reasoning in flux: Enhancing large language models reasoning through uncertainty-aware adaptive guidance. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2401–2416, 2024.

[21] Kaiqu Liang, Zixu Zhang, and Jaime F Fisac. Introspective planning: Aligning robots' uncertainty with inherent task ambiguity. *Advances in Neural Information Processing Systems*, 37:71998–72031, 2024.

[22] David Ellerman. Introduction to logical entropy and its relationship to shannon entropy. *arXiv preprint arXiv:2112.01966*, 2021.

[23] Darryl Hannan, Akshay Jain, and Mohit Bansal. Manymodalqa: Modality disambiguation and qa over diverse inputs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7879–7886, 2020.

[24] OpenAI. Update to gpt-5 system card: Gpt-5.2, 2025.

[25] Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

[26] Amazon Web Services. Amazon nova 2 lite – overview, 2025.

[27] Anthropic. Introducing claude sonnet 4.6, 2026.

[28] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[29] NVIDIA. Nemotron nano 12b v2 vl model card, 2025.

[30] Tejas Anvekar, Krishna Singh Rajput, Chitta Baral, and Vivek Gupta. Rethinking information synthesis in multimodal question answering a multi-agent perspective. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 3674–3686, 2025.

[31] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 2609–2634, 2023.

[32] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[33] Bowen Yu, Cheng Fu, Haiyang Yu, Fei Huang, and Yongbin Li. Unified language representation for question answering over text, tables, and images. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4756–4765, 2023.

[34] Qi Zhi Lim, Chin Poo Lee, Kian Ming Lim, and Ahmad Kamsani Samingan. Unirag: Unification, retrieval, and generation for multimodal question answering with pre-trained language models. *IEEE Access*, 12:71505–71519, 2024.

[35] Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16488–16498. IEEE, 2024.

[36] Yongqi Li, Wenjie Li, and Liqiang Nie. Mmcoqa: Conversational question answering over text, tables, and images. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4220–4231, 2022.

[37] Jiapeng Liu, Liang Li, Shihao Rao, Xiyan Gao, Weixin Guan, Bing Li, and Can Ma. Union is strength! unite the power of llms and mllms for chart question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5487–5495, 2025.

[38] Xiaofeng Zhang, Fanshuo Zeng, Yihao Quan, Zheng Hui, and Jiawei Yao. Enhancing multimodal large language models complex reason via similarity computation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 10203–10211, 2025.

[39] Xianwei Zhuang, Zhihong Zhu, Zhichang Wang, Xuxin Cheng, and Yuexian Zou. Unicott: A unified framework for structural chain-of-thought distillation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[40] Guangyao Li, Henghui Du, and Di Hu. Avqa-cot: When cot meets question answering in audio-visual scenarios. 2024.

[41] Yaoting Wang, Peiwen Sun, Yuanchao Li, Honggang Zhang, and Di Hu. Can textual semantics mitigate sounding object segmentation preference? In *European Conference on Computer Vision*, pages 340–356. Springer, 2024.

[42] Farima Fatahi Bayat, Xin Liu, H Jagadish, and Lu Wang. Enhanced language model truthfulness with learnable intervention and uncertainty expression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12388–12400, 2024.

[43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[44] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[45] 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.

[46] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[47] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[48] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.

[49] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

A Notation

Table 3: Summary of notation used.

Symbol	Description
Q	Input question
M	Set of all modalities
$m \in M$	A specific modality
$M_s \subseteq M$	Set of selected starting modalities
M_a	Modalities under analysis
m_t	Candidate target modality
M_c	Set of conditioning modalities ($M_a \setminus \{m_t\}$)
Ω_m	Set of elements for modality m
$\omega \in \Omega_m$	A specific element of modality m
R	Set of answer requirements
$r \in R$	A specific answer requirement
Ω_m^r	Elements of modality m satisfying requirement r
$\Omega_m^{\neg r}$	Elements of modality m not satisfying r
P_m^r	Partition of modality m w.r.t. requirement r
$H(P_m^r)$	Logical entropy of partition P_m^r
$\mathcal{H}_{uni}[m, r]$	Cache storing $(H(P_m^r), \Omega_m^r)$ for each (m, r) pair
E_m	Set of cross-modal evidence extracted from a modality
$E[m, r]$	Cached cross-modal evidence extracted from modality m w.r.t. requirement r
$\Omega_{m_t}^{r,E_m}$	Target elements consistent with requirement r and cross-modal evidence E_m
$\Omega_{m_t}^{r,\neg E_m}$	Target elements consistent with r but inconsistent with E_m
$P_{m_t\|m}^{r,E_m}$	Conditional partition of target modality given E_m
$H(P_{m_t\|m}^{r,E_m})$	Multimodal logical entropy after applying E_m
Ω_{curr}	Elements of m_t still consistent with r, refined across conditioning steps
E_{curr}	Cross-modal evidence accumulated over the conditioning steps
$\mathcal{H}_{multi}[m_t, r]$	Cache storing $(H(P_{m_t\|m}^{r,E_m}), \Omega_{curr}, E_{curr})$ for each (m_t, r) pair
ΔH	Relative change between $H(P_{m_t}^r)$ and $H(P_{m_t\|m}^{r,E_m})$
$\mathcal{T}$	Set of candidate unimodal and multimodal trajectories
t^*	Optimal trajectory selected via entropy minimization
Ω^*	Subset of elements in m^* consistent with r^* and E^*, as selected by t^*
E^*	Cross-modal evidence used for answer generation
a^*	The final generated answer

B Algorithm

The EDMA algorithm has a worst-case computational complexity of $O(M \cdot R \cdot K + M^2 \cdot R)$, where M denotes the number of modalities, R the number of requirements, and K the maximum number of elements per modality. Stage 1 (Unimodal Partitioning) contributes $O(M \cdot R \cdot K)$, evaluating each of the K elements per modality against each requirement, and also caches the cross-modal evidence extracted for each (m, r) pair. Stage 2 (Cross-Modal Sequential Conditioning) contributes $O(M^2 \cdot R)$, refining each of the $M \cdot R$ target pairs by conditioning on the remaining M modalities via $O(1)$ lookups into this cache. Under $M \sim R \sim K \sim n$, the overall complexity is $O(n^3)$.

Algorithm 1 Entropy-Driven Multimodal Answering (EDMA)

Require: Q, M, $\{\Omega_m, m \in M\}$
Ensure: Final answer a^*

$R \leftarrow LLM(Prompt_R, Q)$ *// Extraction of answer requirements*
$M_s \leftarrow LLM(Prompt_{M_s}, Q, \{\Omega_m, m \in M\})$ *// Prediction of Starting Modalities*
$a^* \leftarrow$ None
$M_a \leftarrow \emptyset$
while $a^* =$ None **and** $M_s \neq \emptyset$ **do**
  $M_a \leftarrow M_a \cup M_s$
  $M \leftarrow M \setminus M_s$
  // Stage 1: Unimodal Partitioning (with Extraction Caching)
  for $(m, r) \in M_a \times R$ **do**
    Generate $P_m^r = \{\Omega_m^r, \Omega_m^{\neg r}\}$ and compute $H(P_m^r)$
    Store $H(P_m^r)$ and Ω_m^r in $\mathcal{H}_{uni}[m, r]$
    $E[m, r] \leftarrow \{LLM(Prompt_{Extract}, Q, \omega) \mid \omega \in \Omega_m^r\}$ *// Cached once per* (m, r)
  end for
  // Stage 2: Cross-Modal Sequential Conditioning
  for $(m_t, r) \in M_a \times R$ **do**
    $M_c \leftarrow M_a \setminus \{m_t\}$, $\Omega_{curr} \leftarrow \Omega_{m_t}^r$, $E_{curr} \leftarrow \emptyset$
    for $m \in M_c$ **do**
      $E_m \leftarrow E[m, r]$ *// $O(1)$ lookup*
      Generate $P_{m_t|m}^{r,E_m} = \{\Omega_{m_t}^{r,E_m}, \Omega_{m_t}^{r,\neg E_m}\}$ using Ω_{curr} and E_m
      $\Omega_{curr} \leftarrow \Omega_{m_t}^{r,E_m}$
      $E_{curr} \leftarrow E_{curr} \cup E_m$
    end for
    $H(P_{m_t|m}^{r,E_m}) \leftarrow H(P_{m_t|m}^{r,E_m})$
    Store $H(P_{m_t|m}^{r,E_m}), \Omega_{curr}, E_{curr}$ in $\mathcal{H}_{multi}[m_t, r]$
  end for
  // Stage 3: Trajectory Selection
  $\mathcal{T} \leftarrow \emptyset$
  for $(m_t, r) \in M_a \times R$ **do**
    $H_u, \Omega_u \leftarrow \mathcal{H}_{uni}[m_t, r]$
    $H_m, \Omega_m, E_m \leftarrow \mathcal{H}_{multi}[m_t, r]$
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{(m_t, r, H_u, \Omega_u, \emptyset)\}$ *// Unimodal path*
    if $H_u > 0$ **then** $\Delta H \leftarrow (H_u - H_m)/H_u$ **else** $\Delta H \leftarrow 0$
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{(m_t, r, H_m, \Omega_m, E_m, \Delta H)\}$ *// Multimodal path*
  end for
  Select $t^* = (m^*, r^*, H^*, \Omega^*, E^*)$ from $\mathcal{T}$ that minimizes H
  // Stage 4: Final Inference
  $a^* \leftarrow LLM(Q, \Omega^*, E^*)$ *// Inference from optimal subset and evidence*
  if $a^* =$ None **and** $M \neq \emptyset$ **then**
    $M_s \leftarrow LLM(Prompt_{M_s}, Q, \{\Omega_m, m \in M\})$
  else
    $M_s \leftarrow \emptyset$
  end if
end while
return a^*

C Experimental Setup

C.1 Baselines

In this section, we provide detailed descriptions of the baselines used in our experiments. We categorize these into multimodal architectures and prompting-based strategies.

C.1.1 Multimodal QA Systems

MAMMQA [30] A multi-agent QA framework for multimodal inputs spanning text, tables, and images. The system comprises two Vision-Language Model (VLM) agents and one text-based LLM agent. The first VLM decomposes the query into sub-questions to sequentially retrieve partial evidence across modalities, the second VLM synthesizes intermediate insights via cross-modal reasoning, and the LLM aggregates the findings into a cohesive final answer.

MultiModalQA [1] The baseline pipeline introduced alongside the MultiModalQA benchmark. It employs modality-specific information extractors combined with an *ImplicitDecomp* module to perform multi-step question decomposition and answer synthesis across text, tables, and images.

UniMMQA [2] A unified multimodal QA framework designed to reduce the modality gap by projecting heterogeneous data (text, tables, and images) into a shared representation space, enabling integrated cross-modal retrieval and answer generation.

Solar [33] A sequence-to-sequence multimodal framework that unifies visual and textual token processing within a single transformer architecture, optimized for cross-modal alignment through joint pre-training objectives.

UniRAG [34] A retrieval-augmented generation (RAG) framework that unifies evidence retrieval across textual, visual, and tabular corpora before feeding the aggregated context into a Large Multimodal Model (LMM) for answer generation.

ManyModalQA [23] The baseline framework introduced with the ManyModalQA benchmark. It incorporates an explicit modality routing mechanism to determine the most relevant data source (text, image, or table) prior to applying specialized extraction modules to return the final answer.

C.1.2 Prompting Strategies

Direct Prompting (DP) A standard zero-shot prompting approach where the model receives the input context and query, and directly outputs the final answer without explicit intermediate reasoning steps.

Chain-of-Thought (CoT) [11] A prompting method that prompts the model to generate a sequence of intermediate natural language reasoning steps before outputting the final answer within a single generation stream.

Planning Prompting (PP) [31] A two-stage prompting technique that decouples reasoning planning from execution: the model first generates a high-level plan of sub-tasks, and subsequently executes each sub-task to derive the final answer.

CoT Self-Consistency (CoT-SC) [32] An ensemble strategy that samples N independent Chain-of-Thought reasoning paths and selects the final answer via majority voting.

PP Self-Consistency (PP-SC) An extension of self-consistency [32] applied to Planning Prompting, sampling N independent plan-and-execute reasoning trajectories and selecting the answer via majority voting.

C.2 Datasets

MultiModalQA [1] is a comprehensive benchmark designed to evaluate a model's ability to perform joint integration across three distinct data modalities: structured tables, natural language text, and images. Unlike earlier multimodal datasets where the answer usually resides within a single source (e.g., just an image or just a paragraph), MultiModalQA is specifically engineered to require multi-hop reasoning. We report results on the validation set since ground-truth annotations for the test set are not publicly accessible.

To further show the generalizability of our methodology, we test it also on **ManyModalQA** [23], which is a multimodal question-answering benchmark designed to evaluate an agent's ability to reason across three distinct data types: textual passages, structured tables, and visual images. Unlike standard multimodal datasets where the target modality is often implied by the question, ManyModalQA contains ambiguous questions where the source of the answer cannot be determined by the question text alone.

D Comparison with Previous Multimodal QA Systems

Table 4 compares EDMA with prior multimodal QA systems. EDMA achieves the best performance across both datasets. On MultiModalQA, it obtains 82.8 EM and 83.5 F1, improving over the strongest prior system (UniMMQA) by $+9.1$ EM and $+3.4$ F1. On ManyModalQA, the gains are substantially larger, with improvements of $+35.9$ EM and $+36.1$ F1 over UniMMQA. We note that these comparisons reflect both methodological improvements and differences in underlying foundation models.

Table 4: Comparison of EDMA against prior specialized multimodal QA frameworks on MultiModalQA and ManyModalQA datasets.

Approach	Model	EM	F1
MultiModalQA [1]			
EDMA	Claude-4.6	**82.8**	**83.5**
UniMMQA [2]	T5-3B [43]	73.7	80.1
UniRAG [34]	LLaVA-v1.5-7B [44]	67.4	71.3
Solar [33]	T5 [43] + BLIP [45]	59.8	66.1
MultiModalQA [1]	RoBERTa-large [46] + ViLBERT [47]	49.3	55.9
ManyModalQA [23]			
EDMA	GPT-5.2	**88.0**	**88.5**
UniMMQA [2]	T5-3B [43]	52.1	52.4
ManyModalQA [23]	ELMo [48] + BERT [49]	39.7	39.7

E Models

E.1 Access to Models

We access GPT-5.2 via its official API and all remaining models through Amazon Bedrock. For all experiments, we use greedy decoding (temperature $= 0$) with a maximum generation length of 1000 tokens.

E.2 Computational Resources

Table 5 details the raw average token consumption, API call counts, monetary cost, and latency per model across evaluated approaches (Nova-2-Lite, GPT-5.2, Kimi-K2.5, Nemotron, Qwen3, and Claude-4.6). Furthermore, Table 6 summarizes the relative computational overhead of EDMA compared to each baseline strategy.

As shown in Table 6, EDMA is on average $22.83\times$ slower than CoT, $45.79\times$ slower than DP, $19.92\times$ slower than PP, and $11.05\times$ slower than MAMMQA. This reflects the higher computational complexity of the algorithm. Regarding financial cost, EDMA is on average $5.06\times$ more expensive

Table 5: Average computational cost and latency across models and approaches.

Model	Approach	Avg. Total Tokens	Avg. # Calls	Avg. Cost ($)	Latency (s)
Nova-2-Lite	CoT	19,775.20	1.00	0.006	∼5.04
Nova-2-Lite	DP	19,576.12	1.00	0.006	∼3
Nova-2-Lite	EDMA	89,909.88	143.35	0.027	∼30
Nova-2-Lite	PP	20,070.06	1.00	0.007	∼5
Nova-2-Lite	MAMMQA	79,972.14	7.00	0.024	∼10
GPT-5.2	CoT	4,616.74	1.00	0.009	∼3
GPT-5.2	DP	4,511.48	1.00	0.008	∼1.50
GPT-5.2	EDMA	68,574.99	181.75	0.120	∼120
GPT-5.2	PP	4,935.78	1.00	0.013	∼2.75
Kimi-K2.5	CoT	19,812.28	1.00	0.012	∼2
Kimi-K2.5	DP	19,587.54	1.00	0.012	∼3.5
Kimi-K2.5	EDMA	91,275.13	165.43	0.055	∼90
Kimi-K2.5	PP	20,021.98	1.00	0.013	∼4.07
Kimi-K2.5	MAMMQA	80,172.47	7.00	0.048	∼11
Nemotron	CoT	19,751.52	1.00	0.003	∼3
Nemotron	DP	19,602.88	1.00	0.003	∼1
Nemotron	EDMA	90,438.65	151.03	0.014	∼90
Nemotron	PP	19,828.78	1.00	0.003	∼3
Nemotron	MAMMQA	79,951.83	7.00	0.012	∼5
Qwen3	CoT	19,836.08	1.00	0.005	∼20
Qwen3	DP	19,574.92	1.00	0.004	∼5
Qwen3	EDMA	90,314.85	149.31	0.020	∼120
Qwen3	PP	20,047.36	1.00	0.005	∼18.37
Claude-4.6	CoT	19,871.12	1.00	0.063	∼18
Claude-4.6	DP	19,586.50	1.00	0.059	∼4
Claude-4.6	EDMA	91,322.96	165.35	0.274	∼180
Claude-4.6	PP	19,937.36	1.00	0.063	∼16.01
Claude-4.6	MAMMQA	81,253.41	7.00	0.244	∼12

than PP, $5.91\times$ more expensive than CoT, and $6.40\times$ more expensive than DP. However, its cost is directly comparable to MAMMQA ($1.14\times$). Although MAMMQA executes only seven API calls per query, it repeatedly inputs all data modalities (text, images, and tables) into the LLM context, making its token consumption and cost comparable to EDMA. Note that estimates for MAMMQA were computed across all models except GPT-5.2 and Qwen3, where evaluations were omitted due to budget restrictions.

Table 6: Average cost and latency overhead of EDMA compared with other approaches.

Approach	Avg. Absolute Cost Difference vs. EDMA ($)	Avg. Cost Ratio ($\frac{\text{EDMA}}{\text{Approach}}$)	Avg. Latency Ratio ($\frac{\text{EDMA}}{\text{Approach}}$)
CoT	0.069	$5.91\times$	$22.83\times$
DP	0.070	$6.40\times$	$45.79\times$
PP	0.068	$5.06\times$	$19.92\times$
MAMMQA	0.011	$1.14\times$	$11.05\times$

F Analysis of Intermediate Operations

Table 7 reports the average accuracy across each intermediate phase of EDMA, consisting in modality selection, criteria extraction, intermediate answer extraction, and multimodal fusion, evaluated specifically on questions where EDMA produced an incorrect final answer.

Across most models, errors primarily originate during multimodal fusion, where the system must cross-reference information extracted from one modality (e.g., text) with another (e.g., images). In

Table 7: Intermediate-step accuracy for questions incorrectly answered by EDMA.

Model	Modality Selection	Criteria Extraction	Intermediate Answers	Multimodal Fusion
Nova-2-Lite	90%	94%	92%	4%
GPT-5.2	96%	98%	96%	6%
Kimi-K2.5	96%	96%	96%	8%
Nemotron	90%	94%	30%	2%
Qwen3	96%	96%	96%	8%
Claude-4.6	96%	98%	96%	8%

these cases, incorrect final answers stem from over-filtered or wrongly filtered partitions, resulting in the selection of a low-entropy yet incorrect multimodal partition. Nemotron is the sole exception: its failures are driven by low accuracy at the earlier Intermediate Answers stage (30%), causing trajectories to fail before reaching the fusion step.

Table 8 presents the corresponding accuracy metrics across all evaluated questions, both correctly and incorrectly answered. Early-stage accuracies (modality selection, criteria extraction, and intermediate answer extraction) remain consistently high across the full dataset. The critical divergence appears in the multimodal fusion phase, confirming that cross-modal fusion constitutes the main point of failure in the methodology.

Table 8: Intermediate-step accuracy across all questions, both correctly and incorrectly answered.

Model	Modality Selection	Criteria Extraction	Intermediate Answers	Multimodal Fusion
Nova-2-Lite	95%	96%	93%	70%
GPT-5.2	98%	98%	98%	82%
Kimi-K2.5	98%	97%	98%	86%
Nemotron	93%	94%	89%	76%
Qwen3	97%	96%	97%	85%
Claude-4.6	98%	98%	97%	91%

G EDMA Statistical Variance

Table 9 reports the mean and standard deviation calculated across ten independent runs. The analysis confirms that the performance advantages observed in single-point comparisons remain stable under repeated evaluation: EDMA's lead over MAMMQA on Claude-4.6, Nemotron, and Nova-2-Lite, as well as MAMMQA's advantage on Kimi-K2.5, hold consistently across runs.

Table 9: Mean $\pm$ standard deviation of EM, F1, and LLM-judge score (LLM-J) over 10 runs at temperature 0. MAMMQA is only evaluated with LLM-J.

Model	Metric	EDMA	PP	DP	CoT	MAMMQA
Claude-4.6	EM	**82.7 $\pm$ 1.9**	57.9 $\pm$ 1.9	56.1 $\pm$ 2.5	57.8 $\pm$ 2.0	–
	F1	**83.0 $\pm$ 1.1**	70.7 $\pm$ 1.3	68.1 $\pm$ 1.3	70.8 $\pm$ 0.8	–
	LLM-J	**88.8 $\pm$ 0.5**	85.4 $\pm$ 0.4	82.6 $\pm$ 0.5	87.5 $\pm$ 0.5	82.6 $\pm$ 1.3
Kimi-K2.5	EM	**80.7 $\pm$ 1.8**	60.2 $\pm$ 2.1	61.9 $\pm$ 0.6	58.1 $\pm$ 3.1	–
	F1	**81.3 $\pm$ 0.3**	72.4 $\pm$ 1.0	73.7 $\pm$ 0.3	68.7 $\pm$ 2.3	–
	LLM-J	83.3 $\pm$ 0.5	81.7 $\pm$ 0.4	52.0 $\pm$ 0.4	79.3 $\pm$ 0.4	**88.3 $\pm$ 0.9**
Nemotron	EM	**80.0 $\pm$ 0.8**	51.9 $\pm$ 0.9	41.2 $\pm$ 0.7	46.4 $\pm$ 0.2	–
	F1	**80.7 $\pm$ 0.8**	61.3 $\pm$ 0.9	50.1 $\pm$ 0.2	54.8 $\pm$ 0.4	–
	LLM-J	**74.9 $\pm$ 0.6**	64.9 $\pm$ 0.6	52.2 $\pm$ 0.4	57.0 $\pm$ 0.5	66.2 $\pm$ 1.1
Nova-2-Lite	EM	**79.1 $\pm$ 0.6**	47.2 $\pm$ 1.8	51.1 $\pm$ 0.6	42.3 $\pm$ 0.6	–
	F1	**80.4 $\pm$ 1.1**	57.1 $\pm$ 1.5	60.3 $\pm$ 0.7	50.6 $\pm$ 0.3	–
	LLM-J	**66.6 $\pm$ 0.4**	42.1 $\pm$ 0.3	64.4 $\pm$ 0.4	53.3 $\pm$ 0.3	65.9 $\pm$ 1.4
Average	EM	**80.7 $\pm$ 1.2**	54.3 $\pm$ 1.7	52.6 $\pm$ 1.1	51.2 $\pm$ 1.5	–
	F1	**81.3 $\pm$ 0.8**	65.4 $\pm$ 1.2	63.1 $\pm$ 0.6	61.2 $\pm$ 0.9	–
	LLM-J	**78.4 $\pm$ 0.5**	68.5 $\pm$ 0.4	62.8 $\pm$ 0.4	69.3 $\pm$ 0.4	75.8 $\pm$ 1.2

H Ablation Studies

This section presents two ablation studies on the MultiModalQA dataset to isolate the contributions of EDMA's key components: entropy minimization and cross-modal conditioning.

H.1 Ablation 1: Random Trajectory Selection

Table 10 shows the performance impact of disabling entropy minimization during trajectory selection. Instead of choosing the candidate trajectory that minimizes logical entropy, a trajectory is selected uniformly at random while keeping all other pipeline components identical. To account for the stochasticity of random selection, results are averaged over 10 runs using the LLM-Judge metric. Across all evaluated models, selecting trajectories uniformly at random causes a severe performance drop (-32.1 percentage points on average) compared to entropy-guided selection. This shows that EDMA's performance gains do not merely stem from task decomposition into intermediate partitions, but critically rely on selecting the trajectory that minimizes logical entropy.

Table 10: Performance comparison between full EDMA and random trajectory selection.

Model	EDMA	EDMA (Random Trajectory)	Difference
GPT-5.2	80.5	45.8	-34.7
Nova-2-Lite	65.9	47.3	-18.6
Kimi-K2.5	83.2	37.1	-46.1
Nemotron	75.1	51.6	-23.5
Qwen3	82.5	48.2	-34.3
Claude-4.6	88.9	53.0	-35.9
Average	**79.3**	**47.2**	**-32.1**

H.2 Ablation 2: Removing Cross-Modal Conditioning

To isolate the contribution of cross-modal conditioning, we disable multimodal partitioning. Specifically, we execute requirement extraction, unimodal partitioning, and entropy-based trajectory selection, but omit the construction of multimodal partitions via cross-modal conditioning. Final answers are thus generated solely from the lowest-entropy unimodal partition.

Table 11: Performance comparison between full EDMA and unimodal-only partitioning.

Model	EDMA	EDMA (Unimodal-only)	Difference
GPT-5.2	80.5	65.5	-15.0
Nova-2-Lite	65.9	48.0	-17.9
Kimi-K2.5	83.2	73.1	-10.1
Nemotron	75.1	56.8	-18.3
Qwen3	82.5	68.4	-14.1
Claude-4.6	88.9	77.8	-11.1
Average	**79.3**	**64.9**	**-14.4**

As shown in Table 11, removing cross-modal conditioning leads to an average performance drop of 14.4 percentage points across all models. This drop is of the same order as the share of questions that require cross-modal reasoning: approximately 35% of questions in MultiModalQA strictly require it, whereas $\sim 65\%$ can be answered using a single modality. This is consistent with the framework failing to resolve cross-modal dependencies without cross-modal conditioning, and confirms that multimodal partitioning is fundamental for handling multimodal queries.

I Statistical significance

I.1 EDMA vs PP

We establish the statistical reliability of the performance gains achieved by EDMA over PP by conducting paired statistical significance testing across four backbones.

Statistical significance is assessed using both a parametric test (paired two-tailed t-test) and a non-parametric test (two-tailed Wilcoxon signed-rank test) at a significance threshold of $\alpha = 0.05$.

Table 12 reports the statistical comparison between EDMA and PP across backbones. EDMA achieves statistically significant improvements ($p < 0.05$) over PP across each of the four backbones and all three metrics evaluated under both parametric and non-parametric tests, demonstrating robust gains across diverse model families.

Table 12: Statistical comparison of EDMA against PP across four backbones. Δ represents the mean score difference (EDMA $-$ PP). Tick (✓) indicates significance at $p < 0.05$.

Backbone	Metric	EDMA (mean)	PP (mean)	Δ	Sig. (t-test)	Sig. (Wilcoxon)
Claude-4.6	EM	82.7	57.9	+24.8	✓	✓
	F1	83.0	70.7	+12.3	✓	✓
	LLM-judge	88.8	85.4	+3.4	✓	✓
Kimi-K2.5	EM	80.7	60.2	+20.5	✓	✓
	F1	81.3	72.4	+8.9	✓	✓
	LLM-judge	83.3	81.7	+1.6	✓	✓
Nemotron	EM	80.0	51.9	+28.1	✓	✓
	F1	80.7	61.3	+19.4	✓	✓
	LLM-judge	74.9	64.9	+10.0	✓	✓
Nova-2-Lite	EM	79.1	47.2	+31.9	✓	✓
	F1	80.4	57.1	+23.3	✓	✓
	LLM-judge	66.6	42.1	+24.5	✓	✓

I.2 Modality Fusion Decreases Entropy

We analyse the role of modality fusion in EDMA through the lens of logical entropy reduction over the candidate answer space. Figure 5 compares the distribution of entropy values before and after modality fusion, which occurs when cross-modal evidence is extracted from one modality and used to condition the target modality.

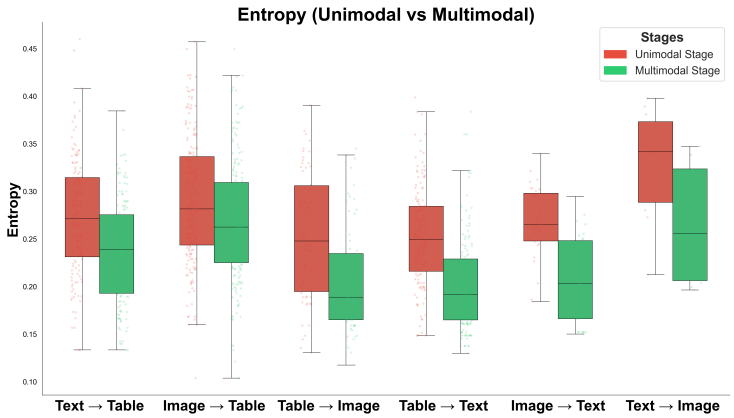


Figure 5: Multimodal entropy distribution consistently shows lower values compared to unimodal entropy distributions.

Conditioning the target modality with complementary information from a secondary modality decreases logical entropy, thereby reducing uncertainty. This shows that modality fusion is associated

with lower entropy, mitigating uncertainty arising from noisy or ambiguous unimodal signals. Table 13 reports the statistical significance of these reductions across all directional modality pairs.

Table 13: Statistical Evaluation of Entropy Reduction via Information Fusion. $\overline{H}$ represents mean entropy; t represents the t-statistic; Point-Biserial (PB) is the correlation between a binary modality indicator (unimodal vs. multimodal) and entropy values. Negative PB values reflect lower entropy under multimodal representations. Tick (✓) indicates significance at $p < 0.05$.

			T-Test		Point-Biserial	
Combination direction	$\overline{H(Uni)}$	$\overline{H(Multi)}$	**Stat** (t)	**Sig.** (p)	**Corr.** (r)	**Sig.** (p)
Image → Table	0.29	0.27	8.84	✓	-0.16	✓
Image → Text	0.27	0.21	5.80	✓	-0.60	✓
Table → Image	0.25	0.21	6.82	✓	-0.36	✓
Table → Text	0.25	0.20	12.16	✓	-0.43	✓
Text → Image	0.33	0.27	4.11	✓	-0.47	✓
Text → Table	0.27	0.24	9.07	✓	-0.29	✓

I.3 Higher Entropy Is Associated with More False Positives

To investigate the relationship between entropy and error rates, we compare entropy levels between correct answers (True Positives, TP) and incorrect answers (False Positives, FP). As detailed in Table 14, false positives consistently exhibit higher mean entropy across all modalities.

Table 14: Statistical comparison of entropy in True Positives (TP) vs. False Positives (FP). $\overline{H}$ denotes mean entropy; t denotes the t-test statistic; PB denotes the point-biserial correlation between entropy and a binary outcome variable (TP = 0, FP = 1). Tick (✓) indicates significance at $p < 0.05$.

			T-Test		Point-Biserial	
Modality	$\overline{H(TP)}$	$\overline{H(FP)}$	**Stat** (t)	**Sig.** (p)	**Corr.** (r)	**Sig.** (p)
Text	0.20	0.24	-8.72	✓	0.26	✓
Image	0.21	0.26	-5.78	✓	0.26	✓
Table	0.21	0.27	-8.66	✓	0.23	✓

We further assess the impact of entropy on False Discovery Rates (FDR) by grouping entropy into Low, Medium, and High level. Table 15 confirms significance via the Friedman test, while Table 16 provides pairwise Wilcoxon signed-rank comparisons across all entropy levels.

Table 15: Statistical comparison across modalities using Mean False Discovery Rates ($\overline{\text{FDR}}$) and the Friedman Test for Related Samples. Q denotes the Friedman test statistic. Tick (✓) indicates significance at $p < 0.05$.

	Mean FDR			Friedman Test	
Modality	$\overline{\text{FDR}}_{\text{Low}}$	$\overline{\text{FDR}}_{\text{Medium}}$	$\overline{\text{FDR}}_{\text{High}}$	**Stat** (Q)	**Sig.** (p)
Text	0.081	0.475	0.503	130.72	✓
Image	0.189	0.766	0.782	108.10	✓
Table	0.133	0.459	0.491	40.36	✓

Table 16: Pairwise comparisons of Mean False Discovery Rates ($\overline{\text{FDR}}$) across entropy levels within each modality using the Wilcoxon Signed-Rank Test for paired samples. Tick (✓) indicates significance at $p < 0.05$.

	Low vs Medium			Low vs High		
Modality	**Low**	**Med.**	**Sig. (p)**	**Low**	**High**	**Sig. (p)**
Text	0.081	0.475	✓	0.081	0.503	✓
Image	0.189	0.766	✓	0.189	0.782	✓
Table	0.133	0.459	✓	0.133	0.491	✓

Table 17 breaks down the False Discovery Rate (FDR) controlling for modality paths and candidate set size. We split candidate set sizes (ranging from 13 to 26) into three buckets containing approximately 33% of the queries each. Even after controlling for modality path and candidate set size, the FDR increases as entropy increases from low to medium to high.

Table 17: False Discovery Rate (FDR) across modality paths, candidate set size buckets, and entropy levels. FDR increases consistently as entropy increases from Low to Medium and High.

Modality Pair	**Candidate Bucket**	**Low Entropy FDR**	**Medium Entropy FDR**	**High Entropy FDR**
Text	13–19	0.161	0.479	0.480
Text	20–23	0.084	0.359	0.482
Text	24–26	0.071	0.426	0.461
Text → Image	13–19	0.250	0.933	1.000
Text → Image	20–23	0.267	0.917	1.000
Text → Table	13–19	0.200	0.350	0.533
Text → Table	20–23	0.167	0.428	0.514
Text → Table	24–26	0.130	0.258	0.481
Image → Text	13–19	0.200	0.678	1.000
Image → Text	20–23	0.167	0.333	0.433
Image → Text	24–26	0.173	0.851	1.000
Image	13–19	0.273	0.758	0.766
Image	20–23	0.185	0.678	0.784
Image	24–26	0.158	0.799	0.860
Image → Table	13–19	0.175	0.650	0.823
Image → Table	20–23	0.272	0.550	0.821
Image → Table	24–26	0.217	0.769	0.879
Table → Text	13–19	0.111	0.565	0.620
Table → Text	20–23	0.125	0.600	0.702
Table → Text	24–26	0.045	0.432	0.436
Table → Image	13–19	0.131	0.889	0.933
Table → Image	20–23	0.067	0.675	0.711
Table → Image	24–26	0.200	0.583	0.875
Table	13–19	0.131	0.235	0.475
Table	20–23	0.077	0.107	0.177
Table	24–26	0.036	0.414	0.414

J Impact Assessment

The introduction of the EDMA methodology enhances the ability of LLMs to solve cross-modal problems by guiding the answering process through entropy minimization and producing a quantifiable answer trajectory. This traceability allows users to understand which modalities contributed most to the final answer, improving the interpretability and reliability of LLM predictions in real-world scenarios. By providing structured answering trajectories, EDMA also makes LLM-based systems more robust and trustworthy. While privacy and trustworthiness are general concerns associated with LLMs, our methodology does not introduce additional risks beyond those already inherent to LLMs.

K Prompts

K.1 Prompts EDMA

K.1.1 Prompt for Modality Decision

Table 18: Prompt used to determine which modalities are required to answer a query.

System Prompt

You are a Question Answering assistant.
Your task is to determine which modality or combination of modalities is required to answer a given question. You are provided with: text paragraphs, images (with titles), and tables.

Select exactly one of the following labels:

- `image`: the answer can be derived from images alone
- `text`: the answer can be derived from text alone
- `table`: the answer can be derived from tables alone
- `image_text`: the answer requires both images and text
- `image_table`: the answer requires both images and tables
- `text_table`: the answer requires both text and tables
- `text_table_image`: the answer requires text, tables, and images

Guidelines:

- Use only the provided information
- Output exactly one label, with no additional text

User Prompt

```
Question: {question}
Images: {images_text}
Text Paragraphs: {paragraphs_text}
Tables: {tables_text}
```

Select the required modality or modality combination.

K.1.2 Prompt for Extraction of Answer Requirements

Table 19: Prompt used to extract structured answer requirements from natural-language questions.

System Prompt

You are a Question Answering assistant.
You task is to convert natural-language questions into structured query frames.

Given a question, extract a structured representation with the following components:

- **Topic.** Select exactly one topic from the predefined set: `{topics}`. Choose the most dominant topic; if unclear, use `Other`.
- **Expected Answer Type.** Identify both: (i) a specific answer type, and (ii) a general entity category. Focus on the type of entity the answer represents.
- **Target Entity.** Identify the main entity referenced in the question.
- **Constraints.** Extract atomic, independently verifiable conditions from the question. Each constraint should include:
  1. a constraint type (e.g., relation, award, time, role, qualifier)
  2. the exact supporting phrase from the question
  3. a normalized version of that phrase

User Prompt

```
Question: {question_text}
```

K.1.3 Prompt for Checking Answer Requirement in Paragraph

Table 20: Prompt to determine the presence of content overlap between a paragraph and an answer requirement.

System Prompt

You are a Question Answering assistant.
Your task is to determine whether a text paragraph (or its title) contains any concept specified in a given answer requirement.

User Prompt

```
Title: {metadata}
Paragraph: {text}
Answer requirement: {r}
```

Determine whether the paragraph contains any concept from the answer requirement. If at least one concept from the answer requirement is explicitly supported by the paragraph or its title, respond `yes`; otherwise, respond `no`.

K.1.4 Prompt for Checking Answer Requirement in Image

Table 21: Prompt to determine the presence of content overlap between an image and an answer requirement.

System Prompt

You are a Question Answering assistant.
Your task is to determine whether an image or its title contains any concept specified by an answer requirement.

User Prompt

```
Title: {metadata}
Image: {image data}
Answer requirement: {r}
```

If at least one concept from the answer requirement is explicitly supported by the image content or its title, respond `yes`; otherwise, respond `no`.

K.1.5 Prompt for Checking Answer Requirement in Table Row

Table 22: Prompt to determine the presence of content overlap between a table row and an answer requirement.

System Prompt

You are a Question Answering assistant.
Your task is to determine whether a table row contains any concept specified a given answer requirement.

User Prompt

```
Table Row: {table row}
Answer requirement: {r}
```

If at least one concept from the answer requirement is explicitly supported by the row content, respond `yes`; otherwise, respond `no`.

K.1.6 Prompt to Extract Information from Text

Table 23: Prompt used for extracting question-relevant evidence from a paragraph.

System Prompt

You are a Question Answering assistant.
Your task is to extract relevant information from a given paragraph in relation to a question. You are given the question, the paragraph title, and the paragraph. Follow the instructions to extract the relevant information:

1. **Step 1.** Question analysis: determine whether the paragraph is related to the question using only the provided inputs.
2. **Step 2.** Evidence extraction: if the paragraph is related to the question, extract the minimal information relevant to answering the question. The description must include only the entity or value that directly connects the paragraph to the question.

User Prompt

```
Question: {question_text}
Paragraph Title: {title}
Paragraph: {text}
```

Provide the relevant information according to the instructions.

K.1.7 Prompt to Extract Information from Image

Table 24: Prompt used for extracting question-relevant evidence from an image.

System Prompt

You are a Question Answering assistant.
Your task is to extract relevant information from a given image in relation to a question. You are given the question, the image title, and the image. Follow the instructions to extract the relevant information:

- **Step 1.** Question analysis: determine whether the image is related to the question using only the provided inputs.
- **Step 2.** Evidence extraction: if the image is related to the question, produce a short textual description of the image that captures the minimal information relevant to answering the question. The description must include only the entity or value that directly connects the image to the question.

User Prompt

```
Question: {question_text}
Image Title: {title}
Image: {text}
```

Provide the relevant information according to the instructions.

K.1.8 Prompt to Extract Information from Table Row

Table 25: Prompt used for extracting question-relevant evidence from a table row.

System Prompt

You are a Question Answering assistant.
Your task is to extract relevant information from a given table row in relation to a question. You are given the question and the table row. Follow the instructions to extract the relevant information:

- **Step 1.** Question analysis: determine whether the table row is related to the question using only the provided inputs.
- **Step 2.** Evidence extraction: if the table row is related to the question, produce a short textual description of the table row that captures the minimal information relevant to answering the question. The description must include only the entity or value that directly connects the table row to the question.

User Prompt

```
Question: {question_text}
Paragraph Title: {title}
Paragraph: {text}
```

Provide the relevant information according to the instructions.

K.1.9 Prompt to Check Mutual Information Between Paragraphs

Table 26: Prompt to determine the presence of mutual information between two paragraphs.

System Prompt

You are a Question Answering assistant.
You are given a question and two paragraphs. Your task is to determine whether any entity, concept, or fact mentioned in one paragraph is also mentioned in the other paragraph.

User Prompt

```
Question: {question_text}
Title Paragraph 1: {title_1}
Paragraph 1: {content_paragraph_1}
Title Paragraph 2: {title_2}
Paragraph 2: {content_paragraph_2}
```

Determine whether any entity, concept, or fact mentioned in `Paragraph 1` is also present in `Paragraph 2`. If any shared entity, name, or concept appears in both paragraphs, respond `yes`. Otherwise respond `no`.

K.1.10 Prompt to Check Mutual Information Between Textual Description and Image

Table 27: Prompt to check the presence of mutual information between an image and a paragraph.

System Prompt

You are a Question Answering assistant.
You are given a question, one image, and an image title and a textual description.
Your task is to determine whether any entity, concept, or fact mentioned in the textual description is also present in the image or image title.

User Prompt

```
Question: {question_text}
Text: {content}
Image Title: {image_title}
Image: {image content}
```

Determine whether any entity, concept, or fact mentioned in the `Text` is also present in the `Image Title` or in the `Image`. If any shared entity, name, or concept appears in both the `Text` and the `Image Title` or in the `Text` and the `Image`, respond `yes`. Otherwise respond `no`.

K.1.11 Prompt to Check Mutual Information Between Textual Description and Table Row

Table 28: Prompt to check the presence of mutual information between a table row and a paragraph.

System Prompt

You are a Question Answering assistant.
You are given a question, paragraph (textual description), a table description, and a table row.
Your task is to determine whether any entity, concept, or fact mentioned in the textual description is also present in the table row. Use the table description to understand the context.

User Prompt

```
Question: {question_text}
Paragraph: {content}
Table Description: {table_description}
Table Row: {table_row}
```

Determine whether any entity, concept, or fact mentioned in the `Paragraph` is also present in the `Table Row`. If any shared entity, name, or concept appears in both the `Paragraph` and the `Table Row`, respond `yes`. Otherwise respond `no`.

K.1.12 Check Answer in Paragraph

Table 29: Prompt to check whether a paragraph contains an entity that matches the answer type.

System Prompt

You are a Question Answering assistant.
You are given a question, a paragraph, the expected answer type. Your task is to determine whether the paragraph contains an entity that completely or partially matches the expected answer type required by the question.
Guidelines:

- This is not keyword matching. Perform semantic reasoning using synonyms, paraphrases, and implicit mentions
- Only return `TRUE` if the paragraph contains a specific entity that could fully or partially answer the question. Partial answers are acceptable if relevant.
- If the paragraph only provides context without an answer entity, return `FALSE`. If uncertain, return `FALSE`.
- If `TRUE`, extract the exact text span from the paragraph without paraphrasing or normalization. If `FALSE`, return `NONE`

User Prompt

```
Question: {question_text}
Expected Answer Type: {answer_class}
Paragraph: {paragraph_text}
```

Check the presence of the answer based on the instructions provided above.

K.1.13 Check Answer in Image

Table 30: Prompt to check if the caption or the image contains an entity that matches the answer type.

System Prompt

You are a Question Answering assistant.
Your task is to determine whether either the caption or the image contain an entity that matches the expected answer type required by the question. You are given the question, the image title, and an image. You must evaluate both the image title and the image.

Guidelines:

- First check whether the caption explicitly contains an entity that satisfies the expected answer type. If not, check whether the image contains a visually identifiable entity that satisfies the answer type. If neither modality satisfies either answer type, return `FALSE`.
- If uncertain, return `FALSE`.
- If evidence comes from the caption, extract the exact text span. If evidence comes from the image, provide a concise visual description. If no match, the returned entity must be `NONE`.
- Critical instructions for the Image: Examine all parts of the image carefully, including background, corners, and small or partially obscured objects. Pay attention to subtle details such as tiny text, small objects, fine patterns, colours, positions, or partially visible entities. Do not rely only on large, obvious objects. Small details may be crucial to answer the question. Only return `TRUE` for contains if the entity is clearly present in the image. If uncertain, return `FALSE`.

User Prompt

```
Question: {question_text}
Expected Answer Type: {answer_class_specific}
Image Title: {caption_text}
```

Check the presence of the answer in the image and its title based on the instructions provided above.

K.1.14 Check Answer in Table Row

Table 31: Prompt to verify whether a table row contains a value that directly answers a question.

System Prompt

You are a Question Answering assistant.
Your task is to determine whether a table row contains a value that directly answers a question.
You are given the question, the expected answer type, the table description, and one table row.

Guidelines:

- Use the description of the table to interpret the row correctly
- Do not use only literal meaning. Also use synonyms and contextual meaning
- Only return `TRUE` if one of the row's values directly answers the question
- If the row contains related information but no direct answer, return `FALSE`
- If uncertain, return `FALSE`
- If `TRUE`, extract the exact cell value from the row that contains the answer. If `FALSE`, return `NONE`

User Prompt

```
Question: {question_text}
Expected Answer Type: {answer_class_specific}
Table Description: {table_description}
Table Row: {row}
```

Check the presence of the answer in table row based on the instructions provided above.

K.2 Baselines Prompts

K.2.1 Prompt for Direct Prompting

Table 32: Direct prompting baseline used in our experiments.

System Prompt

You are a multimodal question answering assistant.
Your task is to answer a given question using three types of provided data sources: text paragraphs, images (with titles), and tables. Carefully review all sources and derive the answer by combining relevant information across modalities.

The final response should be concise and contain only the answer, without any explanation. If the necessary information is not available, output `None`.

User Prompt

```
Question: {question_text}
Images: {images_text}
Text Paragraphs: {paragraphs_text}
Tables: {tables_text}
```

Provide the final answer to the question.

K.2.2 Prompt for Chain-of-Thought

Table 33: Chain-of-Thought (CoT) prompt used in our experiments.

System Prompt

You are a multimodal question answering assistant. Your task is to answer a given question using only the provided information sources: text paragraphs, images (with titles), and tables. You must not rely on any external or prior knowledge. If the provided information is insufficient, output `None`.

Procedure to follow:

- **Step 1.** Extract information: Identify relevant facts from each modality: text (key statements), images (objects, labels, relationships), and tables (rows, columns, values).
- **Step 2.** Align information: Match entities and concepts that appear across different modalities and explicitly link them.
- **Step 3.** Integrate information: Combine the aligned evidence to form a complete understanding, ensuring that all parts of the question are addressed.
- **Step 4.** Derive the answer: Formulate the final answer, clearly grounded in the combined evidence from the modalities.
- **Step 5.** Verify: Ensure that the answer is fully supported by the provided information and that no relevant modality has been ignored.

Output the reasoning separated from the final response. The final response should be concise and contain only the answer, without any explanation.

User Prompt

```
Question: {question_text}
Images: {images_text}
Text Paragraphs: {paragraphs_text}
Tables: {tables_text}
```

Answer the question following the procedure above.

K.2.3 Prompt for Planning Prompting

Table 34: Planning-based prompting baseline used in our experiments.

System Prompt

You are a multimodal question answering assistant. Your task is to answer a given question using the provided data sources: text paragraphs, images (with titles), and tables. You must rely only on this information and not use any external knowledge. If the provided information is insufficient, output `None`.

Before producing the final answer, follow the three phases below:
Phase 1: Strategic Planning
Analyse the question and available data sources, and devise a step-by-step plan. The plan should:

- Identify which data sources are most relevant
- Determine the order in which they should be processed
- Specify how to handle potential conflicts across sources

Phase 2: Execution
Follow the plan to extract and combine information. Every piece of evidence must be explicitly grounded in a source (e.g., table, image title, or paragraph).
Phase 3: Final Answer
Provide a concise answer based on the executed plan. If the required information is not available, output `None`.

User Prompt

```
Question: {question_text}
Images: {images_text}
Text Paragraphs: {paragraphs_text}
Tables: {tables_text}
```

Follow the planning and execution procedure to answer the question.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction match our idea, methodology and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not produce any theoretical results. For assumptions regarding the calculation of logical entropy see Section 2.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: As shown in Section 2 and 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code at the following link EDMA. Data can be found `https://github.com/hannandarryl/ManyModalQA` and `https://github.com/allenai/multimodalqa`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper describes experimental settings and details in Section 3, and in the Appendix in Section C and Section K.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: As shown in Section 3 and in Section I in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As described in Section 3 and in the Appendix in Section E.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: As shown in Appendix J.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit the creators or original owners of assets used in the paper, and respect the license and terms of use explicitly mentioned.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code for this paper can be accessed through EDMA.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper describes how LLMs are used in the methodology in Section 2. Moreover, we relied on LLMs for editing (e.g. grammar, spelling, and word choice).

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.