AIM: Adaptive Interaction in Multi-Agent Debate for Multimodal LLM Inference

Wei Fan
Virginia Tech
fanwei@vt.edu

JinYi Yoon*
Inha University
jinyiyoon@inha.ac.kr

Bo Ji*
Virginia Tech
boji@vt.edu

Abstract

Multi-Agent Debate (MAD) improves reasoning on multimodal tasks, but its accuracy and token efficiency depend on how agents interact during debate. A largely overlooked design choice is which *interaction modalities* agents use to exchange evidence. Existing methods rely on fixed text- or graph-based interaction, which fails to convey fine-grained multimodal evidence and thus degrades accuracy while inflating token cost due to verbose descriptions. To address this, we propose **A**daptive **I**nteraction in **M**ulti-Agent Debate (AIM), an adaptive routing framework that dynamically selects the optimal modality combination for inter-agent interaction in multimodal MAD. Beyond text and graph, AIM introduces *task-dependent modality interaction*, where agents exchange targeted regions of interest from the task-specific multimodal input, such as image regions, audio segments, or spatio-temporal video clips. To determine the best combination for each instance, AIM first generates a structured baseline response and extracts interpretable routing features that capture which evidence the instance requires, how confident the baseline agent is, and how much new evidence each modality would add beyond the baseline response. A lightweight router then uses these routing features to select a modality combination that avoids both *under-fusion* (i.e., agents miss decisive evidence) and *over-fusion* (i.e., redundant modalities inflate token cost) risks. Through extensive experiments across all eight multimodal question-answering (QA) benchmarks, spanning Visual-QA (VQA), Audio-QA, and Video-QA, AIM achieves the highest accuracy. This gain is twofold: combining interaction modalities improves accuracy over the best single-modality baseline by up to 6.4%; while the most accurate fixed combination differs across benchmarks, AIM still performs the best on all benchmarks, achieving additional accuracy improvement of up to 4.2% and reducing token cost by up to 53.4%. Our code is available at https://github.com/Fanwei100/AIM.

1 Introduction

Multi-Agent Systems (MAS) enhance the reasoning capabilities of Large Language Model (LLM)-based agents through collaboration, critique, and iterative refinement [1–5]. Recent work extends MAS to multimodal question-answering (QA) tasks, such as Visual-QA (VQA) [6, 7], Audio-QA [8], and Video-QA [9, 10], where agents jointly reason over language and non-textual inputs (e.g., image, audio, or video). These tasks enable a wide range of real-world applications, such as medical image interpretation [11], speech emotion recognition in mental health monitoring [12], and autonomous driving [13]. Among various MAS paradigms, we focus on Multi-Agent Debate (MAD), a popular approach in which agents iteratively critique and refine each other's responses [14–16]. Despite its effectiveness, multimodal MAD incurs high token cost, as each agent must process multimodal inputs

*Corresponding authors

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

and propagate a growing history over multiple rounds [14, 15]. Therefore, *a key challenge is how to improve the accuracy of multimodal MAD while keeping token costs low.*

A critical but largely overlooked factor driving both accuracy and token cost is the *interaction modality*, the form in which agents exchange evidence during debate. Existing multimodal MAD systems rely primarily on *text-based* interaction, where agents exchange natural-language rationales and critiques [17]. One recent approach explores *graph-based* interaction, which encodes objects and their spatial or semantic relationships using graphs [18]. Despite structural differences, both modalities essentially reduce evidence to text, making them inherently lossy when representing multimodal information (e.g., an image, an audio segment, or a video clip). When a question hinges on a small region of an image, a subtle acoustic cue, or a brief temporal event in a video clip, text or graph descriptions can miss decisive details while adding verbose explanations that increase token cost. This raises a key question: *how should agents select the most appropriate interaction modality for exchanging evidence?* We explore this question motivated by the following three observations.

First, task-dependent modality interaction is more effective than text or graph. Certain task-specific evidence, such as visual details or acoustic cues, is often more faithfully conveyed in its original modality than a textual or graph-based description. Motivated by the finding that providing targeted visual regions to single reasoning agents improves VQA accuracy [19, 20], we introduce *task-dependent modality interaction*, where agents directly exchange targeted Regions of Interest (RoIs) from the original multimodal input (e.g., image, audio, or video). Specifically, agents share image RoIs for VQA, audio segments for Audio-QA, and video spatio-temporal clips for Video-QA. Our preliminary experiments show that task-dependent interaction consistently outperforms both text-only and graph-only interaction across all three task families (see Insight 1).

Second, multimodal interaction provides complementary evidence that improves reasoning. While task-dependent interaction is proven effective, it alone may not capture all the evidence needed for accurate reasoning, since each modality contributes distinct information. For example, text preserves reasoning context, graph highlights structured relations, and task-dependent interaction exposes the original multimodal evidence. We therefore consider *multimodal interaction*, which selects one or more modalities from the set of text, graph, and task-dependent modalities. We observe that combining multiple modalities further improves accuracy and can even reduce token cost by replacing long descriptions with more targeted evidence exchange and requiring fewer debate rounds to reach an answer. However, activating all modalities available is often inefficient, since it introduces redundant evidence and substantially increases token cost without consistent accuracy gain (see Insight 2).

Third, the best modality combination varies across instances, motivating adaptive selection. A fixed modality combination is often insufficient as different questions demand different types of evidence. The best combination depends on the type of question, the demands of reasoning, and the available multimodal cues. These observations motivate *adaptive modality selection* for each instance (see Insight 3). Specifically, adaptive modality selection routes each instance to one modality combination before the debate begins. However, realizing such *modality routing* introduces two key challenges.

(**C1**) *How to efficiently extract informative routing signals from the initial response of a baseline agent?* These signals must capture the question's evidence requirements, the baseline answer's uncertainty, and the value of each modality, without repeated sampling or extra LLM queries.

(**C2**) *How to design a routing mechanism that identifies the optimal modality combination for accurate and efficient inference?* The mechanism must guard against two complementary risks: *(i) under-fusion*: the selected combination misses decisive evidence and induces hallucination; *(ii) over-fusion*: extra modalities add little gain but increase token cost without proportional accuracy improvement.

To address these challenges, we propose **A**daptive **I**nteraction in **M**ulti-Agent Debate (**AIM**), which, to the best of our knowledge, is the first adaptive routing framework that dynamically selects the interaction modality in multimodal MAD. We summarize our main contributions as follows:

First, we present a systematic study of effective inter-agent interaction in multimodal MAD and identify interaction modality as a key design choice. We introduce *task-dependent modality interaction* as an effective modality beyond text and graphs and motivate *adaptive modality selection* by the instance-dependent nature of the optimal modality combination.

Second, we propose AIM to address (**C1**) and (**C2**) (see Fig. 2). For (**C1**), we design a *structured modality-routing prompt* (Phase ①) that elicits routing signals from a single baseline agent, and

extract *interpretable routing features* (Phase ②) that characterize under-fusion and over-fusion risks. For (**C2**), we design an *adaptive modality router* (Phase ③) that selects the modality combination with high accuracy at the lowest token cost per instance. For Audio-QA and Video-QA, we also introduce *temporal compression* to further reduce token cost and improve accuracy.

Third, our evaluation results show that AIM achieves the highest accuracy on all eight benchmarks spanning VQA, Audio-QA, and Video-QA. This gain is twofold: combining interaction modalities improves accuracy over the best single-modality baseline by up to 6.4%; while the most accurate fixed combination differs across benchmarks ($\mathrm{T} + \mathrm{X}$ on four benchmarks, $\mathrm{T} + \mathrm{G} + \mathrm{X}$ on three, and $\mathrm{G} + \mathrm{X}$ on one), AIM still performs the best on all benchmarks, achieving additional accuracy improvement of up to 4.2% and reducing token cost by up to 53.4%.

2 Related Work

We categorize the most relevant literature into two groups and then position our work relative to both.

Multimodal MAD. Multi-agent reasoning has been extended from text-only to multimodal tasks, including VQA [6, 7], Audio-QA [8], Video-QA [9, 10], document understanding [21, 22], medical multimodal VQA [23, 24], and multimodal misinformation detection [25, 26]. Within this broader paradigm, MAD has emerged as a structured reasoning scheme in which agents iteratively critique and refine each other's responses, often improving multimodal reasoning quality [1, 27, 15]. Existing multimodal MAD studies fall into two directions. *Accuracy-driven* methods enrich the debate process to improve answer quality. For example, BDoG introduces graph-based interaction alongside text [18], DART recruits specialized visual tools based on agent disagreement [20], and WISE employs heterogeneous expert debaters [28]. *Efficiency-driven* methods aim to reduce token cost. For instance, CROSS-MODAL MEMORY COMPRESSION converts long debate histories into compact visual representations [29], iMAD selectively triggers debate only when it is beneficial [14], and CONSENSUS replaces iterative debate with single-round interaction [30].

Modalities for Agent Interaction. The effectiveness of multi-agent collaboration depends heavily on the interaction modality through which agents exchange information. Most multimodal systems rely on natural-language transcripts. COMMA benchmarks this text-centric paradigm, showing that text-only interaction is prevalent but struggles to preserve fine-grained multimodal evidence in complex multimodal tasks [17]. To address this, recent work explores alternative interaction modalities. BDoG introduces blueprint graphs for structured spatial and relational grounding in VQA [18], and SPEECHAGENTS shows that native speech and audio signals can function directly as inter-agent messages in audio-centric domains [31]. Beyond text, graph, and audio interaction, a separate line of work explores latent-space interaction. Methods such as INTERLAT [32] and LATENTMAS [33] directly pass hidden representations between agents to reduce information loss. However, latent interaction requires access to the underlying model's hidden states and weights, making it incompatible with black-box APIs (e.g., GPT, Gemini, and Claude). Although text, graph, and audio interactions have been considered, it remains largely unexplored how to effectively exchange original multimodal evidence between agents. Moreover, existing methods typically use a fixed interaction modality in advance rather than dynamically selecting the modality for each instance.

Position of Our Work. Our work addresses a central design problem in multimodal MAD: *How to design inter-agent interaction so that the system achieves high accuracy while remaining token-efficient?* Our work is orthogonal to prior accuracy-driven and efficiency-driven multimodal MAD methods. One can retain these agent designs, debate schedules, and history-compression strategies while adopting our *adaptive modality interaction*. Unlike methods with fixed interaction designs or white-box latent interaction, our framework operates entirely in the prompt space and is compatible with both white-box and black-box LLM agents. We aim to dynamically select the interaction modality for each instance to improve accuracy while remaining token-efficient.

3 Key Insights

The interaction modality affects both accuracy and token cost in multimodal MAD. In this section, we present three key insights that motivate the design of AIM: (i) task-dependent modality interaction

Table 1: Accuracy (%) and average token cost of all seven modality combinations across VQA (GQA), Audio-QA (DCASE), and Video-QA (NextQA) tasks, where T, G, and X denote text-based, graph-based, and task-dependent modality (e.g., image, audio, or video) interactions, respectively.

Task	One Modality						Two Modalities						All	
	T		G		X		T + G		T + X		G + X		T + G + X	
	Acc	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**
VQA	68.9	6,615	60.6	8,921	71.2	6,455	70.8	6,997	**77.6**	**6,060**	76.4	6,612	77.5	7,833
Audio-QA	47.6	**3,772**	44.4	7,145	48.7	4,137	48.0	4,960	**50.9**	3,967	48.9	4,266	49.8	5,196
Video-QA	72.4	**39,814**	74.2	48,647	75.2	100,871	74.4	43,901	76.4	90,193	74.4	97,112	**77.2**	99,765

improves accuracy; (ii) combining modalities further improves both accuracy and token efficiency; (iii) the best modality combination varies across instances, motivating adaptive modality selection.

Insight 1: Task-dependent modality interaction is more effective than text or graph for multimodal MAD. In many multimodal reasoning tasks, decisive evidence resides in a specific part of the original input, such as an image region in VQA, an audio segment in Audio-QA, or a spatio-temporal clip in Video-QA. We propose task-dependent modality interaction, where agents exchange targeted RoIs from the original multimodal input (e.g., image, audio, or video) during debate. As shown in Table 1, task-dependent interaction (X) consistently achieves higher accuracy than text-based (T) and graph-based (G) interaction across all three tasks, revealing that *the original multimodal evidence improves reasoning*. As a VQA sanity check, sharing only raw bounding-box coordinates reduces tokens but causes a substantial accuracy drop, since coordinates do not reveal the visual content inside the region [19]. Thus, the accuracy gain comes from exposing the actual multimodal evidence, not merely pointing to its location. However, the task-dependent modality alone may still miss the evidence that text or graph carries, raising the question of whether combining modalities helps.

Insight 2: Combining modalities provides complementary evidence, but using all modalities is often wasteful. Each modality contributes unique information: text preserves reasoning context, graph highlights structural relationships, and task-dependent modality exposes the original visual, acoustic, or spatio-temporal evidence. Multimodal interaction therefore provides more comprehensive evidence than any single modality. As shown in Table 1, T + X improves over X on GQA, DCASE, and NextQA in both accuracy and token cost. On GQA, accuracy rises from 71.2% to 77.6% while token cost drops from 6,455 to 6,060. Hence, *combining modalities improves both accuracy and token efficiency beyond the task-dependent modality alone.* While multimodal interaction is beneficial, using all modalities at all times may be inefficient. In Table 1, T + X achieves slightly higher accuracy than T + G + X (77.6% vs. 77.5%) and uses 22.6% fewer tokens on GQA. On DCASE, T + X achieves higher accuracy than T + G + X (50.9% vs. 49.8%) at a lower token cost. Therefore, *using all modalities may incur high costs without improving accuracy*. This motivates the need for principled modality-combination selection.

Insight 3: The best modality combination varies across instances, motivating adaptive selection. Although modality-combination selection is essential, no single fixed combination performs best across all instances. To illustrate this, we define the best combination for each instance as the correct one with the lowest token cost, or the lowest-cost one if none is correct. As shown in Fig. 1 for seven VQA question types, the best combination distribution is highly non-uniform: T + X dominates object, attribute, and action questions; G + X dominates spatial and counting questions; knowledge and causal questions spread across multiple combinations. We observe similar instance-dependent patterns in Audio-QA and Video-QA (see Appendix C). These observations suggest selecting the modality combination for each instance based on its evidence requirements.

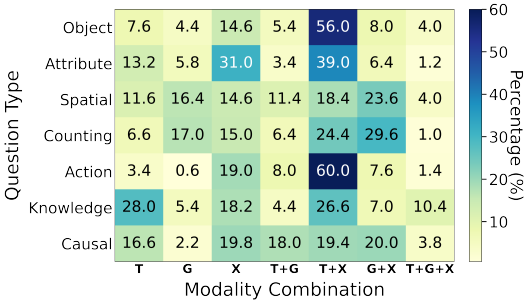


Figure 1: Best modality combination distribution across seven VQA question types.

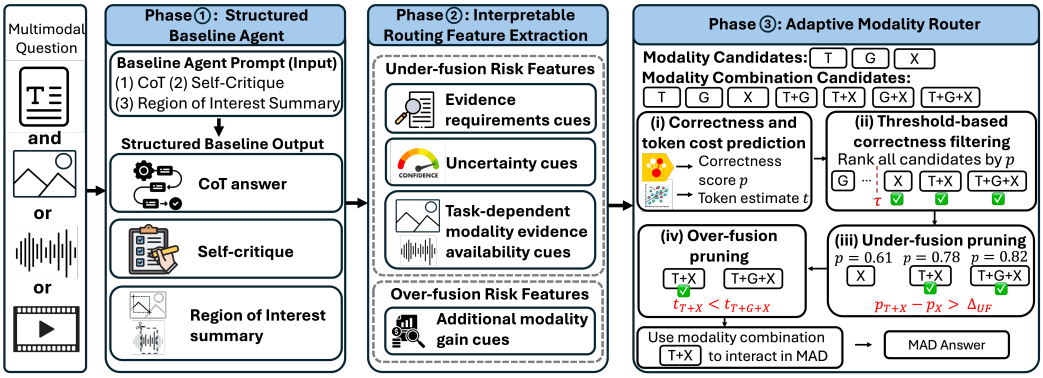


Figure 2: Overview of AIM.

4 Our Design: AIM

In this section, we present AIM, an adaptive routing framework that dynamically selects the best modality combination for inter-agent interaction at the instance level. AIM targets three design goals: *accuracy improvement*, *token efficiency*, and *instance-level adaptivity*. To achieve these goals, the modality selection must guard against two risks. The first is *under-fusion*, where the selected modality combination misses decisive evidence, forcing agents to bridge gaps with unsupported inference that increases hallucination and degrades accuracy. We verify this using FaithScore [34], a widely used hallucination metric (see Section 5.2). As shown in Table 3, single-modality settings yield much lower FaithScores (indicating more hallucination) than richer combinations, suggesting that insufficient modality evidence could be a major source of hallucination in multimodal MAD. The second risk is *over-fusion*, where additional modalities introduce little new evidence while lengthening prompts and debate histories. This would increase token costs without yielding commensurate improvements in accuracy. In Table 1, the full-modality combination often costs more tokens than other combinations without consistently improving accuracy. To avoid both risks, we need to address two challenges: (i) extracting informative routing signals from a single-agent response and (ii) selecting a modality combination that provides sufficient evidence while using necessary modalities only.

As illustrated in Fig. 2, AIM addresses these challenges through three key designs: Phase ①: *structured baseline agent*, which produces an initial answer with compact uncertainty and per-modality evidence cues, informing downstream routing (Section 4.1); Phase ②: *interpretable routing feature extraction*, which extracts routing signals from the baseline response to characterize under-fusion and over-fusion risks (Section 4.2); Phase ③: *adaptive modality router*, which selects the best modality combination and executes the corresponding MAD process (Section 4.3). For Audio-QA and Video-QA, the task-dependent modality has an explicit temporal dimension. Hence, we introduce *temporal compression* to further reduce token cost and potentially improve accuracy by compressing routed audio segments and video region clips during debate (Section 4.4).

4.1 Phase ①: Structured Baseline Agent

Given an input question q with its multimodal input, AIM prompts a single baseline agent to return a structured JSON response with three components. (i) An *initial Chain-of-Thought (CoT) answer with a self-reported confidence score*, where the CoT provides the answer with reasoning and the confidence score quantifies its uncertainty. (ii) A *self-critique with a missing-evidence statement*, stating the strongest reason why the answer could be wrong and what additional information would confirm or refute it. This captures residual uncertainty and the evidence gap that the baseline agent cannot resolve. (iii) Up to K *task-dependent modality summaries*, each describing a candidate region with coordinates normalized to $[0, 1]$ for VQA, a segment with start and end timestamps for Audio-QA, or a spatio-temporal clip with both a spatial bounding box and timestamps for Video-QA. Each region is paired with a brief caption and a confidence score. These components expose routing signals for detecting under-fusion (i.e., missing evidence) and over-fusion (i.e., modality redundancy).

4.2 Phase ②: Interpretable Routing Feature Extraction

From the structured baseline output, we extract interpretable features in four groups aligned with the two risks. The first three groups detect *under-fusion* risk: (i) *evidence requirement cues* include the question type, referenced entities, and missing-evidence statement, which indicate what kind of evidence the question needs, such as spatial/OCR cues for VQA, speech/event cues for Audio-QA, and action/motion cues for Video-QA; (ii) *uncertainty cues* estimate hallucination risk from the confidence score, self-critique, hedge terms, and whether the CoT grounds its answer in instance-specific evidence rather than generic priors; (iii) *task-dependent modality evidence availability cues* estimate whether the missing evidence is likely present in the task-dependent input, using region-level signals such as relevance, confidence, entity coverage, and spatial or temporal distribution. The fourth group reduces *over-fusion* risk: (iv) *additional modality gain cues* estimate how much useful new evidence each modality would add to the baseline, using signals such as coverage of key entities, relationships, events, or temporal context, overlap with the current baseline reasoning, and whether the evidence is question-relevant. We provide the detailed feature list in Appendix B.

4.3 Phase ③: Adaptive Modality Router

For each question q, the router takes the routing feature vector $\mathbf{z}_q \in \mathbb{R}^d$ extracted in Phase ②, where d is the feature dimension, and makes a routing decision $a^*(q)$ over the candidate set of seven options:

$$\mathcal{A} := \{\mathrm{T}, \mathrm{G}, \mathrm{X}, \mathrm{T}+\mathrm{G}, \mathrm{T}+\mathrm{X}, \mathrm{G}+\mathrm{X}, \mathrm{T}+\mathrm{G}+\mathrm{X}\},$$

where T, G, and X denote **T**ext-based, **G**raph-based, and task-dependent modality-based interaction, respectively. Depending on the task, X is image RoIs for VQA, audio segments for Audio-QA, and spatio-temporal clips for Video-QA. A combined option lets agents interact through more than one modality. The router selects one of the seven options in $\mathcal{A}$ rather than deciding on any single modality separately, since the value of one modality depends on which others are used together. For example, two modalities can jointly give more concise evidence with fewer tokens than either alone, while their interaction can improve or hurt accuracy, depending on how the information is combined. To avoid both under-fusion and over-fusion, the router proceeds with four sequential steps as follows.

(i) *Correctness and token-cost prediction*. For each question q and each candidate $a \in \mathcal{A}$, the trained router uses $\mathbf{z}_q$ to predict two quantities: a calibrated correctness probability $p(q, a) \in [0, 1]$ from a binary correctness classifier and an instance-specific token-cost estimate $c(q, a) \in \mathbb{R}^+$ from a regression model, both trained offline on auxiliary datasets. These two estimates capture the likelihood of producing a correct answer and the expected token cost for each candidate.

(ii) *Threshold-based correctness filtering*. The router only retains candidates whose estimated correctness exceeds a tuned threshold $\tau \in [0, 1]$, denoted by

$$\mathcal{C}(q) := \{a \in \mathcal{A} : p(q, a) \geq \tau\}.$$

This removes clearly implausible candidates so that subsequent pruning and cost-based comparison operate on a reduced and more competitive candidate set. If $\mathcal{C}(q) = \emptyset$, meaning that no candidate reaches the correctness threshold, the router falls back to the following:

$$a^{\mathrm{fb}}(q) := \arg\max_{a \in \mathcal{A}} p(q, a),$$

i.e., the modality combination with the highest predicted correctness and thus the most likely to answer correctly among the available options. In this fallback case, the router skips the subsequent steps and directly takes the fallback combination as the final modality routing decision, i.e., $a^*(q) = a^{\mathrm{fb}}(q)$.

(iii) *Under-fusion pruning*. Threshold-based filtering ensures that every candidate in $\mathcal{C}(q)$ is individually plausible, but it does not compare structurally related combinations. In particular, a smaller combination may pass filtering even though adding modalities yields a much higher correctness probability. To correct this, the router compares nested combinations. Let $a \subset s$ denote that s contains all modalities in a and at least one additional modality. The router removes a smaller candidate $a \in \mathcal{C}(q)$ if there exists a larger combination $s \in \mathcal{C}(q)$ with $a \subset s$ such that

$$p(q, s) - p(q, a) > \Delta_{\mathrm{UF}},$$

where $\Delta_{\mathrm{UF}} > 0$ is a tuned under-fusion (UF) margin. A large correctness gap indicates that the smaller combination is likely missing decisive evidence, so it should not remain competitive merely because it uses fewer tokens. The resulting candidate set, denoted by $\mathcal{C}'(q)$, is defined as follows:

$$\mathcal{C}'(q) := \big\{a \in \mathcal{C}(q) : \nexists\, s \in \mathcal{C}(q) \text{ such that } a \subset s \text{ and } p(q,s) - p(q,a) > \Delta_{\mathrm{UF}}\big\}.$$

(iv) *Over-fusion pruning via token-cost-based final selection.* After under-fusion pruning, the remaining candidates in $\mathcal{C}'(q)$ have already passed the correctness screen and are no longer obviously missing critical evidence. The router then prunes over-fused candidates and makes the final decision in a single action, by selecting the candidate with the lowest predicted token cost:

$$a^*(q) := \arg\min_{a \in \mathcal{C}'(q)} c(q,a).$$

If multiple candidates share the same lowest predicted token cost, the router selects the one with the highest predicted correctness probability $p(q,a)$. This step naturally discourages over-fusion: selecting the lowest-cost candidate among the remaining plausible ones prevents the router from including extra modalities that increase token cost without significant accuracy gain. The selected combination $a^*(q)$ is then used to configure the modality interaction for this specific instance q in MAD. The corresponding debate process is executed following the standard MAD procedure.

4.4 Temporal Compression for Task-Dependent Interaction Involving Audio or Video

For Audio-QA and Video-QA, where X has an explicit temporal dimension, AIM further decides *how much* to compress the routed evidence. Since LLM agents tokenize audio and video at fixed rates, compressing routed audio segments or video spatio-temporal clips by a factor $r > 1$ reduces token cost beyond adaptive routing (see Appendix A). Beyond token savings, an appropriate r can also improve accuracy by removing redundant temporal context while preserving core semantic content, helping agents focus on critical evidence. We select r based on the temporal density of the input to avoid under-compression (which wastes tokens) and over-compression (which makes inputs too dense to process and degrades accuracy).

5 Evaluation

We evaluate AIM on eight multimodal QA benchmarks across VQA, Audio-QA, and Video-QA. In the main paper, we compare AIM with all fixed-modality baselines (Section 5.1), analyze its hallucination severity via FaithScore (Section 5.2), and compare AIM with four single-group adaptive routers (Section 5.3). In the appendix, we further provide the feature importance analysis (Appendix B), best combination analysis (Appendix C), latency and cost analyses (Appendix D), ablation studies (Appendix E), cross-LLM-backbone results (Appendix F), and robustness analysis (Appendix G).

Datasets. We train a separate modality router per task family on auxiliary datasets disjoint from all evaluation benchmarks: (i) Visual7W [35] and CLEVR [36] for VQA; (ii) AudioMCQ [37] for Audio-QA; (iii) Video-MME [38] for Video-QA. We use 500 auxiliary instances for VQA and 250 each for Audio-QA and Video-QA. We choose these datasets as they collectively cover diverse question types and modality requirements within each task family.

We evaluate on eight benchmarks: GQA [39], OKVQA [40], VCR [41], and VQA-v2 [42] for VQA; Clotho [43] and DCASE [44] for Audio-QA; IntentQA [45] and NextQA [46] for Video-QA. We do not use these datasets for router training or threshold tuning (see Appendix A for dataset details).

Baselines. We compare AIM against all seven fixed modality interaction configurations. For single-modality interaction, we include: (i) MAD (T) [1]: standard text-only MAD; (ii) Graph (G) [18]: blueprint graph-based interaction; (iii) Task-dependent (X): instantiated as image RoIs for VQA, audio segments for Audio-QA, and spatio-temporal video clips for Video-QA. For two-modality and full-modality interaction, we include: (iv) T + G [18], (v) T + X, (vi) G + X, and (vii) T + G + X.

Metrics. We evaluate each method using three metrics: (i) answer accuracy (**Acc**), the correctness of the final answer after debate; (ii) average token cost (**Tkn**), the total input and output tokens per instance, including both the structured baseline response and the subsequent MAD process (Appendix D.2); (iii) FaithScore [34], assessing hallucination level, where a higher score indicates better-grounded answers. We also report the latency and training cost of AIM (see Appendix D).

Table 2: Accuracy (Acc) and average token cost (Tkn) across eight multimodal QA benchmarks. Accuracy is measured in percentage (%), and tokens denote the average token cost per instance, where K denotes thousands. The highest Acc and the lowest Tkn per dataset are in bold.

Dataset	One Modality						Two Modalities						All		Adaptive	
	T		G		X		T + G		T + X		G + X		T + G + X		AIM	
	Acc	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**
GQA	68.9	6,615	60.6	8,921	71.2	6,455	70.8	6,997	77.6	**6,060**	76.4	6,612	77.5	7,833	**80.1**	6,334
OKVQA	89.8	7,803	89.6	9,871	90.4	7,782	89.8	8,114	93.4	**5,928**	91.2	7,791	93.5	9,773	**93.8**	6,775
VCR	77.6	12,572	78.4	12,588	79.2	12,202	77.6	10,062	79.8	7,869	81.1	10,861	79.6	10,977	**84.2**	**7,788**
VQA-v2	81.0	8,796	77.6	10,583	81.7	7,530	81.0	9,332	84.0	7,012	81.7	8,185	81.1	9,317	**87.7**	**7,006**
Clotho	39.5	**3,692**	41.1	4,103	43.1	4,268	40.2	4,237	43.7	3,897	44.2	3,774	45.1	4,416	**46.8**	3,795
DCASE	47.6	**3,772**	44.4	7,145	48.7	4,137	48.0	4,960	50.9	3,967	48.9	4,266	49.8	5,196	**51.8**	3,891
IntentQA	64.0	**44.3K**	61.1	66.2K	68.4	109.7K	66.8	56.2K	71.2	98.2K	70.0	102.4K	70.8	100.2K	**75.4**	45.8K
NextQA	72.4	**39.8K**	74.2	48.6K	75.2	100.9K	74.4	43.9K	76.4	90.2K	74.4	97.1K	77.2	99.8K	**78.8**	48.6K

Implementation Details. For LLM-based prompting, we use Gemini 2.5 Flash for the primary results and use GPT-5.4-nano for cross-model evaluations (see Appendix F for details). We provide the full MAD protocol, LLM agent configuration, graph interaction details, dataset details, router configurations, feature extraction details, and temporal compression procedures in Appendix A.

5.1 Main Results

We compare the accuracy and token efficiency of AIM against all fixed single-modality, two-modality, and full-modality interaction baselines across eight benchmarks.

Advantage over Fixed Modality Interactions. As shown in Table 2, AIM achieves the highest accuracy on all eight datasets with competitive or even lower token costs in most cases. This gain is twofold. First, single-modality baselines suffer from under-fusion, and the most accurate fixed multimodal combination improves accuracy over the strongest single-modality baseline (X-only) by up to 6.4%. Second, AIM outperforms the most accurate fixed combination on all benchmarks, improving accuracy by up to 4.2%. At the other extreme, the full combination $\mathrm{T} + \mathrm{G} + \mathrm{X}$ suffers from over-fusion, with higher token costs but lower accuracy than AIM across all eight benchmarks. Specifically, AIM reduces token cost by 14.1%-54.3% alongside accuracy improvements.

Critically, we find that *no single fixed-modality combination is universally dominant*. The combination $\mathrm{T} + \mathrm{X}$ achieves the best fixed-combination accuracy on GQA (77.6%), VQA-v2 (84.0%), DCASE (50.9%), and IntentQA (71.2%), while $\mathrm{G} + \mathrm{X}$ leads on VCR (81.1%), and $\mathrm{T} + \mathrm{G} + \mathrm{X}$ leads on OKVQA (93.5%), Clotho (45.1%), and NextQA (77.2%). AIM outperforms each of these dataset-specific best combinations, demonstrating its ability to adapt the modality combination to the evidence requirements of each instance rather than relying on a fixed choice. These gains follow from the accuracy-first objective of AIM, which prioritizes answer accuracy and then minimizes token cost among the combinations expected to answer correctly. For example, on OKVQA, the most accurate fixed combination is $\mathrm{T} + \mathrm{G} + \mathrm{X}$ with 93.5% accuracy and 9,773 tokens. AIM reaches a higher accuracy of 93.8% while using only 6,775 tokens (30.7% fewer). These results remain consistent across three inference runs and five independent router retraining runs with different random seeds.

Cross-Task Analysis. While under-fusion and over-fusion occur consistently across all three task families, the token cost of the task-dependent configuration X and the relative contribution of each modality differ by task. The token cost of X varies substantially, from comparable to T in VQA (e.g., 6,455 for X vs. 6,615 for T on GQA) and Audio-QA (e.g., 4,268 for X vs. 3,692 for T on Clotho), to over 100K tokens per instance in Video-QA (e.g., 109,732 on IntentQA). This cost structure motivates how AIM selects the modality combination. In VQA and Audio-QA, where X provides decisive evidence at reasonable token cost, AIM frequently includes X. In Video-QA, X remains valuable for accuracy (e.g., 68.4% for X-only vs. 64.0% for T-only on IntentQA), but its token cost is much higher. Therefore, AIM includes X only when temporal evidence is decisive to achieve the higher accuracy while reducing token cost by over 50% compared to $\mathrm{T} + \mathrm{G} + \mathrm{X}$. Additionally, the contribution of G is also task-dependent: while G-only is the weakest baseline in VQA, it performs

Table 3: FaithScore comparison across eight multimodal QA benchmarks. Higher FaithScore indicates better-grounded answers with less hallucination.

Dataset	One Modality			Two Modalities			All	Adaptive
	T	G	X	T + G	T + X	G + X	T + G + X	AIM
GQA	0.61	0.50	0.64	0.64	0.71	0.66	0.69	**0.72**
OKVQA	0.67	0.64	0.76	0.78	0.80	0.76	0.80	**0.81**
VCR	0.63	0.66	0.66	0.68	0.68	**0.75**	**0.75**	**0.75**
VQA-v2	0.71	0.63	0.74	0.77	**0.78**	0.74	0.77	0.77
Clotho	0.76	0.69	0.77	0.78	0.82	**0.87**	0.81	0.84
DCASE	0.67	0.64	0.67	0.68	**0.70**	0.68	0.69	**0.70**
IntentQA	0.82	0.81	0.83	**0.84**	**0.84**	0.83	0.82	**0.84**
NextQA	0.84	0.82	0.86	0.87	0.86	**0.88**	**0.88**	**0.88**

more competitively in Audio-QA. These results highlight that none of the modalities is universally effective or redundant across tasks, and thus our instance-level adaptive routing is necessary.

5.2 FaithScore Analysis

To evaluate the severity of under-fusion across modality configurations, we measure FaithScore, which quantifies how well each claim in a generated answer is grounded in the input evidence. The higher scores indicate fewer hallucinated claims [34]. We use FaithScore to detect under-fusion, since insufficient modality evidence forces agents to fill evidence gaps with ungrounded reasoning, which lowers the score. As shown in Table 3, on GQA, the FaithScore of G and T is 0.50 and 0.61, respectively, far below the 0.72 of AIM. This indicates that the agents lack grounding and hallucinate, which degrades accuracy. However, FaithScore penalizes any claim not directly grounded in the input, including correct inferences from commonsense or causal reasoning. Sufficient grounding is necessary to avoid harmful hallucination, but maximizing FaithScore beyond that level is undesirable, as it discourages productive inference [47]. This explains why AIM achieves a FaithScore of 0.77 on VQA-v2, marginally below T + X (0.78), yet obtains much higher accuracy (87.7% vs. 84.0%).

FaithScore distributions also vary across task families. VQA and Audio-QA vary widely across combinations (e.g., 0.50-0.72 on GQA and 0.69-0.87 on Clotho). In other words, on these tasks, which modality combination the agents use largely decides how well their answers are grounded in the evidence. Video-QA shows uniformly high scores (e.g., 0.82-0.88 on NextQA), since video combines visual, temporal, and audio information, leaving less room for hallucination. Overall, AIM achieves the highest or tied-highest FaithScore on six of the eight datasets, consistently avoiding under-fusion. On the other two datasets, AIM's FaithScore is marginally below the per-dataset best but yields higher accuracy, reflecting productive inference rather than evidence insufficiency. We provide case studies where FaithScore penalizes correct inferences of AIM (see Appendix I).

5.3 Comparison with Single-group Adaptive Routers

To evaluate whether all four groups of routing features are necessary for routing, we compare AIM with four single-group adaptive routers, each of which selects the modality combination from only one type of routing feature. Question-type routing decides from what the question asks, confidence-only routing from how confident the baseline agent is in the initial answer, missing-evidence-only routing from what evidence the baseline agent reports missing and whether the task-dependent input contains this evidence, and the cost-aware heuristic from how much new evidence each modality would add. Therefore, each single-group adaptive router uses only part of the routing features, whereas AIM uses all four of them. We provide the exact routing features of each single-group adaptive router in Appendix H.1. For a fair comparison, all routers share the same auxiliary training datasets, LLM backbone, candidate modality combinations, and MAD protocol.

Comparing Table 4 with Table 2, we find that all four single-group adaptive routers achieve higher average accuracy over the eight benchmarks than the best fixed combination T + X, by 1.0% on average. Hence, adaptive modality selection is beneficial even with one type of routing feature.

Table 4: Accuracy (Acc) and average token cost (Tkn) of AIM and four single-group adaptive routers, each of which uses only one type of routing feature. Accuracy is reported as a percentage (%), and Tkn denotes the average per-instance token cost, where K denotes thousands.

Dataset	Question Type		Confidence		Missing Evidence		Cost-Aware		AIM	
	Acc	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**
GQA	78.9	6,508	78.3	6,634	78.7	6,673	78.1	**6,102**	**80.1**	6,334
OKVQA	93.5	6,897	93.2	6,767	93.4	6,994	93.5	**6,433**	**93.8**	6,775
VCR	83.7	8,417	82.1	9,332	82.4	8,937	81.6	7,943	**84.2**	**7,788**
VQA-v2	86.4	7,570	85.2	8,434	86.1	8,892	85.6	7,789	**87.7**	**7,006**
Clotho	46.2	3,891	45.7	**3,706**	46.0	3,809	44.1	3,778	**46.8**	3,795
DCASE	49.7	3,987	50.9	4,382	51.1	4,208	50.5	3,902	**51.8**	**3,891**
IntentQA	72.2	55.9K	71.0	45.8K	71.6	53.7K	70.2	**44.9K**	**75.4**	45.8K
NextQA	78.7	49.1K	77.9	48.6K	**78.8**	52.5K	74.9	**44.7K**	**78.8**	48.6K

However, one type of routing feature is still insufficient, since AIM achieves higher or equal accuracy than every single-group adaptive router on all eight benchmarks. Even compared with question-type routing, the most accurate single-group adaptive router, AIM improves the average accuracy by 1.16% and reduces the average token cost by 8.6%. Moreover, the cost-aware heuristic uses fewer tokens than AIM on five benchmarks but results in 2.51% lower average accuracy, which indicates under-fusion. Therefore, the accuracy of AIM comes from using all four groups of routing features rather than from any single group. We further evaluate the robustness of AIM and the four single-group adaptive routers under three types of routing-feature perturbations: flipping, masking, and shifting (see Appendix H.2). Specifically, we compare the resulting accuracy degradation across the methods.

6 Discussion

Although AIM improves accuracy while remaining token-efficient across VQA, Audio-QA, and Video-QA, it requires a one-time offline training phase per task family. Specifically, training the modality router requires collecting outcomes from all seven modality combinations on auxiliary datasets, which runs the full MAD pipeline seven times per training instance. Despite this upfront cost, the resulting router is lightweight and generalizes to unseen evaluation datasets without additional data collection. This cost could be further reduced through online adaptation.

AIM can also be extended in two directions. First, adaptive modality interaction may benefit broader multi-agent paradigms beyond MAD, such as collaborative software development [2, 48] and multi-agent document understanding [21], where fixed interaction modalities may also cause under-fusion or over-fusion. Second, since current LLM APIs count tokens based on both video duration and frame resolution, jointly optimizing temporal sampling and spatial compression alongside modality selection could further reduce token cost without sacrificing accuracy.

7 Conclusion

We presented AIM, an adaptive multimodal MAD framework that selects the best interaction modality combination per instance to improve reasoning accuracy while remaining token-efficient. By generating a structured baseline response, extracting interpretable routing features, and applying a lightweight router, AIM avoids both the risks of under-fusion and over-fusion, and thus provides agents with decisive evidence without redundant modalities. Across eight benchmarks spanning three multimodal task families, AIM achieves the highest accuracy. Combining modalities improves accuracy by up to 6.4% over the best single-modality baseline, and adapting the combination per instance adds up to 4.2% over the most accurate fixed combination of each benchmark with up to 53.4% fewer tokens. Future work will extend adaptive modality interaction to broader multi-agent paradigms and jointly optimize temporal and spatial compression for video.

Acknowledgments

We sincerely thank the anonymous reviewers of NeurIPS 2026 for their constructive comments and suggestions. This research was supported in part by the Commonwealth Cyber Initiative (CCI) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2026-25602748).

References

[1] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.992. URL https://aclanthology.org/2024.emnlp-main.992/.

[2] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

[3] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W White, Doug Burger, and Chi Wang. Autogen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

[4] Arne Tillmann. Literature review of multi-agent debate for problem-solving. *arXiv preprint arXiv:2506.00066*, 2025.

[5] Chan-Jan Hsu, Davide Buffelli, Jamie McGowan, Feng-Ting Liao, Yi-Chang Chen, Sattar Vakili, and Da-shan Shiu. Group think: Multiple concurrent reasoning agents collaborating at token level granularity. *arXiv preprint arXiv:2505.11107*, 2025.

[6] Bowen Jiang, Zhijun Zhuang, Shreyas S. Shivakumar, Dan Roth, and Camillo J. Taylor. Multi-Agent VQA: Exploring Multi-Agent Foundation Models in Zero-Shot Visual Question Answering, 2024. URL https://cogcomp.seas.upenn.edu/papers/JZSRT24.pdf.

[7] Zhongjian Hu, Peng Yang, Bing Li, and Zhenqi Wang. Multi-agents based on large language models for knowledge-based visual question answering. *ArXiv*, abs/2412.18351, 2024. URL https://api.semanticscholar.org/CorpusID:274992537.

[8] Xie Zhifei, Mingbao Lin, Zihang Liu, Pengcheng Wu, Shuicheng Yan, and Chunyan Miao. Audio-reasoner: Improving reasoning capability in large audio language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23829–23851, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1216. URL https://aclanthology.org/2025.emnlp-main.1216/.

[9] Abhi Kamboj, Gaurav Kumar, Krista Holden, Madhumitha Saravanan, and Pradyumna Narayana. A multi-agent reasoning framework for video question answering. In *Workshop on Scaling Environments for Agents*, 2025. URL https://openreview.net/forum?id=nBp1omOQ4P.

[10] Noriyuki Kugo, Xiang Li, Zixin Li, Ashish Gupta, Arpandeep Khatua, Nidhish Jain, Chaitanya Patel, Yuta Kyuragi, Yasunori Ishii, Masamoto Tanabiki, Kazuki Kozuka, and Ehsan Adeli. Videomultiagents: A multi-agent framework for video question answering, 2025. URL https://arxiv.org/abs/2504.20091.

[11] Zhihong Lin, Donghao Zhang, Qingyi Tao, Danli Shi, Gholamreza Haffari, Qi Wu, Mingguang He, and Zongyuan Ge. Medical visual question answering: A survey. *Artificial Intelligence in Medicine*, 143:102611, 2023.

[12] Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, Junaid Qadir, and Björn Schuller. Survey of deep representation learning for speech emotion recognition. *IEEE Transactions on Affective Computing*, 14(2):1634–1654, 2023. doi: 10.1109/TAFFC.2021.3114365.

[13] Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Visual question answering for autonomous driving. In *European Conference on Computer Vision*, pages 252–269. Springer, 2024.

[14] Wei Fan, JinYi Yoon, and Bo Ji. imad: Intelligent multi-agent debate for efficient and accurate llm inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29403–29411, 2026.

[15] Ao Li, Jinghui Zhang, Luyu Li, Yuxiang Duan, Lang Gao, Mingcai Chen, Weijun Qin, Shaopeng Li, Fengxian Ji, Ning Liu, et al. M3mad-bench: Are multi-agent debates really effective across domains and modalities? *arXiv preprint arXiv:2601.02854*, 2026.

[16] Ashutosh Adhikari and Mirella Lapata. Debating for better reasoning in vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15766–15784, 2025.

[17] Timothy Ossowski, Danyal Maqbool, Jixuan Chen, Zefan Cai, Tyler Bradshaw, and Junjie Hu. Comma: A communicative multimodal multi-agent benchmark. *arXiv preprint arXiv:2410.07553*, 2024.

[18] Changmeng Zheng, Dayong Liang, Wengyu Zhang, Xiao-Yong Wei, Tat-Seng Chua, and Qing Li. A picture is worth a graph: A blueprint debate paradigm for multimodal reasoning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 419–428, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706868. doi: 10.1145/3664647.3681102. URL https://doi.org/10.1145/3664647.3681102.

[19] Xin He, Longhui Wei, Lingxi Xie, and Qi Tian. Incorporating visual experts to resolve the information loss in multimodal large language models. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/123. URL https://doi.org/10.24963/ijcai.2025/123.

[20] Nithin Sivakumaran, Justin Chen, David Wan, Yue Zhang, Jaehong Yoon, Elias Stengel-Eskin, and Mohit Bansal. DART: Leveraging multi-agent disagreement for tool recruitment in multimodal reasoning. In Vera Demberg, Kentaro Inui, and Lluís Marquez, editors, *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5445–5464, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. URL https://aclanthology.org/2026.eacl-long.253/.

[21] Siwei Han, Peng Xia, Ruiyi Zhang, Tong Sun, Yun Li, Hongtu Zhu, and Huaxiu Yao. Mdocagent: A multi-modal multi-agent framework for document understanding. *arXiv preprint arXiv:2503.13964*, 2025.

[22] Aymen Lassoued, Mohamed Ali Souibgui, and Yousri Kessentini. Orca: Orchestrated reasoning with collaborative agents for document visual question answering. *arXiv preprint arXiv:2603.02438*, 2026.

[23] Karishma Thakrar, Shreyas Basavatia, and Akshay Daftardar. Architecting clinical collaboration: Multi-agent reasoning systems for multimodal medical vqa, 2025. URL https://arxiv.org/abs/2507.05520.

[24] Peng Xia, Jinglu Wang, Yibo Peng, Kaide Zeng, Zihan Dong, Xian Wu, Xiangru Tang, Hongtu Zhu, Yun Li, Linjun Zhang, Shujie LIU, Yan Lu, and Huaxiu Yao. MMedagent-RL: Optimizing multi-agent collaboration for multimodal medical reasoning. In *The Fourteenth International*

Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=2awntLXwR6.

[25] Kumud Lakara, Georgia Channing, Juil Sock, Christian Rupprecht, Philip Torr, John Collomosse, and Christian Schroeder de Witt. MAD-sherlock: Multi-agent debate for visual misinformation detection. In *ICML 2025 Workshop on Collaborative and Federated Agentic Workflows*, 2025. URL https://openreview.net/forum?id=3Y3SV2LOlj.

[26] Shuo Yang, Zijian Yu, Zhenzhe Ying, Yuqin Dai, Guoqing Wang, Jun Lan, Jinfeng Xu, Jinze Li, and Edith CH Ngai. Rama: Retrieval-augmented multi-agent framework for misinformation detection in multimodal fact-checking. In *Companion Proceedings of the ACM Web Conference 2026*, pages 416–424, 2026.

[27] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FQepisCUWu.

[28] Anoop Cherian, Suhas Lohit, and Kuan-Chuan Peng. WISE: Weighted iterative society-of-experts for multimodal multi-agent debate with probabilistic consensus. In *ICML 2026 Workshop on Multimodal AI Agents*, 2026. URL https://openreview.net/forum?id=La8oTIGZoM.

[29] Jing Wu, Yue Sun, Tianpei Xie, Suiyao Chen, Jingyuan Bao, Yaopengxiao Xu, Gaoyuan Du, Inseok Heo, Alexander Gutfraind, and Xin Wang. Cross-modal memory compression for efficient multi-agent debate. *arXiv preprint arXiv:2602.00454*, 2026.

[30] Hyungjun Yoon, Mohammad Malekzadeh, Sung-Ju Lee, Fahim Kawsar, and Lorena Qendro. Consensus: Multi-agent collaboration for multimodal sensing. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 32950–32969, 2026.

[31] Dong Zhang, Zhaowei Li, Pengyu Wang, Xin Zhang, Yaqian Zhou, and Xipeng Qiu. Speechagents: Human-communication simulation with multi-modal multi-agent systems. *arXiv preprint arXiv:2401.03945*, 2024.

[32] Zhuoyun Du, Runze Wang, Huiyu Bai, Zouying Cao, Xiaoyong Zhu, Yu Cheng, Bo Zheng, Wei Chen, and Haochao Ying. Enabling agents to communicate entirely in latent space. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27106–27129, 2026.

[33] Jiaru Zou, Ruizhong Qiu, Gaotang Li, Xiyuan Yang, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, James Zou, Mengdi Wang, and Ling Yang. Latent collaboration in multi-agent systems. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 168813–168847. PMLR, 06–11 Jul 2026. URL https://proceedings.mlr.press/v306/zou26k.html.

[34] Liqiang Jing, Ruosen Li, Yunmo Chen, and Xinya Du. FaithScore: Fine-grained evaluations of hallucinations in large vision-language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5042–5063, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.290. URL https://aclanthology.org/2024.findings-emnlp.290/.

[35] Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4995–5004, 2016. doi: 10.1109/CVPR.2016.540.

[36] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997, 2017. doi: 10.1109/CVPR.2017.215.

[37] Haolin He, Xingjian Du, Renhe Sun, Zheqi Dai, Yujia Xiao, Mingru Yang, Jiayi Zhou, Xiquan Li, Zhengxi Liu, Zining Liang, et al. Measuring audio's impact on correctness: Audio-contribution-aware post-training of large audio language models. In *International Conference on Learning Representations*, volume 2026, pages 110411–110432, 2026.

[38] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24108–24118, 2025.

[39] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019. doi: 10.1109/CVPR.2019.00686.

[40] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019.

[41] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[42] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[43] Samuel Lipping, Parthasaarathy Sudarsanam, Konstantinos Drossos, and Tuomas Virtanen. Clotho-aqa: A crowdsourced dataset for audio question answering. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 1140–1144. IEEE, 2022.

[44] Haolin He, Mingru Yang, Renhe Sun, Jiayi Zhou, Jian Liu, Qianhua He, and Qiuqiang Kong. Audio question answering at the dcase 2025 challenge.

[45] Jiapeng Li, Ping Wei, Wenjuan Han, and Lifeng Fan. Intentqa: Context-aware video intent reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11963–11974, 2023.

[46] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021.

[47] Chenxu Yang, Qingyi Si, Lanrui Wang, and Zheng Lin. Breaking the trade-off between faithfulness and expressiveness for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34232–34240, 2026.

[48] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.810. URL `https://aclanthology.org/2024.acl-long.810/`.

[49] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[50] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, 95:102026, 2022. ISSN 0895-6111. doi: https://doi.org/10.1016/j.compmedimag.2021.102026. URL `https://www.sciencedirect.com/science/article/pii/S0895611121001750`.

[51] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[52] Deepak Gupta, Utku Kose, Ashish Khanna, and Valentina Emilia Balas. *Deep learning for medical applications with unique data*. Academic Press, 2022.

[53] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[54] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL http://arxiv.org/abs/1908.10084.

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

[56] Nema Salem and Sahar Hussein. Data dimensional reduction and principal components analysis. *Procedia Computer Science*, 163:292–299, 2019. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2019.12.111. URL https://www.sciencedirect.com/science/article/pii/S1877050919321507. 16th Learning and Technology Conference 2019Artificial Intelligence and Machine Learning: Embedding the Intelligence.

[57] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

[58] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 203–213, 2020.

[59] Ralf Gommers, Pauli Virtanen, Matt Haberland, Evgeni Burovski, Tyler Reddy, Warren Weckesser, Travis E Oliphant, David Cournapeau, Andrew Nelson, Pamphile Roy, et al. scipy/scipy: Scipy 1.15. 0. *Zenodo*, 2024.

[60] Jan Newmarch. Ffmpeg/libav. In *Linux sound programming*, pages 227–234. Springer, 2017.

[61] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[62] {Scott M.} Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, {Jordan M.} Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and {Su In} Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):56–67, January 2020. ISSN 2522-5839. doi: 10.1038/s42256-019-0138-9. Publisher Copyright: © 2020, The Author(s), under exclusive licence to Springer Nature Limited.

Appendix

In the appendix, we include implementation details, additional analyses, and additional evaluation results. First, we provide the implementation details in Appendix A and describe the routing features with the SHAP-based feature importance analysis in Appendix B. Then, we extend the best combination analysis to Audio-QA and Video-QA in Appendix C and analyze the latency and cost of AIM in Appendix D. Next, we report the ablation studies in Appendix E, the cross-backbone results using GPT-5.4-nano in Appendix F, the robustness analysis in Appendix G, and the comparison with single-group adaptive routers in Appendix H. Lastly, we provide FaithScore case studies in Appendix I.

A Implementation Details

In this section, we provide the implementation details for our experiments, including LLM agent configuration, the MAD protocol, graph interaction details, dataset descriptions, router configurations, feature extraction settings, temporal compression procedures, and result stability analysis.

LLM Agent Configuration. For LLM agents, we use a temperature of 0.0 to reduce randomness in the outputs and a maximum of 512 tokens. The structured baseline agent proposes up to $K{=}3$ task-dependent modality summaries per instance. All compared methods share the same backbone, graph pipeline, crop tool, and debate protocol, varying only the interaction modality combination.

MAD Protocol. All methods use the same MAD protocol with three agents in affirmative, negative, and judge roles for at most five rounds. The affirmative and negative agents present and critique candidate answers. The debate stops when the judge reaches a conclusion. Otherwise, the judge must produce a final answer at the fifth round. The judge's answer is used for evaluation. This protocol is shared by AIM and all fixed-modality baselines.

Graph Interaction. Graph interaction follows the blueprint graph design [18]. AIM and all fixed baselines use the same graph construction, pruning settings, and graph representation. The graph provides structured evidence. The original blueprint graph design supports only VQA, where the graph is built from entities and relations in the image. We extend the graph modality to Audio-QA and Video-QA by building the graph from sound events in the audio and from visible events or actions in the video, respectively. For Audio-QA and Video-QA, the prompts serialize each graph as a JSON object with an `events` list and a `relationships` list. Each event has an identifier, a description, and attributes, where an event describes a sound event for Audio-QA and a visible event or action for Video-QA. Each relationship links two event identifiers and specifies the relation between the events.

Dataset Details. We evaluate on eight benchmarks across three task families. For VQA, GQA [39] evaluates compositional visual reasoning over real-world images using scene-graph annotations, OKVQA [40] requires external knowledge, VCR [41] evaluates visual commonsense reasoning over movie scenes, and VQA-v2 [42] contains natural image questions with balanced answer distributions. For Audio-QA, Clotho [43] contains open-ended questions over environmental sound recordings, and DCASE [44] focuses on acoustic event detection recordings. For Video-QA, IntentQA [45] evaluates intent and causal reasoning over daily-life videos, and NextQA [46] evaluates temporal and causal dynamics in short video clips. Each router is trained on auxiliary instances disjoint from these benchmarks, with 500 instances from Visual7W [35] and CLEVR [36] for VQA, 250 from AudioMCQ [37] for Audio-QA, and 250 from Video-MME [38] for Video-QA.

Router Configurations. The adaptive modality router comprises a correctness classifier and a token-cost regressor, with task-specific configurations for each task family. For VQA, the correctness classifier is a CatBoost [49] model trained with binary cross-entropy [50] loss, using 380 estimators, a maximum tree depth of 4, and a learning rate of 0.03. The token-cost regressor is a LightGBM [51] model trained with root mean squared error [52] loss, using 330 estimators, a maximum tree depth of 4, and a learning rate of 0.03. The acceptance threshold is $\tau = 0.52$, and the under-fusion pruning margin is $\Delta_{\mathrm{UF}} = 0.08$. For Audio-QA and Video-QA, both the correctness classifier and token-cost regressor are XGBoost [53] models. The classifier uses 300 estimators, a maximum tree depth of 6, and a learning rate of 0.05; the regressor uses 300 estimators, a maximum tree depth of 5, and a

learning rate of 0.05. Meanwhile, the acceptance threshold is $\tau = 0.55$, and the under-fusion pruning margin is $\Delta_{\mathrm{UF}} = 0.10$. These hyperparameters are selected via grid search on the validation set of the training data and remain fixed across all evaluation benchmarks.

Feature Extraction Details. We extract 109 routing features for VQA, 98 for Audio-QA, and 100 for Video-QA. For feature extraction, we use all-MiniLM-L6-v2 [54] for text embeddings and CLIP ViT-B/32 [55] for image embeddings, and each text or image representation is reduced to 32 dimensions by PCA [56]. For VQA, we embed the full image and each of the up to three proposed regions. We then aggregate the region embeddings into two representations, one by taking the mean over the regions and one by taking the maximum over the regions in each dimension, and compare each representation with the full image embedding to obtain the image vs. crop-mean and image vs. crop-max similarity features. For Audio-QA, we use wav2vec2-base [57] to encode the full recording and up to three proposed segments at 16 kHz, and a shared PCA projection reduces each representation to 16 dimensions. For Video-QA, we use X3D-M [58] to encode the video and reduce each representation to 32 dimensions by PCA. We cap concatenated text at 600 words and use batch sizes of 64, 32, and 4 for text, image, and audio encoding, respectively. These embeddings are combined with the numerical and categorical routing features, and therefore the number of routing features differs from the input dimension of the correctness classifier and the token-cost regressor. The router models are trained on one NVIDIA RTX 4090 GPU.

Temporal Compression. For temporal compression, we apply the following procedure to compress routed evidence. For Audio-QA, AIM compresses the selected audio segment through SciPy-based audio resampling [59]. For Video-QA, AIM compresses the visual stream through FFmpeg-based frame subsampling [60], compresses the accompanying audio track through SciPy-based audio resampling, and then recombines the two streams into a synchronized clip. The compression ratio r is selected separately for each evaluation dataset: we use $r = 2.5$ for Clotho and IntentQA, and $r = 1$ (no compression) for DCASE and NextQA, reflecting that the latter two datasets contain content that is already sufficiently dense to leave no room for further compression.

Result Stability. To evaluate the stability of the results of AIM, we consider two sources of variation on all eight benchmarks. First, we repeat the end-to-end evaluation three times with the same trained router and decoding settings, where each run includes the structured baseline LLM response, feature extraction, router selection, and final MAD execution. Since feature extraction and routing are deterministic given fixed LLM outputs, the variation across the three runs stems from minor nondeterminism in LLM API outputs even at temperature 0.0. The results of AIM in Table 2 are the mean over these three runs. Second, we retrain the router of each task family five times with different random seeds while keeping the auxiliary training data, routing features, hyperparameters, and LLM outputs fixed. From Table 5, we find that the results of AIM are stable under both sources of variation. Across the three inference runs, the standard deviation of accuracy is at most 0.16% and the relative standard deviation of token cost is at most 3.78%. Similarly, across the five retrainings, the standard deviation of accuracy is at most 0.35% and the relative standard deviation of token cost is at most 4.74%. Moreover, the mean accuracy over the five retrainings remains above the most accurate fixed combination on all eight benchmarks. Therefore, the conclusions in Section 5.1 do not depend on one particular inference run or one particular random seed.

B Feature Details and SHAP-based Feature Importance Analysis

This section provides a comprehensive inventory of the routing features used by the modality router across all three task families and validates their contributions through SHapley Additive exPlanations (SHAP) [61] analysis. The VQA router uses 109 features, the Audio-QA router uses 98 features, and the Video-QA router uses 100 features, reflecting differences in how each task-dependent modality X is represented in the feature space. Although the four-group organization is shared across all three task families, the third group adapts to the task-dependent modality X (image regions for VQA, audio segments for Audio-QA, spatio-temporal clips for Video-QA), and the fourth group adjusts the modality-specific gain and similarity features accordingly. We first detail how the features are organized into four groups aligned with the under-fusion and over-fusion routing risks introduced in Section 4. We then present per-feature SHAP importance results for both the correctness classifier

Table 5: Stability of AIM across three inference runs with the same trained router and across retraining the router with five different random seeds. Each cell reports the mean and one standard deviation.

Dataset	Inference (3 runs)		Retraining (5 seeds)	
	Acc	Tkn	Acc	Tkn
GQA	80.1 ± 0.16	$6{,}334 \pm 96$	80.3 ± 0.35	$6{,}291 \pm 183$
OKVQA	93.8 ± 0.03	$6{,}775 \pm 87$	93.6 ± 0.21	$6{,}832 \pm 149$
VCR	84.2 ± 0.08	$7{,}788 \pm 161$	83.9 ± 0.33	$7{,}843 \pm 281$
VQA-v2	87.7 ± 0.07	$7{,}006 \pm 122$	87.7 ± 0.15	$7{,}112 \pm 282$
Clotho	46.8 ± 0.08	$3{,}795 \pm 45$	46.9 ± 0.24	$3{,}701 \pm 91$
DCASE	51.8 ± 0.09	$3{,}891 \pm 36$	51.5 ± 0.13	$3{,}848 \pm 53$
IntentQA	75.4 ± 0.06	$45{,}752 \pm 1{,}010$	75.2 ± 0.09	$46{,}053 \pm 1{,}880$
NextQA	78.8 ± 0.05	$48{,}636 \pm 1{,}840$	79.0 ± 0.11	$47{,}658 \pm 2{,}260$

and the token-cost regressor, confirming that all four feature groups contribute meaningfully and that no individual feature can be removed without losing useful routing signal.

B.1 Router Feature Design

Feature Derivation. The routing features are derived from the question and the fields of the structured baseline output in two steps, which are shared across the three task families and require no additional LLM query and no evaluation labels. First, each feature group is derived from specific fields. The evidence requirement cues are derived from the question and the missing-evidence statement, and the uncertainty cues are derived from the initial answer, the confidence score, and the self-critique. Similarly, the evidence availability cues are derived from the candidate regions, which are image RoIs for VQA, audio segments for Audio-QA, and video clips for Video-QA, and the additional modality gain cues are derived from the overlap between each candidate modality and the structured baseline output. Second, each field is expanded by a small set of fixed operators. Numeric fields such as confidence and relevance scores are used directly, list fields such as target nouns are expanded into the list length and the leading elements, and lexical cues such as hedge terms and generic priors are matched into binary flags and counts. Moreover, each of the up to three candidate regions keeps separate features, and the relevance scores of the candidate regions are aggregated into the mean, standard deviation, minimum, maximum, the fraction of low-relevance regions, and a bucketed distribution. Finally, semantic overlap is measured by the cosine similarity between embeddings, and free-text fields are embedded and reduced by PCA. All derived features are kept without feature selection, since every feature contributes to at least one of the two router models in the SHAP analysis in Appendix B.2.

B.1.1 VQA Features (109 Features)

Table 6 lists all 109 VQA features organized into four groups. The first three groups primarily detect *under-fusion*, where the selected modality subset misses decisive evidence needed for a correct answer. The fourth group primarily reduces *over-fusion*, where additional modalities contribute little useful evidence while unnecessarily increasing token cost.

Evidence requirement cues (20 features) characterize the question's information needs by combining the question type and referenced entities with the baseline's missing-evidence statement. These features encode the extracted target nouns and their count, the question demand flags (e.g., whether answering requires spatial reasoning, OCR, counting, or interaction understanding), and a text statement of what visual evidence would confirm the hypothesis. They reveal the likely modality demand for each instance.

Uncertainty cues (16 features) capture how well the baseline answer is supported. The core signals are the confidence score and the router's calibrated version of it, along with a calibration flag that labels each instance as overconfident, underconfident, or well-calibrated. Diagnostic signals detect hedge terms (e.g., *maybe*, *possibly*), generic priors (e.g., *usually*, *typically*), and whether the reasoning refers instance-specific visual evidence rather than relying on general knowledge. A self-critique field

provides a counter-argument against the initial hypothesis. High uncertainty or reliance on generic priors can indicate hallucination risk, signaling that additional modalities may be needed.

Visual evidence availability cues (52 features) estimate whether the missing evidence is likely present in the image. For each of up to three proposed image RoIs (crops), the features include the normalized bounding box coordinates, a confidence score that the crop contains decisive evidence, caption relevance to the question, and whether the caption mentions the target nouns. Aggregated statistics summarize the crop-level signals: mean, standard deviation, minimum, and maximum caption relevance; the fraction of low-relevance crops; the variance of crop confidence; and a relevance distribution. High-relevance crops that contain the key entities suggest that image-based reasoning can recover missing evidence, whereas low-relevance crops suggest that invoking the image modality is unlikely to help.

Additional modality gain cues (21 features) estimate how much useful new evidence each candidate modality would contribute beyond the baseline. For the image-based modality, these include the estimated marginal gain in target-coverage, the overlap of text-image evidence, and the risk scores for irrelevant entities and distracting context. For the graph-based modality, the features capture the coverage of the target-node, the confidence of the subgraph, the quality of the relationship match and the risk scores for spurious relations, clutter and attribute loss. Seven cosine similarity features measure the semantic overlap between the question and various text artifacts (initial answer, self-critique, evidence statement, crop captions, and all artifacts combined) as well as between the full image and the crop embeddings. Low overlap between the question and existing artifacts suggests that adding a modality may introduce new information, while high overlap suggests less benefit from adding more modalities.

Within the 109 features, 71 are numeric values fed directly to the router models. The remaining 38 are free-text fields, captions, reasoning chains, and nested structured records that are PCA-embedded into fixed-dimensional vectors before routing. The numeric features are amenable to per-feature SHAP analysis; the PCA-embedded features contribute collectively through their principal components but cannot be attributed individually.

B.1.2 Audio-QA Features (98 Features)

Table 7 lists all 98 Audio-QA features. The following describes the differences between Audio-QA and VQA features.

Evidence requirement cues (20 features) retain the same target-noun and demand-structure features as VQA but replace the six visual demand flags with audio demand flags. For example, whether the question requires understanding speech content, identifying specific sounds or instruments, temporal reasoning about the order of auditory events, recognizing emotion or tone, identifying speakers, or counting auditory entities.

Uncertainty cues (16 features) are identical to VQA. Phase ① always begins with a text-only response regardless of the task family, so the uncertainty signals are the same across all three settings.

Audio evidence availability cues (43 features) estimate whether the missing evidence is likely present in the audio signal. Instead of image crops with spatial bounding boxes, the structured prompt proposes up to three audio clips, each defined by a start and end timestamp. For each clip, the features include the temporal boundaries, a confidence score, a short description of the clip content, description relevance to the question, and whether the description mentions the target nouns. Aggregated statistics (mean, standard deviation, minimum, and maximum description relevance; fraction of low-relevance clips; variance of clip confidence; relevance distribution) summarize the clip-level signals. The number of features is less than VQA since audio segments only need two-coordinate temporal boundaries instead of four-coordinate bounding boxes across three proposed regions.

Additional modality gain cues (19 features) follow the same structure as VQA but replace image-specific gain and risk features with audio-specific ones: estimated marginal coverage gain from the audio modality, text-audio evidence overlap, and risk scores for irrelevant and distracting audio content. The graph features are identical to VQA. Five cosine similarity features measure semantic overlap between the question and the text artifacts (initial answer, self-critique, evidence statement, clip descriptions, and all artifacts combined).

B.1.3 Video-QA Features (100 Features)

Table 8 lists all 100 Video-QA features. The following describes the differences between Video-QA and VQA features.

Evidence requirement cues (20 features) use the same target-noun and demand-structure features as VQA but with six video-specific demand flags, which indicate whether the question requires action recognition, temporal reasoning about the sequence of events, spatial reasoning about entity relationships within frames, appearance recognition, OCR, or counting entities across frames.

Uncertainty cues (16 features) are identical to VQA.

Video evidence availability cues (43 features) estimate whether the missing evidence is likely present in the video signal. The structured prompt proposes up to three video region clips, each defined by a spatial bounding box and start and end timestamps. For each clip, the features include the temporal boundaries, a confidence score, a short description, description relevance, target-noun containment, and matched-noun lists.

Additional modality gain cues (21 features) replace image-specific features used in VQA with video-specific ones: estimated marginal coverage gain from the video modality, text-video evidence overlap, and risk scores for irrelevant entities and distracting context in the proposed clips. The graph features are still identical across all three task families. Five cosine similarity features measure semantic overlap between the question and text artifacts. Video PCA embeddings are enabled in the Video-QA router using X3D-M [58] with 32 principal components, so two additional similarity features are available: video vs clip-mean embedding and video vs clip-max embedding.

B.2 SHAP-based Feature Importance Analysis

To evaluate whether the routing features contribute meaningfully to the router's decisions across all three task families, we conduct a SHAP analysis on both the correctness classifier and the token-cost regressor for VQA, Audio-QA, and Video-QA. We compute SHAP values on the held-out test split using TreeExplainer [62], which provides exact Shapley values for tree-based models, and report the mean absolute SHAP value for each numeric feature as the measure of global feature importance. The results are presented in Fig. 3 and 4 for VQA (71 numeric features), Fig. 5 and 6 for Audio-QA (77 numeric features), and Fig. 7 and 8 for Video-QA (82 numeric features). Across all three task families, every numeric feature achieves a non-negligible mean absolute SHAP value in at least one of the two models, showing that each feature carries useful routing signal.

Correctness Classifier. The most important correctness signals differ across the three task families.

For VQA, uncertainty cues are the most important group, contributing 38.1% of the total SHAP importance. Text confidence (0.124) and hedge term indicator (0.100) showing that lexical markers of uncertainty (e.g., *maybe*, *possibly*) provide complementary evidence beyond the numeric confidence score. They help the router identify cases where the initial text answer may be unreliable. Visual evidence availability also contributes 25.3%, showing that crop relevance and caption relevance help estimate whether image evidence can improve the answer. Additional modality gain cues contribute 21.7%, while evidence requirement cues contribute 14.8%, mainly through entity count features.

In Audio-QA, additional modality gain cues take over at 48.3%, driven by cosine similarity features that measure the semantic gap between the question and the baseline's reasoning. The question vs. self-critique similarity (0.175) is the single most important feature, with three more similarity features in the top seven. This shift aligns with the under-fusion tendency observed in Appendix E.1, since similarity features are the primary cues that distinguish audio instances requiring additional evidence from those already served well by text alone. Audio evidence availability cues contribute 31.8% through segment temporal boundaries (0.058) and target noun matches (0.047), showing that the proposed audio segments carry discriminative routing signal. Uncertainty cues are less important than in VQA, contributing 9.8%, while evidence requirement cues contribute 10.1%.

In Video-QA, additional modality gain cues and video evidence availability contribute most to the correctness classifier, contributing 48.9% and 39.8%, respectively. The router relies heavily on whether the proposed clips are temporally relevant (0.179) and semantically aligned with the question (0.176), indicating that clip temporal position and semantic coverage are both critical for predicting correctness. Multi-event grounding (0.053) is also important, reflecting the need to reason over event

sequences in video tasks. In contrast, uncertainty cues contribute only 1.3%, showing that baseline text confidence is much less informative for Video-QA than for VQA.

Across the three settings, uncertainty cues decline sharply from 38.1% in VQA to 9.8% in Audio-QA and 1.3% in Video-QA. As X moves from static images to temporal audio and video signals, the router shifts from relying on baseline confidence to relying on whether the proposed X regions contain decisive evidence.

Token Cost Regressor. For token cost prediction, modality-specific evidence availability is the strongest signal across all three task families. It contributes 39.6% in VQA, 39.3% in Audio-QA, and 52.0% in Video-QA. This pattern shows that the cost model mainly depends on how much visual, audio, or video evidence needs to be processed. In VQA, visual crop features contribute most to token cost prediction, with the first crop's bounding box coordinate (34.6), crop confidence (29.7), and caption relevance (25.5) ranking among the top features. Evidence requirement cues are also important, with entity count (122.7) serving as the strongest individual cost predictor. Questions involving more target entities tend to require more crop descriptions and more relational reasoning. In Audio-QA, the cost pattern is similar, but the SHAP magnitudes are lower. Entity count (68.4) remains the strongest cost predictor, while audio segment boundaries and relevance scores provide additional cost signals. The lower SHAP scale matches the lower token cost range of Audio-QA. The group distribution is very similar to VQA, with uncertainty cues at 22.3%, evidence requirement cues at 19.6%, and gain cues at 18.8%. In Video-QA, video evidence availability has the largest impact on token cost. Features related to the second proposed clip are especially important, suggesting that deciding whether to use an additional clip is a major source of cost variation. Modality gain cues and evidence requirement cues also contribute strongly. The cost SHAP magnitudes are much larger than those in VQA and Audio-QA, consistent with the substantially higher token costs of Video-QA.

Cross-Task Comparison. The SHAP results show that different task families require different routing signals. For correctness prediction, the main signal shifts from uncertainty in VQA to modality gain and task-dependent evidence availability in Audio-QA and Video-QA. This indicates that routing becomes more dependent on localized task-dependent evidence as the modality changes from images to audio and video. For token cost prediction, the three task families share a common pattern. Evidence availability and evidence requirement cues are consistently strong predictors. Entity count is especially useful across tasks, while modality-specific evidence features capture the cost of processing crops, segments, and clips. These results validate the four feature groups in AIM: evidence requirements, baseline uncertainty, modality-specific evidence availability, and marginal modality gain. Each group provides complementary signals for accurate and token-efficient modality selection across tasks.

Table 6: List of 109 router features organized by routing-risk group. The first three groups primarily detect *under-fusion* (missing decisive evidence); the fourth group primarily reduces *over-fusion* (unnecessary token cost).

Group	Feature Name	Description
Evidence requirement cues	`target_nouns`	Key nouns/entities from question and options
	`target_nouns_len`	# of extracted target nouns
	`target_nouns_{0,1,2,3}`	Target noun at index 0-3 **(4 features)**
	`demand`	Structured question demand summary
	`demand_question_category`	Assigned high-level question categories
	`demand_question_category_len`	# of assigned categories
	`demand_question_category_{0,1}`	Category at index 0, 1 **(2 features)**
	`demand_target_noun_count`	# of target nouns referenced in question
	`demand_requires_localization`	Requires locating entities in image
	`demand_requires_interaction`	Requires understanding an interaction
	`demand_requires_spatial_reasoning`	Requires spatial reasoning
	`demand_requires_color_or_texture`	Requires color or texture cues
	`demand_requires_ocr`	Requires reading text (OCR)
	`demand_requires_counting`	Requires counting entities
	`demand_relation_cue_count`	# of relation/spatial cue terms in question
	`text_evidence_to_confirm`	Visual evidence needed to confirm hypothesis
Uncertainty cues	`text`	Structured text-only output
	`text_initial_answer`	short initial answer justification from text output
	`text_confidence`	Text output confidence score [0-1]
	`text_reasoning`	Short reasoning supporting the initial answer
	`text_signals`	Diagnostic signals from text artifact
	`text_signals_has_hedges`	Whether hedge terms (e.g., *maybe*) appear
	`text_signals_hedge_terms_found`	List of detected hedge terms
	`text_signals_cites_visual_evidence`	Whether visual evidence is cited in reasoning
	`text_signals_generic_priors`	Whether generic priors (e.g., *usually*) are used
	`text_signals_prior_terms_found`	List of detected generic-prior terms
	`text_signals_prior_terms_len`	# of generic-prior terms found
	`text_signals_prior_terms_0`	First generic-prior term found
	`text_self_critique`	Self-critique of the hypothesis answer
	`text_self_critique_neg_reasoning`	Counter-argument against the hypothesis
	`router_text_confidence`	Router's text-artifact confidence score
	`router_confidence_calibration_flag`	Calibration: overconfident / underconfident / ok
Visual evidence availability cues	`image`	Structured image-grounded output
	`image_crops`	List of proposed image crops
	`image_crop_`k	Record for crop k, $k \in \{0, 1, 2\}$ **(3 features)**
	`image_crop_`k`_box`	Normalized bounding box for crop k **(3 features)**
	`image_crop_`k`_box_{0..3}`	Box coords x1,y1,x2,y2 for crop k **(12 features)**
	`image_crop_`k`_confidence`	Confidence crop k has decisive evidence **(3 features)**
	`image_crop_`k`_caption`	Short caption of crop k **(3 features)**
	`image_crop_`k`_caption_relevance`	Caption relevance to question for crop k **(3 features)**
	`image_crop_`k`_contains_targets`	Whether crop k caption mentions targets **(3 features)**
	`image_crop_`k`_matched_nouns`	Target nouns matched in crop k **(3 features)**
	`image_crop_{0,1}_matched_nouns_len`	# of matched target nouns in crop 0, 1 **(2 features)**
	`image_crop_{0,1}_matched_nouns_0`	First matched noun for crop 0, 1 **(2 features)**
	`image_crop_0_matched_nouns_1`	Second matched noun for crop 0
	`router_best_caption_relevance`	Max caption relevance across all crops
	`router_crop_relevance_mean`	Mean caption relevance across crops
	`router_crop_relevance_std`	Std. of caption relevance across crops
	`router_crop_relevance_min`	Min caption relevance across crops
	`router_crop_relevance_max`	Max caption relevance across crops
	`router_low_relevance_crop_ratio`	Fraction of crops with low relevance
	`router_crop_confidence_variance`	Variance of crop confidence scores
	`router_crop_relevance_distribution`	Bucketed crop relevance distribution
	`router_crop_rel_dist_very_low`	Share of crops in very-low relevance bucket
	`router_crop_rel_dist_low`	Share of crops in low relevance bucket
	`router_crop_rel_dist_medium`	Share of crops in medium relevance bucket
	`router_crop_rel_dist_high`	Share of crops in high relevance bucket
	`router`	Structured router feature bundle
	`router_image_marginal_coverage_gain`	Estimated target-coverage gain from image

Continued on next page

Group	Feature Name	Description
Additional modality gain cues	router_text_image_overlap	Text-image evidence overlap
	router_image_irrelevant_entity_risk	Risk that crops emphasize irrelevant entities
	router_image_distracting_context_risk	Risk that crops include distracting context
	router_graph_target_node_coverage	Target-entity coverage among graph nodes
	router_graph_subgraph_confidence	Confidence of target-focused subgraph
	router_graph_relation_match	Graph-question relation match
	router_graph_irrelevant_node_ratio	Fraction of irrelevant graph nodes
	router_graph_spurious_relation_risk	Risk of spurious graph relations
	router_graph_clutter_risk	Risk of graph clutter or noise
	router_graph_attribute_loss_risk	Risk of attribute loss in graph abstraction
Additional modality gain cues	router_graph_marginal_relation_gain	Relation-coverage gain from graph
	router_image_graph_overlap	Image-graph evidence overlap
	similarity_q_initial	Cosine similarity: question vs. initial answer
	similarity_q_critique	Cosine similarity: question vs. self-critique
	similarity_q_evidence	Cosine similarity: question vs. evidence
	similarity_q_caption	Cosine similarity: question vs. crop captions
	similarity_q_all	Cosine similarity: question vs. all text artifacts
	similarity_image_cropmean	Cosine similarity: image vs. mean crop
	similarity_image_cropmax	Cosine similarity: image vs. max crop

Table 7: List of 98 Audio-QA router features organized by routing-risk group. The first three groups primarily detect *under-fusion* (missing decisive evidence); the fourth group primarily reduces *over-fusion* (unnecessary token cost).

Group	Feature Name	Description
Evidence requirement cues	`target_nouns`	Key nouns/entities from question and options
	`target_nouns_len`	# of extracted target nouns
	`target_nouns_{0,1,2,3}`	Target noun at index 0–3 **(4 features)**
	`demand`	Structured question demand summary
	`demand_question_category`	Assigned high-level question categories
	`demand_question_category_len`	# of assigned categories
	`demand_question_category_{0,1}`	Category at index 0, 1 **(2 features)**
	`demand_target_noun_count`	# of target nouns referenced in question
	`demand_requires_speech_content`	Requires understanding speech or dialogue
	`demand_requires_sound_identification`	Requires identifying specific sounds or instruments
	`demand_requires_temporal_reasoning`	Requires reasoning about temporal order of sounds
	`demand_requires_emotion_or_tone`	Requires recognizing emotion, mood, or tone
	`demand_requires_speaker_identification`	Requires identifying who is speaking
	`demand_requires_counting`	Requires counting auditory entities or events
	`demand_relation_cue_count`	# of relation/temporal cue terms in question
	`text_evidence_to_confirm`	Audio evidence needed to confirm hypothesis
Uncertainty cues	`text`	Structured text-only output
	`text_initial_answer`	Short initial answer justification from text output
	`text_confidence`	Text output confidence score [0–1]
	`text_reasoning`	Short reasoning supporting the initial answer
	`text_signals`	Diagnostic signals from text artifact
	`text_signals_has_hedges`	Whether hedge terms (e.g., *maybe*) appear
	`text_signals_hedge_terms_found`	List of detected hedge terms
	`text_signals_cites_visual_evidence`	Whether modality-specific evidence is cited
	`text_signals_generic_priors`	Whether generic priors (e.g., *usually*) are used
	`text_signals_prior_terms_found`	List of detected generic-prior terms
	`text_signals_prior_terms_len`	# of generic-prior terms found
	`text_signals_prior_terms_0`	First generic-prior term found
	`text_self_critique`	Self-critique of the hypothesis answer
	`text_self_critique_neg_reasoning`	Counter-argument against the hypothesis
	`router_text_confidence`	Router's text-artifact confidence score
	`router_confidence_calibration_flag`	Calibration: overconfident / underconfident / ok
Audio evidence availability cues	`audio`	Structured audio-grounded output
	`audio_clips`	List of proposed audio clips
	`audio_clip_`k	Record for clip k, $k \in \{0, 1, 2\}$ **(3 features)**
	`audio_clip_`k`_start`	Start timestamp (seconds) for clip k **(3 features)**
	`audio_clip_`k`_end`	End timestamp (seconds) for clip k **(3 features)**
	`audio_clip_`k`_confidence`	Confidence clip k has decisive evidence **(3 features)**
	`audio_clip_`k`_description`	Short description of clip k content **(3 features)**
	`audio_clip_`k`_description_relevance`	Description relevance to question for clip k **(3 features)**
	`audio_clip_`k`_contains_targets`	Whether clip k description mentions targets **(3 features)**
	`audio_clip_`k`_matched_nouns`	Target nouns matched in clip k **(3 features)**
	`audio_clip_{0,1}_matched_nouns_len`	# of matched target nouns in clip 0, 1 **(2 features)**
	`audio_clip_{0,1}_matched_nouns_0`	First matched noun for clip 0, 1 **(2 features)**
	`audio_clip_0_matched_nouns_1`	Second matched noun for clip 0
	`router_best_description_relevance`	Max description relevance across all clips
	`router_clip_relevance_mean`	Mean description relevance across clips
	`router_clip_relevance_std`	Std. of description relevance across clips
	`router_clip_relevance_min`	Min description relevance across clips
	`router_clip_relevance_max`	Max description relevance across clips
	`router_low_relevance_clip_ratio`	Fraction of clips with low relevance
	`router_clip_confidence_variance`	Variance of clip confidence scores
	`router_clip_relevance_distribution`	Bucketed clip relevance distribution
	`router_clip_rel_dist_very_low`	Share of clips in very-low relevance bucket
	`router_clip_rel_dist_low`	Share of clips in low relevance bucket
	`router_clip_rel_dist_medium`	Share of clips in medium relevance bucket
	`router_clip_rel_dist_high`	Share of clips in high relevance bucket
	`router`	Structured router feature bundle
	`router_audio_marginal_coverage_gain`	Estimated target-coverage gain from audio

Continued on next page

Group	Feature Name	Description
Additional modality gain cues	router_text_audio_overlap	Text–audio evidence overlap
	router_audio_irrelevant_content_risk	Risk that clips emphasize irrelevant content
	router_audio_distracting_content_risk	Risk that clips include distracting content
	router_graph_target_node_coverage	Target-entity coverage among graph nodes
	router_graph_subgraph_confidence	Confidence of target-focused subgraph
	router_graph_relation_match	Graph–question relation match
	router_graph_irrelevant_node_ratio	Fraction of irrelevant graph nodes
	router_graph_spurious_relation_risk	Risk of spurious graph relations
	router_graph_clutter_risk	Risk of graph clutter or noise
	router_graph_attribute_loss_risk	Risk of attribute loss in graph abstraction
	router_graph_marginal_relation_gain	Relation-coverage gain from graph
	router_audio_graph_overlap	Audio–graph evidence overlap
	similarity_q_initial	Cosine similarity: question vs. initial answer
	similarity_q_critique	Cosine similarity: question vs. self-critique
	similarity_q_evidence	Cosine similarity: question vs. evidence
	similarity_q_description	Cosine similarity: question vs. clip descriptions
	similarity_q_all	Cosine similarity: question vs. all text artifacts

Table 8: List of 100 Video-QA router features organized by routing-risk group. The first three groups primarily detect *under-fusion* (missing decisive evidence); the fourth group primarily reduces *over-fusion* (unnecessary token cost).

Group	Feature Name	Description
Evidence requirement cues	`target_nouns`	Key nouns/entities from question and options
	`target_nouns_len`	# of extracted target nouns
	`target_nouns_{0,1,2,3}`	Target noun at index 0–3 **(4 features)**
	`demand`	Structured question demand summary
	`demand_question_category`	Assigned high-level question categories
	`demand_question_category_len`	# of assigned categories
	`demand_question_category_{0,1}`	Category at index 0, 1 **(2 features)**
	`demand_target_noun_count`	# of target nouns referenced in question
	`demand_requires_action_recognition`	Requires understanding actions or activities
	`demand_requires_temporal_reasoning`	Requires reasoning about event sequence
	`demand_requires_spatial_reasoning`	Requires spatial reasoning within frames
	`demand_requires_appearance_recognition`	Requires recognizing visual appearance details
	`demand_requires_ocr`	Requires reading in-frame text (OCR)
	`demand_requires_counting`	Requires counting entities across frames
	`demand_relation_cue_count`	# of relation/temporal cue terms in question
	`text_evidence_to_confirm`	Video evidence needed to confirm hypothesis
Uncertainty cues	`text`	Structured text-only output
	`text_initial_answer`	Short initial answer justification from text output
	`text_confidence`	Text output confidence score [0–1]
	`text_reasoning`	Short reasoning supporting the initial answer
	`text_signals`	Diagnostic signals from text artifact
	`text_signals_has_hedges`	Whether hedge terms (e.g., *maybe*) appear
	`text_signals_hedge_terms_found`	List of detected hedge terms
	`text_signals_cites_visual_evidence`	Whether modality-specific evidence is cited
	`text_signals_generic_priors`	Whether generic priors (e.g., *usually*) are used
	`text_signals_prior_terms_found`	List of detected generic-prior terms
	`text_signals_prior_terms_len`	# of generic-prior terms found
	`text_signals_prior_terms_0`	First generic-prior term found
	`text_self_critique`	Self-critique of the hypothesis answer
	`text_self_critique_neg_reasoning`	Counter-argument against the hypothesis
	`router_text_confidence`	Router's text-artifact confidence score
	`router_confidence_calibration_flag`	Calibration: overconfident / underconfident / ok
Video evidence availability cues	`video`	Structured video-grounded output
	`video_clips`	List of proposed video region clips
	`video_clip_`k	Record for clip k, $k \in \{0, 1, 2\}$ **(3 features)**
	`video_clip_`k`_start`	Start timestamp (seconds) for clip k **(3 features)**
	`video_clip_`k`_end`	End timestamp (seconds) for clip k **(3 features)**
	`video_clip_`k`_confidence`	Confidence clip k has decisive evidence **(3 features)**
	`video_clip_`k`_description`	Short description of clip k content **(3 features)**
	`video_clip_`k`_description_relevance`	Description relevance to question for clip k **(3 features)**
	`video_clip_`k`_contains_targets`	Whether clip k description mentions targets **(3 features)**
	`video_clip_`k`_matched_nouns`	Target nouns matched in clip k **(3 features)**
	`video_clip_{0,1}_matched_nouns_len`	# of matched target nouns in clip 0, 1 **(2 features)**
	`video_clip_{0,1}_matched_nouns_0`	First matched noun for clip 0, 1 **(2 features)**
	`video_clip_0_matched_nouns_1`	Second matched noun for clip 0
	`router_best_description_relevance`	Max description relevance across all clips
	`router_clip_relevance_mean`	Mean description relevance across clips
	`router_clip_relevance_std`	Std. of description relevance across clips
	`router_clip_relevance_min`	Min description relevance across clips
	`router_clip_relevance_max`	Max description relevance across clips
	`router_low_relevance_clip_ratio`	Fraction of clips with low relevance
	`router_clip_confidence_variance`	Variance of clip confidence scores
	`router_clip_relevance_distribution`	Bucketed clip relevance distribution
	`router_clip_rel_dist_very_low`	Share of clips in very-low relevance bucket
	`router_clip_rel_dist_low`	Share of clips in low relevance bucket
	`router_clip_rel_dist_medium`	Share of clips in medium relevance bucket
	`router_clip_rel_dist_high`	Share of clips in high relevance bucket
	`router`	Structured router feature bundle
	`router_video_marginal_coverage_gain`	Estimated target-coverage gain from video

Continued on next page

Group	Feature Name	Description
Additional modality gain cues	router_text_video_overlap	Text–video evidence overlap
	router_video_irrelevant_entity_risk	Risk that clips emphasize irrelevant entities
	router_video_distracting_context_risk	Risk that clips include distracting context
	router_graph_target_node_coverage	Target-entity coverage among graph nodes
	router_graph_subgraph_confidence	Confidence of target-focused subgraph
	router_graph_relation_match	Graph–question relation match
	router_graph_irrelevant_node_ratio	Fraction of irrelevant graph nodes
	router_graph_spurious_relation_risk	Risk of spurious graph relations
	router_graph_clutter_risk	Risk of graph clutter or noise
	router_graph_attribute_loss_risk	Risk of attribute loss in graph abstraction
	router_graph_marginal_relation_gain	Relation-coverage gain from graph
	router_video_graph_overlap	Video–graph evidence overlap
	similarity_q_initial	Cosine similarity: question vs. initial answer
	similarity_q_critique	Cosine similarity: question vs. self-critique
	similarity_q_evidence	Cosine similarity: question vs. evidence
	similarity_q_description	Cosine similarity: question vs. clip descriptions
	similarity_q_all	Cosine similarity: question vs. all text artifacts
	similarity_video_clipmean	Cosine similarity: video vs. mean clip embedding
	similarity_video_clipmax	Cosine similarity: video vs. max clip embedding

C Best Combination Distribution for Audio-QA and Video-QA

We extend the best combination analysis from VQA (Section 3) to Audio-QA and Video-QA. For each instance, we define the best combination as the correct one with the lowest token cost, or the lowest-cost combination if none is correct. We group questions into task-specific question types.

Best Combination Analysis for Audio-QA. We group Audio-QA instances into five question types: (i) *counting*, which asks how many auditory events or entities are present; (ii) *attribute*, which asks about acoustic properties, tone, emotion, or musical characteristics; (iii) *temporal*, which asks about temporal ordering or duration of sounds; (iv) *environment*, which asks about scene or setting identification; and (v) *source*, which asks what produces a sound. As shown in Fig. 9a, the best combination distribution varies across question types. Counting is the only question type where the task-dependent modality X (32.4%) is higher than T (29.4%), indicating that the raw audio signal is essential for distinguishing auditory event counts. For attribute and environment questions, graph-based interaction G is more frequently the best combination (20.2% and 24.4%) than other non-text modalities, reflecting that structured relational cues help ground fine-grained properties and scene context. For temporal questions, X (20.6%) and T + X (11.8%) benefit more from direct access to the audio signal, since reasoning about the order of auditory events requires hearing the original sequence. Source identification is the question type that prefers T most (59.1%), yet over 40% of instances still benefit from non-text modalities.

Best Combination Analysis for Video-QA. We group Video-QA instances into five question types: (i) *action*, which asks about behaviors or activities; (ii) *scene*, which asks about visual setting, appearance, or emotional state; (iii) *temporal*, which asks about temporal ordering of events; (iv) *spatial*, which asks about spatial relationships or counting entities across frames; and (v) *causal*, which asks about motivations or causes. As shown in Fig. 9b, the best combination distribution again varies by question type but differs from VQA and Audio-QA. Action questions benefit from a wider range of combinations, with T at 45.7%, G at 13.9%, and T + G + X at 13.4%, showing that recognizing behaviors and activities can require both structured relations and multimodal evidence. T + G is consistently strong across temporal (16.2%) and causal (15.7%) questions, suggesting that graph-structured relations help ground event sequences and causal chains. For spatial questions, X benefits more (14.3%) than in other question types, since raw video frames provide direct spatial layout information. Scene questions show the strongest G contribution (23.3%), reflecting that structured scene graphs effectively capture visual context.

Cross-Task Comparison. Across all three task families, no single modality combination is best for every question type, consistent with the findings for VQA in Section 3. The best combination varies by task. In VQA, T + X is most prominent (Fig. 1). In Audio-QA and Video-QA, T is



Figure 3: Feature importance (mean absolute SHAP value) of 71 numeric VQA router features for the **correctness classifier** (CatBoost), which predicts whether the selected modality subset will yield a correct answer. The remaining 38 features (free-text fields, captions, reasoning chains, and nested structured records) are PCA-embedded before routing and excluded from per-feature SHAP analysis.

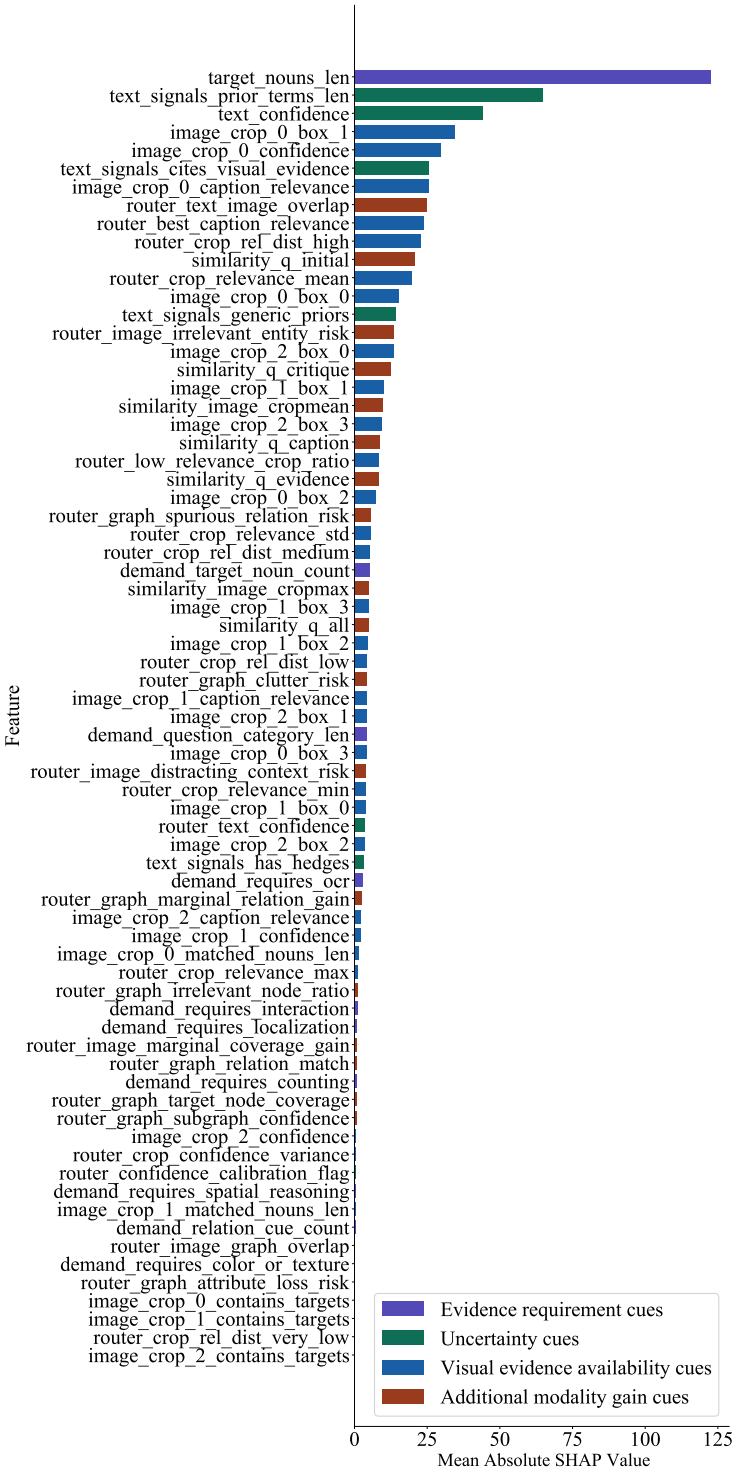


Figure 4: Feature importance (mean absolute SHAP value) of 71 numeric VQA router features for the **token-cost regressor** (LightGBM).



Figure 5: Feature importance (mean absolute SHAP value) of 77 non-PCA Audio-QA router features for the **correctness classifier** (XGBoost).

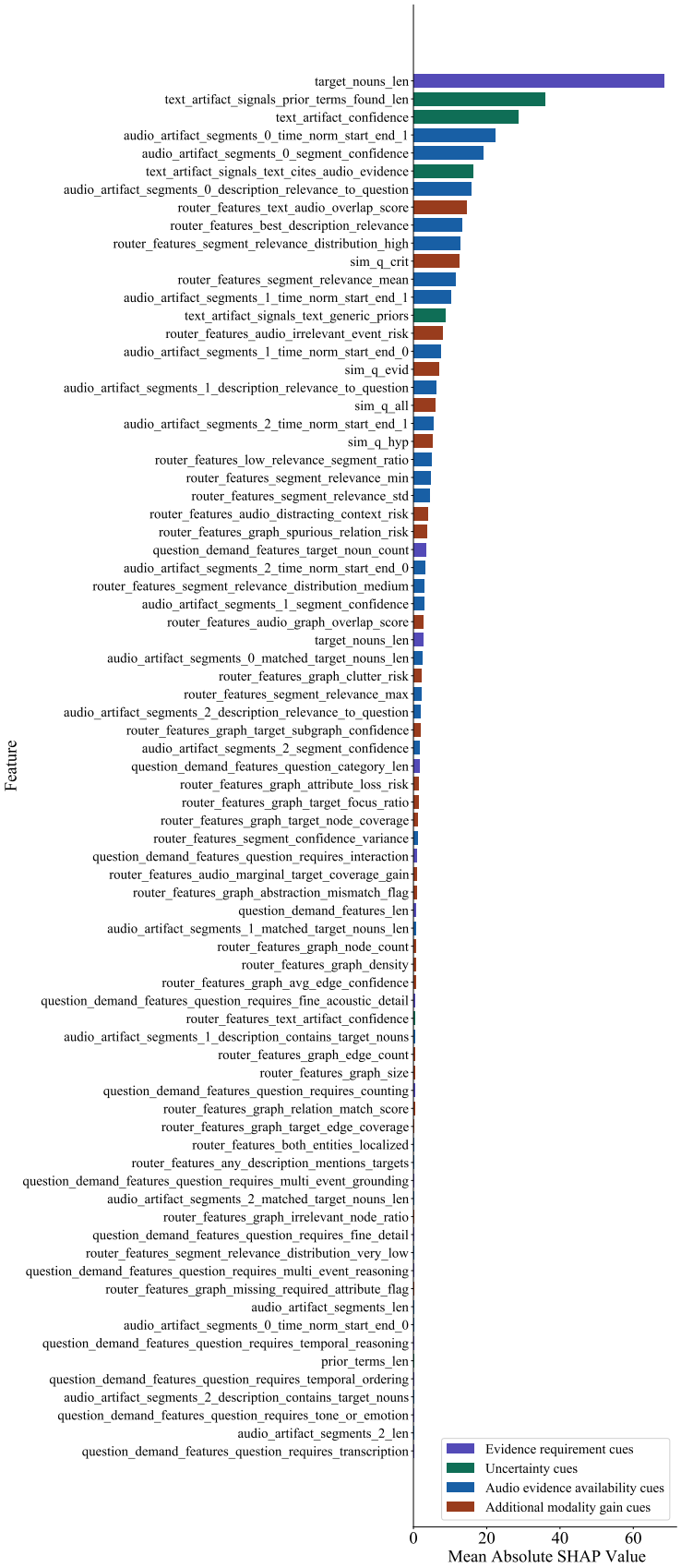


Figure 6: Feature importance (mean absolute SHAP value) of 77 non-PCA Audio-QA router features for the **token-cost regressor** (XGBoost).

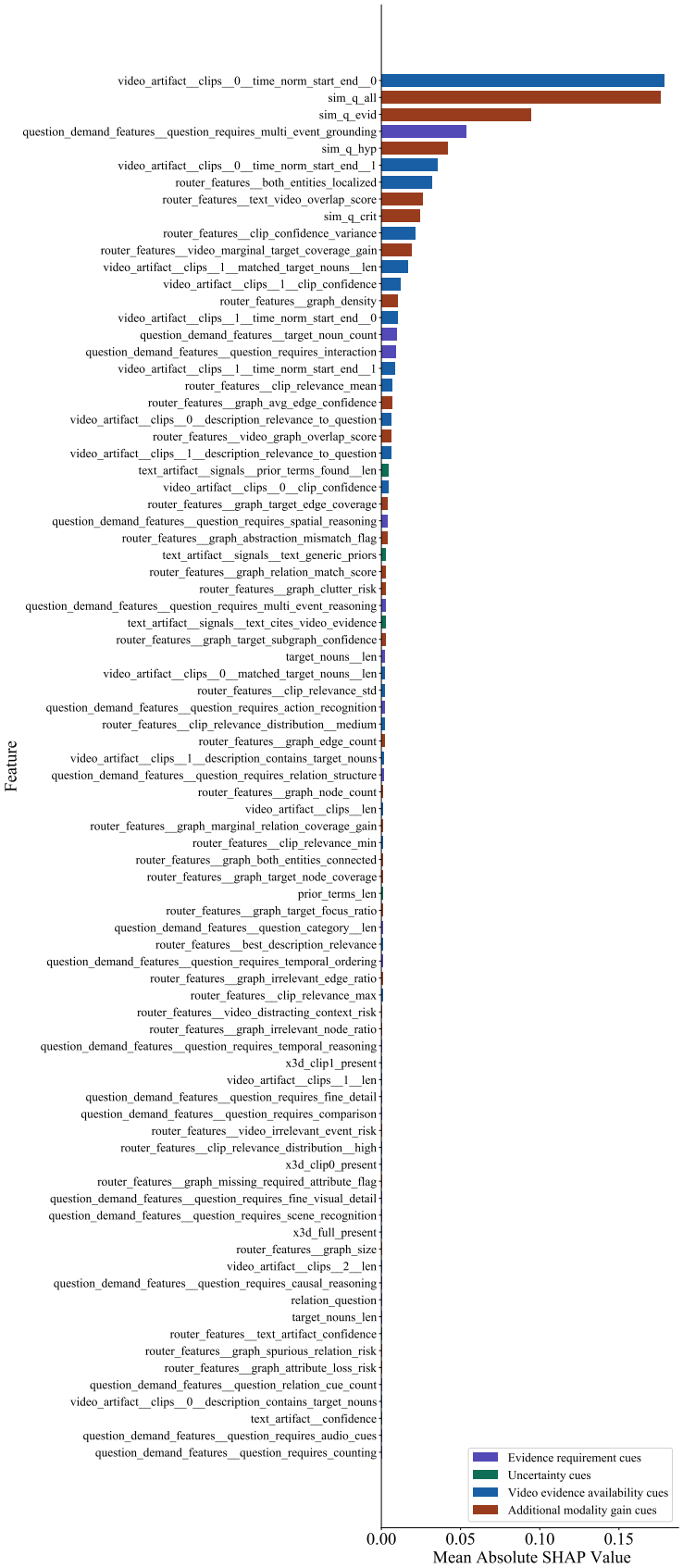


Figure 7: Feature importance (mean absolute SHAP value) of 82 non-PCA Video-QA router features for the **correctness classifier** (XGBoost).

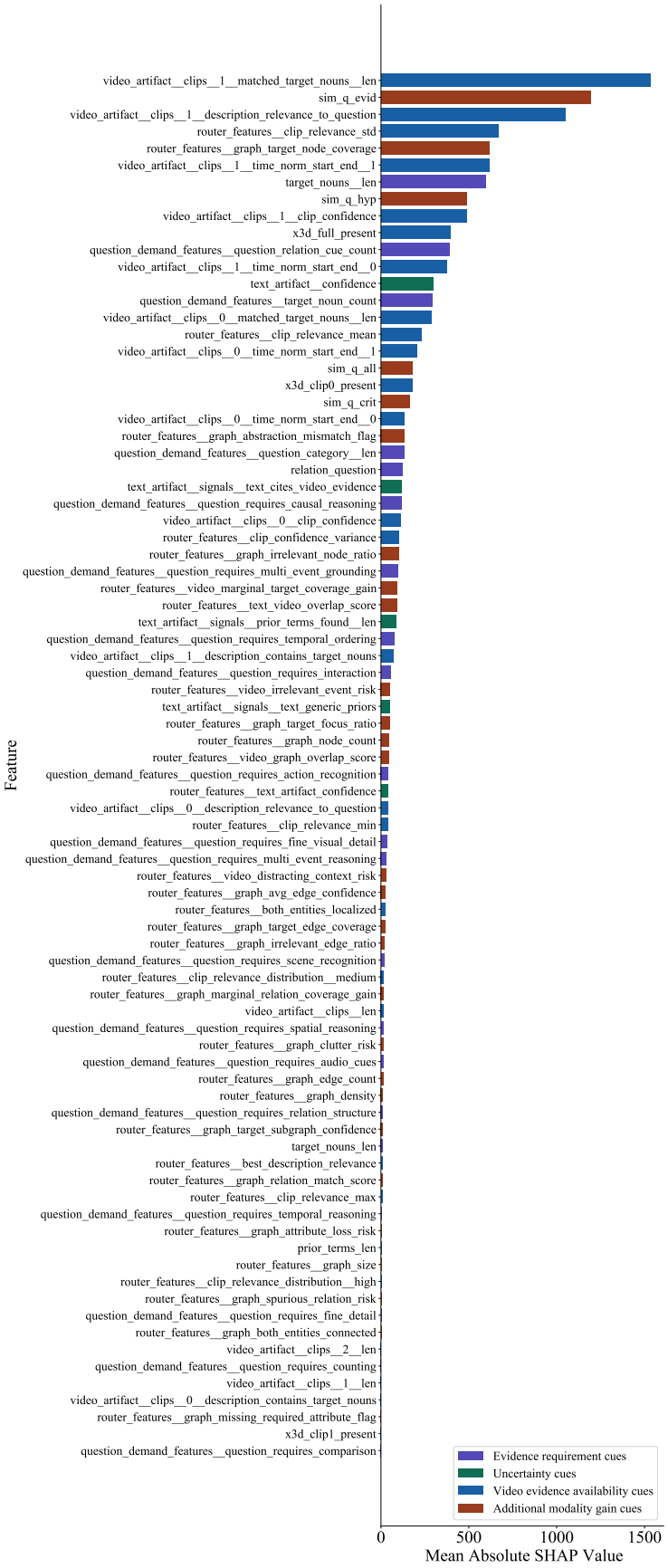
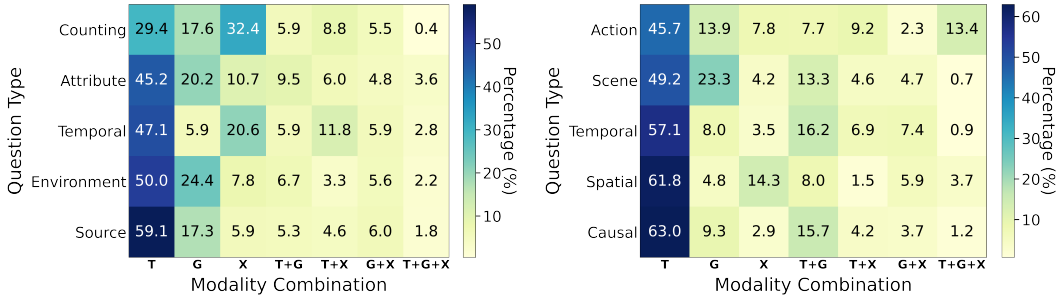


Figure 8: Feature importance (mean absolute SHAP value) of 82 non-PCA Video-QA router features for the **token-cost regressor** (XGBoost).



(a) Audio-QA (b) Video-QA

Figure 9: Best combination distribution across five question types. Each row sums to 100%.

more likely to be selected, but X and G benefit more in Audio-QA while T + G benefits more in Video-QA. These differences reflect the token cost of each task-dependent modality. In VQA and Audio-QA, the task-dependent modality uses fewer tokens, so combinations that include it can achieve higher accuracy while keeping token cost low. In Video-QA, the task-dependent modality uses substantially more tokens, so the best combination more often favors T + G, which provides structured evidence beyond text alone. These results reinforce the motivation for adaptive modality selection, since the best combination depends not only on the question type but also on the token cost of the task-dependent modality in each task family.

D Latency and Cost Analyses

In this section, we analyze the latency and cost of AIM. We evaluate the latency overhead of the routing pipeline in Appendix D.1, report the token cost of AIM by phase in Appendix D.2, and quantify the one-time offline cost of training the routers in Appendix D.3.

D.1 Latency Overhead Analysis

To evaluate the latency overhead of AIM, we measure the end-to-end latency per instance of AIM and of the seven fixed combinations on all eight benchmarks. The latency overhead of AIM covers the structured baseline response, feature extraction, and adaptive modality routing, relative to directly executing MAD with all seven fixed-modality baselines. We report the average latency per instance in seconds in Table 9. AIM's feature extraction and routing add negligible overhead to the total latency (less than 0.3s per instance in our measurements). The remaining latency stems from the structured baseline LLM response and MAD execution, both required by any MAD method. On average, the end-to-end latency of AIM is 16.5s for VQA, 11.7s for Audio-QA, and 15.4s for Video-QA. By adaptively selecting the best modality combination per instance, AIM balances accuracy and latency. The total latency depends on multiple factors, including the number of modalities processed, the input size of each modality (e.g., uploading and encoding video is much slower than text), the prompt length during debate, and the LLM generation time. Since AIM selects the modality combination that can answer correctly at low token cost, it avoids expensive modalities when they are unnecessary, which often reduces latency as well. Across the eight benchmarks, AIM is faster than the highest-accuracy fixed-modality combination on four datasets while achieving higher accuracy on all eight. On OKVQA, AIM is faster than T + G + X (14.0s vs. 21.3s) while improving accuracy by 0.3%. On NextQA, AIM is faster than T + G + X (14.6s vs. 28.2s) while improving accuracy by 1.6%. On IntentQA, AIM is faster than T + X (16.1s vs. 19.4s) while improving accuracy by 4.2%. On VCR, AIM is comparable to G + X (22.0s vs. 22.5s) while improving accuracy by 3.1%. On the remaining four datasets (GQA, VQA-v2, Clotho, and DCASE), AIM is 1.5-6.8s slower than the highest-accuracy fixed combination, but these latency increases are modest relative to the accuracy gains of 0.9%-3.7%.

Table 9: Latency comparison across eight multimodal QA benchmarks (in seconds). Bold indicates the lowest latency for each dataset.

Dataset	One Modality			Two Modalities			All	Adaptive
	T	G	X	T + G	T + X	G + X	T + G + X	AIM
GQA	8.4	15.3	**6.8**	8.2	10.0	16.7	18.1	16.4
OKVQA	8.7	12.1	**6.4**	8.3	6.7	19.0	21.3	14.0
VCR	12.5	14.7	**8.8**	14.2	15.7	22.5	25.4	22.0
VQA-v2	8.0	10.7	**6.1**	8.2	6.8	18.2	21.4	13.6
Clotho	6.0	**5.8**	9.8	6.2	9.7	10.0	10.2	11.7
DCASE	5.3	9.1	**5.2**	6.9	9.1	10.7	17.3	11.6
IntentQA	**9.8**	15.6	28.7	10.2	19.4	22.3	20.2	16.1
NextQA	**10.1**	18.2	29.1	13.1	25.3	22.9	28.2	14.6

D.2 Token Cost Analysis

The token cost of AIM in Table 2 covers both the Phase ① structured baseline response and the final MAD execution with the selected modality combination, whereas feature extraction in Phase ② and the routing decision in Phase ③ are local computations that do not consume LLM tokens and add less than 0.3s latency per instance (Appendix D.1). We report the two parts of the token cost of AIM in Table 10, averaged over the benchmarks of each task family, and make two observations. First, the MAD execution accounts for most of the token cost of AIM, with 68.4%, 66.2%, and 83.7% on VQA, Audio-QA, and Video-QA, respectively, and the share is largest on Video-QA since the video clips exchanged in the debate are expensive. Second, the structured baseline response is cheap relative to the debate it serves. For example, on Video-QA, the structured baseline response uses 7,683 tokens, whereas the MAD execution uses 39,511 tokens. Therefore, the token cost of AIM is determined by the modality combination selected for the debate rather than by the structured baseline response.

Table 10: Average token cost per instance of AIM, split into the Phase ① structured baseline agent and the final MAD execution with the selected modality combination.

Task Family	Structured Baseline Response	MAD Execution	Total
VQA	2,203	4,773	6,976
Audio-QA	1,298	2,545	3,843
Video-QA	7,683	39,511	47,194

D.3 Offline Training Cost and Amortization

To quantify the one-time offline cost of AIM, we report the training cost of the adaptive modality router for each task family, amortize the training cost over inference queries.

Training Cost. Training the adaptive modality router requires a one-time data collection. In the data collection, we run MAD under each of the seven modality combinations on every auxiliary instance and record the correctness and token cost of each modality combination. Each auxiliary instance therefore requires one debate per candidate modality combination, and the number of candidate modality combinations is $2^M - 1$ for M interaction modalities. AIM uses $M = 3$, and a new task family changes what the task-dependent modality X represents without increasing M. Moreover, the data collection is not repeated at inference, since MAD executes only the selected modality combination. We report the data collection cost and the router training time for each task family in Table 11 and make two observations. First, the data collection uses 172.64M tokens in total at an estimated API cost of 195.25 USD, and Video-QA accounts for most of the cost with 137.35M tokens, since video inputs are expensive. Second, training the three routers takes only 47.9s in total on one NVIDIA RTX 4090 GPU. Therefore, the one-time cost of AIM stems almost entirely from running MAD on the auxiliary datasets, and this cost is paid once per task family.

Table 11: Training cost of the adaptive modality router for each task family.

Task Family	Auxiliary Instances	Collection Tokens	Estimated API Cost (USD)	Router Training Time (s)
VQA	500	27.73M	31.36	19.3
Audio-QA	250	7.56M	8.55	13.4
Video-QA	250	137.35M	155.34	15.2
Total	1,000	172.64M	195.25	47.9

Amortization. To evaluate how fast the one-time cost is recovered, we amortize the one-time cost over the evaluation queries in our study. The amortized cost is 107 tokens and 0.00012 USD per VQA query, 1,443 tokens and 0.0016 USD per Audio-QA query, and 19,337 tokens and 0.022 USD per Video-QA query. The amortized cost can be further reduced as each router serves more queries, since each router is trained once on the auxiliary datasets and then applied to every benchmark of the same task family without retraining, as in our evaluation. Video-QA has the largest one-time cost, but Video-QA also has the largest saving at inference. On NextQA, AIM achieves 78.8% accuracy with 48,636 tokens per query, whereas the most accurate fixed combination $\mathrm{T} + \mathrm{G} + \mathrm{X}$ achieves 77.2% accuracy with 99,765 tokens per query. The saving is 51,129 tokens per query, and therefore the saving recovers the full Video-QA data collection cost of 137.35M tokens after about 2,686 queries.

E Ablation Studies

In this section, we present two ablation studies on the design of AIM. We compare the structured routing prompt with a standard CoT prompt in Appendix E.1, and evaluate the contribution of each feature group by removing one feature group at a time in Appendix E.2.

E.1 Structured Routing Prompt vs. CoT Prompt

A key design choice in AIM is the structured modality-routing prompt (Phase ①), which generates not only an initial answer but also explicit confidence scores, self-critique, missing-evidence statements, and task-dependent modality summaries. These structured outputs enable the extraction of a set of (109 for VQA, 98 for Audio-QA, 100 for Video-QA) interpretable routing features that drive the adaptive modality router. To evaluate the importance of this design, we replace the structured prompt with a standard CoT prompt that produces only a reasoning trace and a final answer, and retrain the router using only features extractable from CoT outputs (e.g., answer length, token count, text embeddings of the reasoning trace). All other components, including the router classifiers, token-cost regressors, and the routing decision rule, remain identical.

As shown in Table 12, replacing the structured routing prompt with a standard CoT prompt consistently degrades accuracy across all eight benchmarks. The structured prompt improves accuracy by 3.0% on GQA, 5.0% on VQA-v2, 5.5% on Clotho, and 10.2% on IntentQA, with an average gain of 4.45% across all benchmarks. These improvements are especially large on audio and video tasks, where the CoT-based router tends to favor cheaper (i.e., lower token cost) text-based modality since audio-based and video-based evidence is more token-expensive and its usefulness is not explicitly represented in the CoT output. Although the CoT variant can still extract weak routing cues from the reasoning trace, the structured prompt provides more reliable confidence, missing-evidence, and modality-usefulness signals for adaptive routing.

The token-cost results further reveal that the structured prompt improves routing quality in two complementary ways. On GQA, OKVQA, VCR, and DCASE, it achieves both higher accuracy and lower token cost. For example, compared with the CoT prompt, the structured prompt reduces the average token cost by 300 tokens on GQA, 380 tokens on OKVQA, 679 tokens on VCR, and 151 tokens on DCASE, while still improving accuracy on all four datasets. This indicates that the structured routing features help the router avoid over-fusion: when additional modalities are unlikely to help, the router can select the modality combination with lower token costs without sacrificing accuracy. However, on VQA-v2, Clotho, IntentQA, and NextQA, the structured prompt uses more tokens but obtains substantially higher accuracy. The increase of token usage is small on VQA-v2 and

Clotho, where the structured prompt uses only 102 and 80 additional tokens, respectively, but yields sizable accuracy gains of 5.0% and 5.5%. The effect is more pronounced on the video benchmarks: the structured prompt uses 744 additional tokens on IntentQA and 7,759 additional tokens on NextQA, but improves accuracy by 10.2% and 3.4%. These results suggest that the CoT-based router often fails to incorporate sufficient task-dependent evidence, especially on audio and video benchmarks where the relevant modality is more expensive to process. As the CoT prompt does not explicitly indicate whether the relevant audio segments or spatio-temporal video clips are available, informative, or necessary, the router tends to favor cheaper text-dominant choices. Although this behavior can reduce token cost in some cases, it also causes the model to miss critical decisive evidence.

Importantly, the CoT-based router is not entirely ineffective. Although CoT outputs provide only weak routing cues, such as reasoning length, token count, and semantic information in the generated rationale, the adaptive modality selection mechanism can still extract some useful features from them and may therefore outperform a text-only interaction strategy. However, the consistent accuracy gap between the CoT variant and the structured modality-routing prompt demonstrates that these implicit signals alone are insufficient for accurate modality routing. The structured prompt provides explicit signals about confidence, self-critique, missing evidence, and task-dependent modality usefulness, enabling the router to distinguish cases in which a lower-cost modality combination is sufficient from those requiring richer multimodal evidence. These results demonstrate that the structured modality-routing prompt is essential for adaptive modality routing.

Table 12: Ablation study on routing prompt design for AIM across eight benchmarks. We compare AIM with a standard CoT prompt and the full structured routing prompt.

Dataset	**CoT Prompt**		**Structured Routing Prompt**	
	Acc. (%)	**# Tokens**	**Acc. (%)**	**# Tokens**
GQA	77.1	6,634	**80.1**	**6,334**
OKVQA	91.1	7,155	**93.8**	**6,775**
VCR	81.0	8,467	**84.2**	**7,788**
VQA-v2	82.7	**6,904**	**87.7**	7,006
Clotho	41.3	**3,715**	**46.8**	3,795
DCASE	49.2	4,042	**51.8**	**3,891**
IntentQA	65.2	**45,008**	**75.4**	45,752
NextQA	75.4	**40,877**	**78.8**	48,636

E.2 Leave-One-Feature-Group-Out Ablation

To evaluate the contribution of each feature group beyond the SHAP analysis in Appendix B.2, we remove one of the four feature groups from the routing features, retrain the correctness classifier and the token-cost regressor on the remaining three feature groups with the same auxiliary datasets, and evaluate the retrained router on all eight benchmarks. We report the accuracy and token cost of each retrained router in Table 13, averaged within each task family. We find that removing any one feature group lowers accuracy and increases token cost in every task family, so all four feature groups contribute to the routing decision. The most important feature group differs across task families. Removing the uncertainty cues causes the largest accuracy drop on VQA, at 0.9%, whereas removing the task-dependent evidence availability cues causes the largest accuracy drop on Audio-QA and Video-QA, at 0.7% and 1.5%. In every task family, the two feature groups with the largest accuracy drops are also the two feature groups with the highest SHAP importance for the correctness classifier in Appendix B.2. Moreover, removing any one feature group lowers accuracy by at most 1.5%, which shows that the router does not depend on one feature group alone. Therefore, the four feature groups are complementary, and each feature group adds information that the other three feature groups do not provide.

Table 13: Accuracy (Acc) and average token cost (Tkn) of AIM after removing one feature group and retraining the router, averaged within each task family. Accuracy is reported as a percentage (%), and Tkn denotes the average per-instance token cost, where K denotes thousands.

Removed Feature Group	VQA		Audio-QA		Video-QA	
	Acc	Tkn	Acc	Tkn	Acc	Tkn
None (AIM)	86.5	6,976	49.3	3,843	77.1	47.2K
Evidence requirement cues	86.0	7,596	48.9	4,183	76.7	49.5K
Uncertainty cues	85.6	7,432	48.9	4,097	76.8	48.1K
Task-dependent evidence availability cues	85.8	7,749	48.6	4,142	75.6	52.4K
Additional modality gain cues	86.1	7,081	48.7	3,994	75.8	49.8K

F Cross-Backbone Generalization: GPT-5.4-nano

To verify whether AIM generalizes across LLM backbones, we replace Gemini 2.5 Flash with GPT-5.4-nano. As GPT-5.4-nano does not support audio or video inputs, we restrict this study to the four VQA benchmarks. All other components, including the graph pipeline, crop tool, debate protocol, router configuration, and feature extraction, are kept identical to the Gemini 2.5 Flash experiments.

Results. As shown in Table 14, AIM achieves the highest accuracy on all four VQA benchmarks under the GPT-5.4-nano backbone while remaining token-efficient. Compared with the highest-accuracy fixed-modality baseline on each dataset, AIM improves accuracy by 7.1% on GQA, 8.8% on OKVQA, 10.4% on VCR, and 9.3% on VQA-v2, with an average accuracy gain of 8.9%. AIM also reduces token cost by 9.4%-55.3% relative to the corresponding highest-accuracy fixed-modality baselines. These results show that adaptive interaction modality selection remains effective after changing the LLM backbone.

Comparison with Gemini 2.5 Flash. Compared with the Gemini 2.5 Flash results in Section 5, AIM's average accuracy gain over the best fixed-modality baseline increases from 2.4% to 8.9%. Under GPT-5.4-nano, all fixed-modality baselines achieve lower accuracy (e.g., X drops from 71.2% to 60.8% on GQA, T + X drops from 84.0% to 64.9% on VQA-v2). AIM's absolute accuracy is also lower than under Gemini 2.5 Flash (e.g., 74.4% vs. 80.1% on GQA). However, adaptive modality selection yields a much larger accuracy gain with GPT-5.4-nano. This suggests that AIM is particularly useful when fixed-modality baselines have lower accuracy, because the router can recover additional correct answers by choosing different modality combinations for different instances. Across both backbones, AIM consistently outperforms every fixed-modality baseline, showing that the routing mechanism generalizes across LLM backbones with different reasoning capabilities.

Why AIM Improves More on GPT-5.4-nano. The larger accuracy gain on GPT-5.4-nano arises since different modality combinations on GPT-5.4-nano tend to answer different types of instances correctly. Even when each fixed combination individually achieves low accuracy, different combinations still answer different subsets of instances correctly. Thus, the set of instances answered correctly by at least one combination can be much larger than the set answered correctly by any single fixed combination. AIM exploits this complementarity by selecting the best modality combination for each instance. Under GPT-5.4-nano, each fixed-combination baseline has lower accuracy individually, but their successes remain complementary: many instances are still answered correctly by at least one combination. This gives AIM more room to improve accuracy through adaptive selection.

G Robustness Analysis

To evaluate the robustness of AIM to an unreliable structured baseline output, we modify the structured baseline output of Phase ① at evaluation time and measure the accuracy and token cost of AIM on all eight benchmarks. The structured baseline output has three components, which are the initial answer with the confidence score, the self-critique with the missing-evidence statement, and the task-dependent modality summaries. We consider four modifications. In Appendix G.1, we

Table 14: Accuracy (Acc) and average token cost (Tkn) across four VQA benchmarks using the GPT-5.4-nano backbone. Accuracy is reported as a percentage (%), and Tkn denotes the average per-instance token cost.

Dataset	One Modality						Two Modalities						All		Adaptive	
	T		G		X		T + G		T + X		G + X		T + G + X		AIM	
	Acc	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**
GQA	58.8	2,093	59.0	2,097	60.8	**1,854**	61.9	2,957	67.3	2,472	61.8	4,958	64.2	5,714	**74.4**	1,912
OKVQA	68.6	3,224	74.5	3,191	75.5	**2,525**	74.9	3,862	78.4	2,987	72.5	5,161	76.8	6,714	**87.2**	2,707
VCR	43.1	10,277	45.1	10,462	47.1	9,521	47.6	11,319	47.1	9,179	49.0	12,741	52.8	19,515	**63.2**	**8,723**
VQA-v2	51.0	2,864	56.9	2,942	62.7	**2,741**	58.8	3,786	64.9	3,379	64.1	5,029	62.0	6,675	**74.2**	2,832

mask the key evidence words in the missing-evidence statement, or shift the confidence score of the initial answer and the coordinates, confidence scores, and relevance scores of the task-dependent modality summaries. In Appendix G.2, we make the baseline agent fully confident in an incorrect initial answer by forcing the confidence score to its maximum, or move the regions, segments, or clips in the task-dependent modality summaries to wrong locations. In all cases, we use the router trained on the unmodified auxiliary datasets without retraining, so the results measure how the trained router responds to an unreliable input. We report the results in Table 15, where the last column is the result of AIM without modification from Table 2.

Table 15: Accuracy (Acc) and average token cost (Tkn) of AIM when one component of the structured baseline output is masked, shifted, or made incorrect at evaluation time, without retraining the router. The last column is the result of AIM without modification. Accuracy is reported as a percentage (%), and Tkn denotes the average per-instance token cost, where K denotes thousands.

Dataset	Incomplete or Perturbed Output				Incorrect Output				No Modification	
	Masked Statement		Shifted Scores		Wrong Confidence		Mislocated Regions			
	Acc	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**	**Acc**	**Tkn**
GQA	80.1	6,392	80.1	6,383	79.6	6,670	79.4	6,550	80.1	6,334
OKVQA	93.6	6,734	93.7	6,855	93.6	6,881	93.5	6,983	93.8	6,775
VCR	84.1	7,625	84.1	8,207	83.8	7,746	83.8	8,152	84.2	7,788
VQA-v2	87.7	6,991	87.7	7,468	87.2	7,358	86.8	7,704	87.7	7,006
Clotho	46.3	3,817	46.7	3,877	46.5	3,760	46.2	4,018	46.8	3,795
DCASE	51.7	3,841	49.6	3,949	51.4	3,794	50.9	4,492	51.8	3,891
IntentQA	75.0	46.0K	74.3	51.6K	75.3	45.5K	74.6	51.9K	75.4	45.8K
NextQA	78.8	49.0K	78.8	48.4K	78.7	48.5K	78.0	54.4K	78.8	48.6K

G.1 Incomplete or Perturbed Outputs

To evaluate AIM when the structured baseline output is incomplete, we mask the key evidence words in the missing-evidence statement, such as the nouns that name the needed evidence, so the router no longer knows which evidence the baseline agent considers missing. We find that the masked statement has little effect on AIM. The accuracy drops by only 0.16% on average, and the token cost changes by at most 2.1% on any benchmark, since the missing-evidence statement is only one of the evidence requirement cues and the router still receives the question type and the other three feature groups. To evaluate AIM when the structured baseline output is perturbed, we add an independent Gaussian perturbation $\epsilon \sim \mathcal{N}(0, 0.1^2)$ to the confidence score of the initial answer and to every normalized numerical field of the task-dependent modality summaries, which are the bounding box coordinates for VQA, the start and end timestamps for Audio-QA and Video-QA, the confidence score of each summary, and the relevance score of each caption. As a result, the uncertainty cues and the evidence availability cues are extracted from perturbed values. We find that the shifted scores have a larger but still small effect. The accuracy drops by 0.45% and the token cost increases by 3.7% on average, and six of the eight benchmarks lose at most 0.1% accuracy. The two exceptions are DCASE, where the accuracy drops by 2.2%, and IntentQA, where the token cost increases by 12.7%. This difference stems from the feature groups that each router relies on. According to the

SHAP analysis in Appendix B.2, the routers of Audio-QA and Video-QA rely most on the evidence availability cues and the additional modality gain cues, and both feature groups are extracted from the shifted scores of the summaries, so the shifted scores change more routing decisions in these two task families. Nevertheless, under both modifications, AIM remains more accurate than the most accurate fixed combination on average in every task family. Therefore, an incomplete or perturbed structured baseline output changes the accuracy of AIM by less than 0.5% on average.

G.2 Incorrect Outputs

To evaluate AIM when the baseline agent is confident in a wrong answer, we set the confidence score to its maximum of 1.0 for every instance whose initial answer is incorrect and keep the other uncertainty cues unchanged. If the router trusted the confidence score alone, the router would select the text-only combination for these instances and answer incorrectly. We find that the wrong confidence has little effect on AIM. The accuracy drops by only 0.31% and the token cost increases by only 0.9% on average, since the correctness classifier also uses the hedge terms, the self-critique, and the other three feature groups, which still identify the instances that need additional evidence. To evaluate AIM when the task-dependent modality summaries point to the wrong part of the input, we randomly move one third of the regions, segments, or clips to locations that do not overlap the original locations in space or time, and extract the evidence availability cues again from the moved locations. We find that the mislocated regions have the largest effect among the four modifications, but AIM remains accurate. The accuracy drops by 0.68% on average, whereas the token cost increases by 8.5% on average, and the token increase is larger on Audio-QA and Video-QA, at 10.7% and 12.6%, than on VQA, at 5.3%. This result follows from the accuracy-first objective of AIM, which minimizes token cost only among the combinations expected to answer correctly. When the evidence availability cues become unreliable, AIM keeps accuracy by spending more tokens, and the increase is larger on Audio-QA and Video-QA, whose routers rely more on the evidence availability cues according to the SHAP analysis in Appendix B.2. Nevertheless, under both modifications, AIM remains at least as accurate as the most accurate fixed combination on all eight benchmarks. Therefore, AIM remains accurate even when the baseline agent is confident in a wrong answer or points to the wrong part of the input.

H Comparison with Single-group Adaptive Routers

Section 5.3 compares AIM with four single-group adaptive routers, each of which selects the modality combination with the routing features of a single feature group. In this section, we first list the exact routing features that each single-group adaptive router uses in Appendix H.1. We then compare the robustness of AIM and the four single-group adaptive routers in Appendix H.2, where we flip, mask, or shift the routing features of each single-group adaptive router and measure how much the accuracy of each method drops.

H.1 Details of the Single-group Adaptive Routers

Each single-group adaptive router selects the modality combination with the routing features of a single feature group, whereas the adaptive modality router of AIM uses all four feature groups. Question-type routing uses only the question type features, which are the evidence requirement cues without the missing-evidence statement and include the question categories, the referenced entities, and the question demand flags, such as whether the question requires counting or spatial reasoning. Similarly, missing-evidence-only routing uses only the missing-evidence statement, which states what additional evidence would confirm or refute the initial answer, and the task-dependent modality evidence availability cues, which estimate whether the missing evidence is present in the input. Moreover, confidence-only routing uses only the uncertainty cues, which include the confidence score, hedge terms, generic priors, and the self-critique. Finally, the cost-aware heuristic uses only the additional modality gain cues, which estimate how much new evidence each modality adds and the risk of adding it. We give the exact feature names used by each single-group adaptive router for each task family in Table 16, following the feature tables in Appendix B.1.

Table 16: Routing features used by each single-group adaptive router. The features in the "All" row of a router are used in every task family, and the features in the "VQA", "Audio-QA", and "Video-QA" rows are used only in that task family. The notation `a_{b, c}` is short for the two features `a_b` and `a_c`, and $k \in \{0, 1, 2\}$ is the index of a region, segment, or clip. The feature names are the same as in Tables 6, 7, and 8.

Single-group Adaptive Router	Task Family	Features Used
Question-type routing	All	target_nouns, target_nouns_len, target_nouns_{0,1,2,3}, demand, demand_question_category, demand_question_category_len, demand_question_category_{0,1}, demand_target_noun_count, demand_relation_cue_count
	VQA	demand_requires_{localization, interaction, spatial_reasoning, color_or_texture, ocr, counting}
	Audio-QA	demand_requires_{speech_content, sound_identification, temporal_reasoning, emotion_or_tone, speaker_identification, counting}
	Video-QA	demand_requires_{action_recognition, temporal_reasoning, spatial_reasoning, appearance_recognition, ocr, counting}
Missing-evidence-only routing	All	text_evidence_to_confirm
	VQA	image, image_crops, image_crop_k, image_crop_k_box, image_crop_k_box_{0,1,2,3}, image_crop_k_{confidence, caption, caption_relevance, contains_targets, matched_nouns}, image_crop_{0,1}_matched_nouns_{len, 0}, image_crop_0_matched_nouns_1, router_best_caption_relevance, router_crop_relevance_{mean, std, min, max, distribution}, router_low_relevance_crop_ratio, router_crop_confidence_variance, router_crop_rel_dist_{very_low, low, medium, high}
	Audio-QA	audio, audio_clips, audio_clip_k, audio_clip_k_{start, end, confidence, description, description_relevance, contains_targets, matched_nouns}, audio_clip_{0,1}_matched_nouns_{len, 0}, audio_clip_0_matched_nouns_1, router_best_description_relevance, router_clip_relevance_{mean, std, min, max, distribution}, router_low_relevance_clip_ratio, router_clip_confidence_variance, router_clip_rel_dist_{very_low, low, medium, high}
	Video-QA	video, video_clips, video_clip_k, video_clip_k_{start, end, confidence, description, description_relevance, contains_targets, matched_nouns}, video_clip_{0,1}_matched_nouns_{len, 0}, video_clip_0_matched_nouns_1, router_best_description_relevance, router_clip_relevance_{mean, std, min, max, distribution}, router_low_relevance_clip_ratio, router_clip_confidence_variance, router_clip_rel_dist_{very_low, low, medium, high}
Confidence-only routing	All	text, text_initial_answer, text_confidence, text_reasoning, text_signals, text_signals_{has_hedges, hedge_terms_found, cites_visual_evidence, generic_priors, prior_terms_found, prior_terms_len, prior_terms_0}, text_self_critique, text_self_critique_neg_reasoning, router_text_confidence, router_confidence_calibration_flag
Cost-aware heuristic	All	router, router_graph_{target_node_coverage, subgraph_confidence, relation_match, irrelevant_node_ratio, spurious_relation_risk, clutter_risk, attribute_loss_risk, marginal_relation_gain}, similarity_q_{initial, critique, evidence, all}

Continued on next page

Single-group Adaptive Router	Task Family	Features Used
	VQA	router_image_marginal_coverage_gain, router_text_image_overlap, router_image_{irrelevant_entity_risk, distracting_context_risk}, router_image_graph_overlap, similarity_q_caption, similarity_image_{cropmean, cropmax}
	Audio-QA	router_audio_marginal_coverage_gain, router_text_audio_overlap, router_audio_{irrelevant_content_risk, distracting_content_risk}, router_audio_graph_overlap, similarity_q_description
	Video-QA	router_video_marginal_coverage_gain, router_text_video_overlap, router_video_{irrelevant_entity_risk, distracting_context_risk}, router_video_graph_overlap, similarity_q_description, similarity_video_{clipmean, clipmax}

H.2 Robustness to Incorrect Routing Features

In this section, we compare the robustness of AIM and the four single-group adaptive routers to incorrect routing features. For each single-groupr adaptive router, we flip, mask, or shift the feature group used by the single-group adaptive router and apply the same change to AIM. We report the accuracy and token cost of each router before and after each change in Table 17, averaged over the eight benchmarks.

To evaluate the robustness to incorrect question type features, we flip the question type features. We find that the accuracy of question-type routing drops by 1.1%, whereas the accuracy of AIM drops by only 0.3%, since question-type routing relies on the question type features alone, whereas AIM still uses the missing-evidence statement and the other three feature groups. To evaluate the robustness to incorrect confidence features, we shift the confidence features in the uncertainty cues by a Gaussian perturbation $\epsilon \sim \mathcal{N}(0, 0.1^2)$. We find that the accuracy of confidence-only routing drops by 0.2% and the accuracy of AIM drops by 0.1%, but confidence-only routing is less accurate than AIM even before the change, at 73.0% compared with 74.8%.

To evaluate the robustness to incorrect missing-evidence features, we mask the missing-evidence features, which are the missing-evidence statement and the evidence availability cues. We find that the accuracy of missing-evidence-only routing drops by 1.7% while its token cost decreases from 18,214 to 17,823, which indicates under-fusion. In contrast, the accuracy of AIM drops by only 0.4% while the token cost of AIM increases from 16,247 to 17,491, since the accuracy-first objective of AIM minimizes token cost only among the combinations expected to answer correctly, and AIM therefore keeps accuracy by spending more tokens when the evidence availability cues become unreliable.

To evaluate the robustness to incorrect additional modality gain cues, we shift the additional modality gain cues by the same Gaussian perturbation. We find that the accuracy of AIM stays at 74.8%, and only the token cost of AIM increases from 16,247 to 17,003. The additional modality gain cues are used to drop a modality that adds little new evidence, and therefore AIM drops fewer modalities and uses more tokens when these cues are shifted. However, the other three feature groups still decide which modalities the instance requires, so the accuracy of AIM stays unchanged, whereas the accuracy of the cost-aware heuristic drops by 0.3% from an already lower 72.3%.

Therefore, the two most accurate single-group adaptive routers, question-type routing and missing-evidence-only routing, are the least robust, and the other two single-group adaptive routers are more stable but less accurate. Under each of the four changes, the accuracy of AIM drops less than the accuracy of the corresponding single-group adaptive router, since AIM uses all four feature groups and does not rely on one feature group.

I FaithScore Case Studies

In this section, we present three VQA cases that show why a FaithScore above a sufficient level is enough and why maximizing FaithScore is not desirable. FaithScore counts a claim as grounded only

Table 17: Accuracy (Acc) and average token cost (Tkn) of each single-group adaptive router and of AIM before $\rightarrow$ after the feature group used by the single-group adaptive router is flipped, masked, or shifted, averaged over the eight benchmarks. Accuracy is reported as a percentage (%), and Tkn denotes the average per-instance token cost.

Changed Features	**Method**	**Acc**	**Tkn**
Question type features	Question-type routing	73.7 $\rightarrow$ 72.6	17,784 $\rightarrow$ 18,766
	AIM	74.8 $\rightarrow$ 74.5	16,247 $\rightarrow$ 16,903
Confidence features	Confidence-only routing	73.0 $\rightarrow$ 72.8	16,707 $\rightarrow$ 17,346
	AIM	74.8 $\rightarrow$ 74.7	16,247 $\rightarrow$ 16,377
Missing-evidence features	Missing-evidence-only routing	73.5 $\rightarrow$ 71.8	18,214 $\rightarrow$ 17,823
	AIM	74.8 $\rightarrow$ 74.4	16,247 $\rightarrow$ 17,491
Additional modality gain cues	Cost-aware heuristic	72.3 $\rightarrow$ 72.0	15,693 $\rightarrow$ 16,981
	AIM	74.8 $\rightarrow$ 74.8	16,247 $\rightarrow$ 17,003

when the claim is directly observable in the input, so a correct inference drawn from the input lowers FaithScore in the same way as a hallucination. However, many questions ask for something that is not directly observable, such as the role of a person or the kind of a dish, and can only be answered by an inference from observed evidence. A correct answer to such a question therefore needs both observed evidence and an inference drawn from that evidence. The observed evidence keeps the answer from being pure speculation, and the inference is what the question asks for. As a result, a FaithScore above a sufficient level shows that the answer is anchored in observed evidence, whereas a higher FaithScore would remove the inference that the question requires.

We show the image, the question, the answer of AIM, the observed evidence, the penalized inference, and the FaithScore of each case in Fig. 10. In the first case from OKVQA, the question asks why a person on the field is dressed in orange and black. AIM observes that the person wears orange and black whereas the players wear blue, infers that the person has a different role from the players, and answers "referee". In the second case from VCR, the question asks why Riley looks smug. AIM observes that Riley looks at Jackie with a slight smile, infers approval from the smile and the gaze, and answers that Riley approves and agrees with what Jackie is saying. In the third case from VQA-v2, the question asks what kind of pie is shown. AIM observes that the pie is brown and has lines on top, infers that a brown pie with lines on top is an apple pie, and answers "apple". In all three cases, the answer is correct, the observed evidence is visible in the image, and the inference is required, since the role of a person, the attitude of a person, and the kind of a pie are not directly observable. However, FaithScore counts each inference as an ungrounded claim, and the three cases receive FaithScores of only 0.70, 0.69, and 0.67. Therefore, a low FaithScore in these cases reflects an inference that the question requires rather than a hallucination. This result also explains why AIM obtains higher accuracy than T + X on VQA-v2 with a slightly lower FaithScore in Section 5.2.



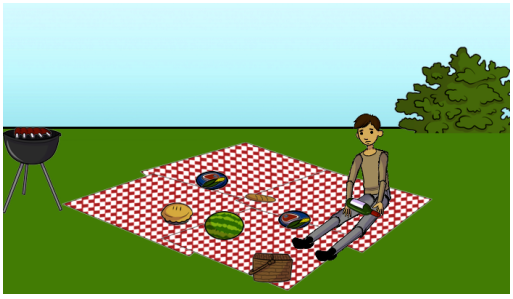
Question: What would be the reason a person on the field is dressed in orange and black?
Answer of AIM: referee
Observed evidence: The person wears orange and black, and the players wear blue.
Penalized inference: The person has a different role from the players and is therefore a referee.
FaithScore: 0.70

(a) OKVQA



Question: Why does Riley look so smug?
Answer of AIM: Riley approves and agrees with what Jackie is saying.
Observed evidence: Riley looks at Jackie with a slight smile.
Penalized inference: The slight smile and direct gaze indicate approval.
FaithScore: 0.69

(b) VCR



Question: What kind of pie is it?
Answer of AIM: apple
Observed evidence: The pie is brown and has lines on top.
Penalized inference: A brown pie with lines on top is an apple pie.
FaithScore: 0.67

(c) VQA-v2

Figure 10: Three VQA cases in which AIM answers correctly but receives a low FaithScore. Each case shows the image, the question, the answer of AIM, the evidence observed in the image, the inference that FaithScore penalizes, and the FaithScore.

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 abstract and introduction claim that AIM adaptively selects interaction modality for multimodal MAD, which improves accuracy while remaining token-efficient. These claims are directly supported by the method description in Section 4 and the empirical results in Section 5.

   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 6, including the one-time offline cost of training a router for each task family and the need to run all seven modality combinations on auxiliary training data. It also identifies future directions, including online adaptation and joint temporal/spatial compression, which clarify the current scope of AIM.

   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 present theoretical theorems, formal guarantees, or proof-based claims.

   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: Sections 4 and 5 disclose the information needed to reproduce the results, including the AIM pipeline, structured baseline output, seven candidate modality combinations, routing decision rule, temporal compression strategy, datasets, baselines, evaluation metrics, LLM backbone, decoding settings, and shared debate protocol. Appendix A and Appendix B further specify the router-training/evaluation separation, task-specific router models and hyperparameters, feature encoders and PCA settings, temporal compression details, stability protocol, and comprehensive routing-feature inventories. The supplementary code provides the implementation scripts and reproduction instructions for the main experiments.

   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: Our code is available at `https://github.com/Fanwei100/AIM`. The code package includes a README, environment specification, dataset preparation instructions, and scripts for training the router and reproducing for AIM and the fixed-modality baselines. The experiments use existing public datasets.

   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: Section 5 and Appendix A specifies the experimental setup needed to interpret the results, including the eight evaluation benchmarks, auxiliary router-training datasets, fixed-modality baselines, evaluation metrics, backbone, decoding parameters, shared debate protocol, and dataset-specific temporal compression ratios. Appendix A further provides the training details for the lightweight router, including the correctness-classifier and token-cost-regressor model families, losses, estimator counts, tree depths, learning rates, routing thresholds, under-fusion margins, feature encoders, PCA dimensions, batch sizes, hardware, and the grid-search procedure used to select hyperparameters on validation data, with evaluation datasets kept disjoint from router training and threshold tuning.

   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: The stability analysis in Appendix A reports three repeated end-to-end runs with the same trained router and decoding settings, using empirical mean and one-standard-deviation statistics for accuracy and token cost across all eight benchmarks, confirming that results are stable despite minor LLM API nondeterminism.

   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: Appendix A states that router models are trained on a single NVIDIA RTX 4090 GPU with lightweight resource requirements. The LLM backbone is accessed via API (Appendix A).

   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 work uses only publicly available academic benchmarks with proper citations, and does not introduce human-subject experiments, private user data, scraped sensitive datasets, or a new high-risk generative model.

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: [N/A]

Justification: This work does not have any direct societal impact.

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 does not release pre-trained language models, image generators, or scraped datasets. The router models are lightweight gradient-boosted trees that do not pose misuse 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: All datasets, pre-trained models, router implementations, and tools are cited with their original references in Section 5 and Appendix A.

    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: [N/A]

    Justification: The paper does not release new datasets or pre-trained models as standalone assets.

    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 or human subjects research.

    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 human subjects research.

    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 uses Gemini 2.5 Flash and GPT-5.4-nano as the LLM backbones for the MAD agents. Section 5 specifies the model and decoding settings, including temperature 0.0 and a maximum output length of 512 tokens. The LLM is used as the standard agent reasoning engine in the core inference pipeline. The paper's novelty lies in the adaptive modality-routing framework built on top of this LLM backbone.

    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.