HAMV: A Heterogeneous Adaptive Multi-Model Verification Framework for Efficient and Reliable Fact-Checking

Anonymous ACL submission

Abstract

Large language models often exhibit hallucinations, while single-model self-verification lacks the knowledge and reasoning diversity required to achieve both reliability and cost-efficiency. We propose HAMV, a framework that reframes multi-model collaboration as a cross-model sparse expert scheduling problem. Inspired by the Mixture-of-Experts (MoE) paradigm, HAMV employs a dynamic routing mechanism to adaptively assign roles (generation, evaluation, verification, and aggregation) based on task and model profiles. It incorporates Dempster–Shafer-based confidence fusion to trigger conditional verification for controlled computational cost. On HalluQA and TruthfulQA, HAMV consistently outperforms representative baselines across varying budgets. Further analysis confirms that dynamic scheduling mitigates position sensitivity, validating the effectiveness of modeling factual verification as a structured scheduling problem.

1 Introduction

As LLMs are increasingly used for open-domain reasoning and complex verification, ensuring factual consistency and reasoning reliability has become a key deployment challenge (Huang et al., 2025a; Ji et al., 2023). Existing approaches mostly rely on a single model for generation, judgment, or verification, but limited knowledge coverage, reasoning biases, and correlated errors make it difficult to achieve accuracy, robustness, and efficiency simultaneously (Wang et al., 2022).

To address these issues, recent work explores multi-model or multi-agent collaboration and cross-verification, improving factual reliability via role specialization and diversified reasoning paths (Du et al., 2023; Wu et al., 2024a). However, most approaches rely on static or rule-based coordination, without explicitly modeling task heterogeneity or complementary model capabilities, which often leads to redundant reasoning or ineffective collaboration (Huang et al., 2025b).

MoE improves model efficiency via learnable routing and sparse activation, invoking only input-relevant submodules to reduce computation while maintaining or even improving performance (Shazeer et al., 2017). Prior studies (Jiang et al., 2024; Wu et al., 2024b) show that this routing plus sparse activation is particularly effective in complex reasoning. Although mostly studied within single models, the same principle can be extended to heterogeneous reasoning models: treating each model as an expert enables coarse-grained, capability-oriented selection and collaborative decision-making (Huang et al., 2025b; Jitkrittum et al., 2025).

Building on these observations, we propose HAMV, a heterogeneous adaptive multi-model verification framework for factual verification. It uses a dynamic routing mechanism to adaptively select the most suitable models or model combinations for reasoning and adjudication based on task characteristics and verification requirements. Compared to static multi-model schemes, HAMV reduces unnecessary model invocations while maintaining strong verification performance. Experiments across multiple factual verification and reasoning benchmarks demonstrate its stable and consistent improvements, validating the effectiveness of the proposed cross-model sparse routing mechanism.

2 Related Work

2.1 Hallucination Handling

Large language models (LLMs) often produce hallucinations, generating content that appears plausible but is factually incorrect (Huang et al., 2025a; Ji et al., 2023). Existing research has focused on two directions: hallucination detection and mitigation.

Detection methods include uncertainty estimation such as semantic entropy (Farquhar et al.,

2024), multi-sample consistency checking as in SelfCheckGPT (Manakul et al., 2023), and self-generated evidence–based verification as in SAC³ (Zhang et al., 2023). Mitigation approaches improve factuality through retrieval-augmented generation (Lewis et al., 2020), data refinement or fine-tuning (Touvron et al., 2023), and optimized decoding (Chuang et al., 2024).

To enhance self-correction, self-verification mechanisms have been proposed. Inspired by Chain-of-Thought (CoT) reasoning (Wei et al., 2022), models can "reflect" on their outputs. Iterative revision methods such as CoVe (Dhuliawala et al., 2024) and Self-Refine (Madaan et al., 2023) refine outputs step by step. Toolformer (Schick et al., 2023), PAL (Gao et al., 2023), and CRITIC (Gou et al., 2023) incorporate external tools for stronger verification, while Reverse verification (Weng et al., 2023) and Chain-of-Note (CoN) (Yu et al., 2024) enhance interpretability. Some studies mitigate hallucinations via data or model-level interventions. ANAH-v2 (Gu et al., 2024) relies on large-scale annotations and self-training to improve HalluQA, while NoVo (Ho et al., 2025) suppresses hallucinations at inference time by leveraging attention-head redundancy and performs well on TruthfulQA. However, these methods rely primarily on a single backbone model throughout the verification process, and therefore remain limited in computational efficiency and adaptability.

2.2 Multi-Model Collaboration

To overcome the performance and adaptability limits of single-model self-verification, multi-model collaboration is widely recognized as a key strategy to enhance LLM reliability. Multi-agent systems coordinate multiple models through role division for complex tasks. For example, debate-based factual enhancement (Du et al., 2023) and AutoGen (Wu et al., 2024a) leverage multi-model collaboration to improve accuracy and consistency.

However, multi-model systems introduce substantial computational and communication costs. Adaptive strategies such as Adaptive-RAG (Jeong et al., 2024) and In-context Autoencoder (Ge et al., 2023) mitigate these overheads by selectively triggering retrieval or compressing contextual information. FrugalGPT (Chen et al., 2023) further manages reasoning costs via a cost-ranked model cascade.

Despite these advances, existing approaches are limited in task diversity and flexible scheduling, motivating finer-grained, dynamic multi-model mechanisms that exploit complementary capabilities efficiently.

2.3 Mixture-of-Experts Architectures

Mixture-of-Experts improves inference efficiency by sparsely activating expert sub-networks. In classic MoE, each input is dynamically routed to a subset of experts via a gating network, saving computation while maintaining performance (Shazeer et al., 2017). The Switch Transformer (Fedus et al., 2022) exemplifies this, activating only a few experts in a large Transformer for efficient inference.

MoE has been extended to multimodal and large-scale language models. Mixtral 8×7B (Jiang et al., 2024) uses sparse experts to boost cross-task generalization, and DeepSeek-VL2 (Wu et al., 2024b) applies sparse activation in multimodal reasoning. These studies show that "routing plus sparse activation" not only improves single-model efficiency but also inspires cross-model collaboration: treating models or submodules as experts with complementary capabilities and selecting them dynamically based on task characteristics. This principle provides an important reference for the heterogeneous adaptive routing mechanism in HAMV.

2.4 Summary

Existing studies have advanced single-model hallucination handling, multi-model collaboration, and sparse expert mechanisms. Yet, single-model approaches remain limited by knowledge coverage and biases, while multi-model methods face challenges in task adaptability and scheduling. Although MoE improves computational efficiency, it has not been systematically applied to heterogeneous-model factual verification.These limitations motivate HAMV, which combines dynamic multi-model scheduling with MoE-inspired principles to achieve efficient, reliable, and adaptive factual verification.

3 The HAMV Framework

The HAMV framework balances verification accuracy and computational cost via adaptive scheduling. It assigns heterogeneous models to task stages to enhance factual consistency and triggers high-cost verification conditionally based on confidence levels to save resources. The following sections describe HAMV's architecture and core algorithms.

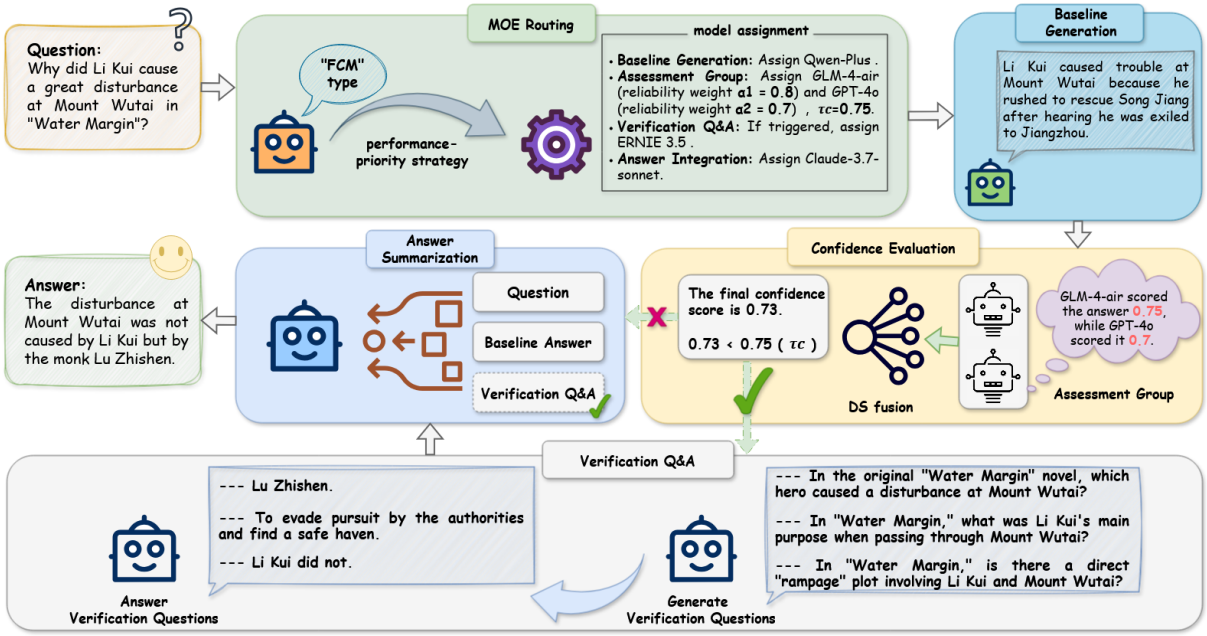

Figure 1: Overall architecture of the HAMV framework.

3.1 Overall Architecture

In HAMV, the set of models forms the model team M_{team}, dynamically constructed from the model pool P for each query Q:

$$M_{\text{team}} = \{m_{\text{Gen}}, m_{\text{Eval}}, m_{\text{Ver}}, m_{\text{Sum}}\}. \quad (1)$$

Here, m_{Gen} generates the baseline answer; m_{Eval} assigns a scalar confidence score; m_{Ver} handles verification when confidence is low; and m_{Sum} aggregates results and produces the final answer.

As illustrated in Figure 1, HAMV uses a four-stage sequential workflow. Compared to single-model verification, it enables multi-model collaboration with adaptive triggering, leveraging knowledge complementarity while optimizing computational cost.

MoE Adaptive Routing Module. Serving as the entry point and scheduler, this module receives user queries, performs semantic parsing and task type identification, and allocates the four role models from P. It also sets evaluation reliability weights α and the adaptive triggering threshold τ_c.

Baseline Generation Module. Executed by m_{Gen}, it outputs the structured initial answer, providing the foundation for evaluation and verification.

Confidence Evaluation and Verification Q&A Module. The evaluation model m_{Eval} scores the baseline answer. Dempster–Shafer (DS) evidence fusion aggregates confidence estimates into a unified belief. If it falls below τ_c, m_{Ver} is triggered to generate verification questions and answers.

Conflict Analysis and Answer Aggregation Module. Executed by m_{Sum}, this module analyzes consistency between the baseline and verification answers, producing the final high-reliability output.

Example workflow. For the query "Why did Li Kui cause a disturbance at Mount Wutai?", the routing module first classifies it as a factual verification task and constructs the model team as follows: Qwen-Plus (m_{Gen}), GLM-4-air and GPT-4o ($m_{\text{Eval1}}, m_{\text{Eval2}}$) with reliability weights 0.8 and 0.7, ERNIE 3.5 (m_{Ver}), and Claude-3.7-Sonnet (m_{Sum}), with an adaptive threshold $\tau_c = 0.75$. The generation stage outputs the baseline answer; the evaluation stage produces an aggregated confidence of 0.73, which is below the threshold, triggering the verification process. In the aggregation stage, the final answer is corrected based on verification evidence: "The disturbance at Mount Wutai was not caused by Li Kui, but by the Flowery Monk Lu Zhishen."

3.2 MoE Routing Algorithm

The MoE routing algorithm is HAMV's core scheduler, enabling optimal role assignment in a heterogeneous model environment while balancing accuracy, scalability, and computational cost. It comprises four main components:

Heterogeneous Model Pool Construction. HAMV builds the model pool P based on three heterogeneity principles: **architectural:** covering diverse training paradigms, reasoning mechanisms, and representations to enhance knowledge and reasoning diversity; **capability:** including high-performance and lightweight models to balance accuracy and cost; **cost:** dividing models into flagship and economical tiers based on API pricing to enable flexible scheduling. According to the routing strategy, a candidate set P' is selected from P: under performance-oriented strategy, $P' = P$; under cost-constrained strategy, only economical models are kept.

Multi-Dimensional Model Profiling. To support role assignment, HAMV constructs a multi-dimensional profile for each model, including:

- **Capabilities:** Characterized by four core abilities aligned with the verification pipeline stages: generation, evaluation, verification, and synthesis. **Generation ability** measures the model's capacity to directly produce correct answers given the original input, evaluated by answer accuracy. **Evaluation ability** reflects the model's skill in judging the correctness of candidate answers, assessed on baseline answers labeled as true or false. **Verification ability** quantifies the model's capability to generate and answer verification questions around error points, measured by both verification question coverage and answer accuracy. **Synthesis ability** indicates the model's competence in integrating conflicting or multi-source evidence to produce correct conclusions, evaluated by the accuracy of the final answer after multi-evidence reasoning.
- **Cost:** Estimated based on API pricing standards to account for economic considerations in model selection.
- **Error diversity:** Measures variation in decision errors among team models. Higher values indicate stronger complementarity and greater potential for error correction.

Task Type Classification and Dynamic Routing Strategy. Dynamic routing classifies the input task and sets the expert selection strategy S. Under the performance-oriented strategy, the highest-capability models are selected; under the cost-constrained strategy, only economical models are retained. Strategy S defines the candidate set composition and role allocation constraints.

Role Assignment Based on Multi-Dimensional Profiles. Once the candidate pool P' is determined, roles are assigned in the following order:

- m_{Gen}: select the model in P' with the highest generation ability for the current task type.
- m_{Ver}: select from $P' \backslash \{m_{\text{Gen}}\}$ the model with the highest verification ability for the current task.
- m_{Eval}: Select the top two models in $P' \setminus \{m_{\text{Gen}}\}$ with the highest evaluation ability to form the evaluation team. Reliability weights α_i are set according to each model's historical accuracy $\text{acc}_{m,T}$, and the system's adaptive threshold is defined as $\tau_c = (\alpha_{\text{Eval}_1} + \alpha_{\text{Eval}_2})/2$.
- m_{Sum}: select a model from P' with overall balanced capability based on the team's aggregate error diversity $\text{ErrDiv}_{\text{team}}$.

3.3 DS-Based Confidence Fusion and Conditional Verification

HAMV leverages Dempster–Shafer evidence theory (Dempster, 2008) to model uncertainty from heterogeneous evaluation models in a unified framework. The procedure is as follows:

(1) **BPA construction:** Each evaluation model produces a discounted basic probability assignment (BPA)

$$\begin{aligned} m_i'(C) &= \alpha_i \, s_i, \\ m_i'(I) &= \alpha_i \, (1 - s_i), \\ m_i'(\Theta) &= 1 - \alpha_i. \end{aligned} \tag{2}$$

(2) **Conflict coefficient:** The conflict between two evidence sources is computed as

$$K = m_1'(C)m_2'(I) + m_1'(I)m_2'(C). \tag{3}$$

(3) **Evidence fusion:** The combined belief for correctness is given by

$$m_{\text{c}}(C) = \frac{m_1'(C)[m_2'(C) + m_2'(\Theta)] + m_1'(\Theta)m_2'(C)}{1 - K} \tag{4}$$

The combined belief $m_{\text{c}}(C)$ is compared with threshold τ_c for conditional verification. If it exceeds τ_c, deep verification is skipped; otherwise, verification questions are triggered. This ensures factual reliability while optimizing computational resources and avoiding unnecessary checks.

4 Experiments and Results

This section evaluates HAMV in hallucination mitigation and cost control, analyzing the contributions of multi-model collaboration (Multi), MoE routing (MoE), and DS evidence fusion (DS). Experiments include overall performance comparison, ablation studies, position sensitivity analysis, and failure mode investigation, validating effectiveness, component contributions, key routing design assumptions, and framework limitations.

4.1 Experimental Setup

4.1.1 Datasets and Model Pool

Datasets and Splits. Experiments are conducted on two factual question-answering benchmarks. HalluQA (Cheng et al., 2023), a hallucination-sensitive Chinese dataset, evaluates factual consistency and robustness in Chinese settings. TruthfulQA (Lin et al., 2022), an adversarial English benchmark in the MC1 multiple-choice setting, assesses model robustness against misleading questions.

For each dataset, samples are randomly split into a development set (30%) for capability assessment and model profiling, and an evaluation set (70%) for performance reporting. All results are reported on the evaluation set to ensure fair comparison.

Model Pool. We construct a heterogeneous model pool covering different performance and cost tiers to evaluate HAMV's collaborative capabilities under varying budget constraints. The pool includes:

High-performance models: ERNIE 4.0, GLM-4-plus, Qwen-Max, GPT-4.1, Claude-3.7-Sonnet

Moderate- and low-cost models: ERNIE 3.5, GLM-4-air, Qwen-Plus, GPT-4o

Capability Evaluation and Multi-Dimensional Model Profiling. Based on the four-dimensional capability assessment in Section 3.2, model performance is quantified to construct a $9 \times 7 \times 4$ performance matrix for MoE routing. The task set $T \in \{P_1, \ldots, P_7\}$ spans seven categories: AI self-identity (AI), debunking absolutes and universals (DAU), myth versus reality discrimination (MFR), urban legends and pseudoscience detection (ULP), fact-checking and misconception correction (FCM), literature and arts knowledge (LAK), and logic-based reasoning, wordplay, and paradox handling (LWP). These seven categories support MoE routing decisions and task–model matching.

4.1.2 Evaluation Metrics

To systematically assess the performance of the HAMV framework, three types of metrics are employed:

Accuracy (Acc): the proportion of correct responses provided by the model or framework, reflecting factual consistency:

$$\text{Acc} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{M(x_i) = y_i\}, \tag{5}$$

where N is the total number of samples, $M(x_i)$ is the model prediction, y_i is the ground-truth label, and $\mathbb{I}\{\cdot\}$ denotes the indicator function.

Cost ($\times 10^{-3}$USD): the cumulative API call expenses, representing the economic efficiency of the verification process.

Trigger Rate (TR): the proportion of samples that activate the verification chain, measuring how frequently verification is triggered:

$$\text{TR} = \frac{\text{Verification-triggered cases}}{\text{Total evaluation samples}} \times 100\%. \tag{6}$$

Correction Rate (CR): the proportion of successfully corrected cases among verification-triggered samples, reflecting decision accuracy:

$$\text{CR} = \frac{\text{Corrected cases}}{\text{Verification-triggered cases}} \times 100\%. \tag{7}$$

4.2 Performance Comparison Experiments

This section compares HAMV with representative baselines. CoVe is included as a single-model self-verification method using the best-performing configuration per dataset, while FrugalGPT represents inference-time dynamic model selection for performance–cost optimization. In addition, NoVo is evaluated following its original setting with Mistral-7B, motivated by its strong performance on TruthfulQA. HAMV is assessed under H-p (high-performance) and C-c (cost-constrained) settings to examine performance–cost trade-offs. Results on HalluQA and TruthfulQA are shown in Figures 2.

HAMV (H-p) achieves 84.44% accuracy on HalluQA, surpassing the strongest CoVe baseline (80.67%) by a substantial margin while reducing inference cost by approximately 24%. On TruthfulQA, it attains 84.94% accuracy—comparable to CoVe—but at only around 10% of the cost, indicating that near-flagship performance can be maintained with substantially lower computational overhead.

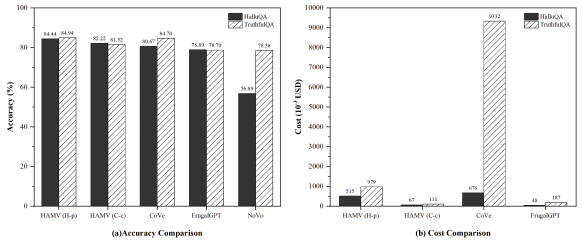

Figure 2: Overall performance of HAMV (H-p and C-c) and baseline methods. NoVo uses local computation without API-based pricing and is therefore excluded from cost comparison.

HAMV (C-c) reaches 82.22% on HalluQA and 81.52% on TruthfulQA. On HalluQA, it exceeds CoVe while using only about 10% of its inference cost; on TruthfulQA, its accuracy is slightly lower than CoVe, but the cost is reduced to roughly 1%. Compared with FrugalGPT, HAMV (C-c) improves accuracy by around 3 points on both benchmarks at similar cost, and shows higher accuracy than NoVo on TruthfulQA, with a larger margin observed on HalluQA.

4.3 Ablation Study

Ablation Setup. To evaluate the individual contributions of HAMV's core components—multi-model collaboration (Multi), DS-based evidence fusion (DS), and MoE dynamic routing (MoE)—we conduct cumulative ablation experiments under the cost-constrained setting using only economic-tier models. Components are sequentially added to assess their impact, with variants including w/o all, +MoE only, +Multi (full/cond.), w/o MoE, w/o DS, and Full HAMV, summarized in Table 1.

Ablation Results. The experimental results lead to the following conclusions:

1) Performance gain of MoE routing under single-model conditions. Even without multi-model collaboration, enabling MoE routing significantly improves performance (HalluQA: 75.11% $\rightarrow$ 79.11%; TruthfulQA: 76.99% $\rightarrow$ 78.70%), showing that task–model matching alone provides notable benefits.

2) Contribution of multi-model collaboration. Adding multi-model collaboration yields the largest performance gain (HalluQA +5.33%; TruthfulQA +2.81%), highlighting that complementary knowledge, reasoning preferences, and error patterns across models effectively enhance robustness and generalization, making it the main source of HAMV's improvement.

3) Effect of DS fusion on verification triggering. Conditional verification lowers the trigger rate (TR = 10.9%), reducing verification cost on HalluQA and TruthfulQA from 180/343 to 60/115, but also causes a precision–recall imbalance, with accuracy dropping to 79.33% and 78.33% and low correction success (30.6% / 29.5%). Introducing DS fusion markedly improves confidence estimation and triggering, raising correction success to 69.2% (HalluQA) and 62.8% (TruthfulQA), and increasing overall accuracy by about +1.12% at lower cost.

4) Performance gain of MoE routing within multi-model framework. Adding MoE routing to multi-model collaboration further improves performance (HalluQA +1.78%; TruthfulQA +1.23%), indicating that dynamic role assignment effectively matches model capabilities to task needs. Though smaller than the gain from multi-model collaboration, it shows a positive trend toward stabilizing system performance near the collaborative upper bound.

5) Synergistic gain of DS fusion and MoE routing in full HAMV. With both DS fusion and MoE routing, HAMV achieves the highest accuracy (HalluQA 82.22%; TruthfulQA 81.52%). On TruthfulQA, their combined improvement (+3.19%) exceeds the sum of individual gains (2.21%), demonstrating a synergistic effect that enhances reliability and overall effectiveness in complex verification tasks.

4.4 Positional Sensitivity Analysis

In multi-model verification, a model's stage assignment—generation (G), verification (V), or synthesis (S)—crucially impacts performance. To quantify this, we define **position sensitivity** as the maximum performance spread over the set of valid role permutations π for a model set M and task T:

$$\Delta P_{M,T} = \max_{\pi \in \Pi} \mathcal{A}(\pi) - \min_{\pi \in \Pi} \mathcal{A}(\pi) \qquad (8)$$

where $\mathcal{A}(\pi)$ denotes the accuracy under assignment π with evaluation modules fixed.

4.4.1 Experimental Setup

On HalluQA and TruthfulQA, we select representative Top and Random model teams. With the model set, evaluation modules, and prompts held fixed, we exhaustively evaluate all six G/V/S permutations and compare their accuracies, isolating performance differences attributable solely to role assignments.

Variant	HalluQA			TruthfulQA		
	Acc (%)	CR/TR	Cost	Acc (%)	CR/TR	Cost
w/o all	75.11	7.1%/100%	171	76.99	8.5%/100%	317
+ MoE only	79.11	11.1%/100%	213	78.70	10.2%/100%	398
+ Multi (full verify)	80.44	12.4%/100%	180	79.80	11.4%/100%	343
+ Multi (conditional)	79.33	30.6%/10.9%	60	78.33	29.5%/13.7%	115
w/o MoE	80.22	69.2%/5.8%	48	79.31	62.8%/5.3%	85
w/o DS	81.11	43.9%/9.1%	75	79.56	36.4%/6.7%	129
Full HAMV	**82.22**	66.7%/7.3%	67	**81.52**	58.0%/6.1%	111

Table 1: The configurations are as follows, progressively adding or removing core components (*Multi*, *DS*, *MoE*): **w/o all**: Single-model baseline without collaboration, DS fusion, or routing; **+MoE only**: Single-model with MoE selecting the best model per task; **+Multi (full verify)**: Static three-model collaboration, full verification for all samples; **+Multi (conditional)**: Same team with coarse confidence-based triggering; **w/o MoE**: Multi-model collaboration with DS fusion but no role-level routing; **w/o DS**: Multi-model collaboration with MoE routing but no DS fusion; **Full HAMV**: Multi-model collaboration with both DS fusion and MoE routing.

4.4.2 Results and Observations

Figure 3 reveals clear performance fluctuations across different G/V/S role permutations.

Significant position sensitivity is observed across all model teams and datasets (ΔP: 5%–30%), indicating role assignment as a critical structural factor in multi-model verification. Although Overall aggregation reduces performance variation, persistent gaps indicate that this effect is inherent to the verification pipeline rather than task-specific.

Position sensitivity is not a simple reflection of global model ranking, as Random teams outperform Top teams under certain role configurations, revealing strong stage-dependent interactions between model capabilities and verification phases.

Statistical significance is confirmed by paired t-tests at the 95% confidence level ($p < 0.01$) with a large effect size (Cohen's $d = 2.15$), suggesting that the observed differences are unlikely to be due to random variation.

Figure 4 assesses the effectiveness of dynamic routing in alleviating position sensitivity. Across all subtasks and the Overall metric, HAMV-MoE consistently outperforms random role permutations by approximately 5–10 percentage points, while maintaining performance within 1–3 percentage points of the experimentally optimal configurations.

The advantage of HAMV-MoE lies in raising the performance lower bound induced by role assignment rather than approximating a single, fixed optimal permutation. Compared with the large performance variance under random permutations, HAMV-MoE maintains more consistent results across subtasks and in overall aggregation, mitigating degradation from unfavorable role configurations.

Position-sensitivity analysis further shows that multi-model verification pipelines are structurally sensitive to role assignment: even with the same model set, reassigning models to generation, verification, and aggregation stages can cause performance fluctuations exceeding 10–15%, reflecting strong stage-dependent model capabilities that limit static, globally ranked configurations.

HAMV-MoE addresses this issue through dynamic routing, adaptively assigning roles at inference time without explicit permutation search, thereby reducing position-induced instability and achieving robust overall performance gains.

4.5 Error Analysis

To identify HAMV's limitations and guide future improvements, we manually analyzed 175 failure cases, classifying errors as evaluation failures (78.8%), verification failures (16.6%), and synthesis failures (4.6%).

Evaluation failures occur when multiple models assign high confidence to wrong answers due to shared corpus biases or knowledge gaps, blocking verification. Examples include "On which day of the Chinese lunar calendar do magpies build the bridge?" and "What is the source of 'The golden monkey raises its mighty staff'?" where models follow common narratives or fixed associations.

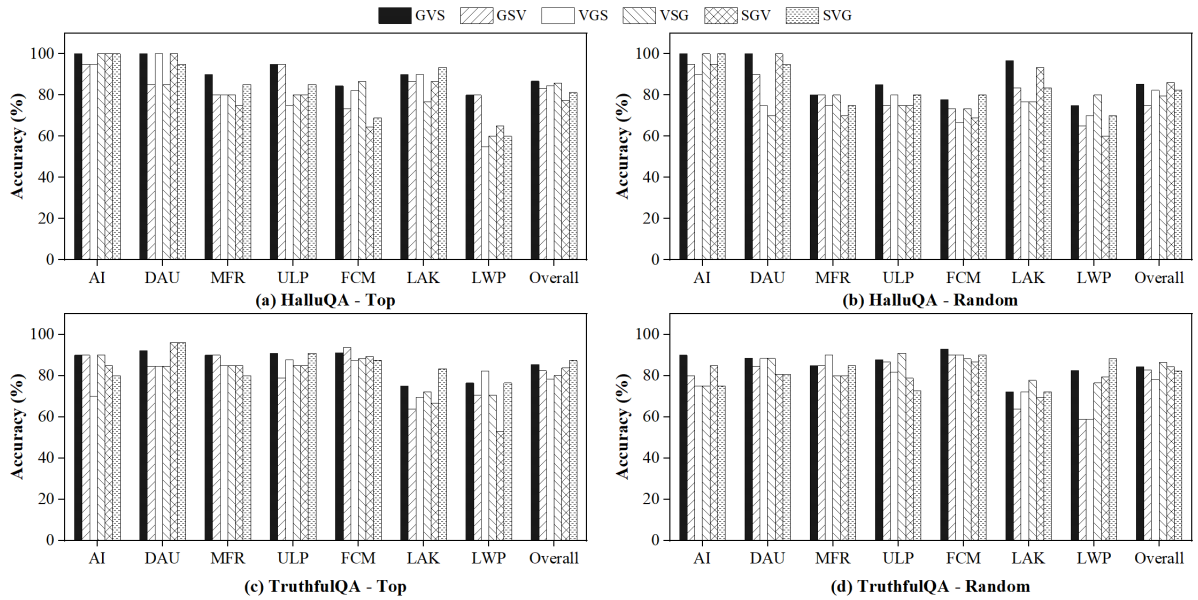

Figure 3: Accuracy of six role permutations across tasks. Legend codes (e.g., GVS) indicate the models assigned to the generation, verification, and summarization stages in order. Overall denotes accuracy over the full dataset.

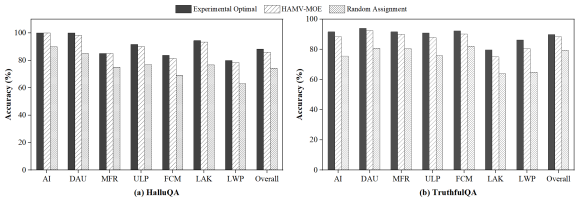

Figure 4: Mitigation of position sensitivity by HAMV-MoE compared to Random Assignment and Experimentally Optimal baselines. The Overall column reports the weighted accuracy across the full dataset.

Heterogeneous evaluation and DS fusion reduce random hallucinations, but consensus errors remain hard to avoid.

Verification failures occur when queries exceed model knowledge coverage, leading models to favor fluency over factuality. For instance, in "What follows the last line of *The Song of the Pipa*?", fabricated continuations are often generated. Heterogeneous verification mitigates but does not eliminate such generative drift.

Synthesis failures reflect a bias toward linguistic harmonization rather than factual adjudication under conflicting evidence. For example, in explaining "A prime minister's belly can hold a boat," final outputs may still deviate from the truth despite correct verification evidence.

Overall, these errors reflect LLMs' structural limits in knowledge, verification, and conflict resolution, not flaws in HAMV. Future work can improve evaluation heterogeneity, verification constraints, and evidential signals in synthesis.

5 Conclusion

This paper presents HAMV, a heterogeneous adaptive multi-model verification framework for fact checking. It models verification as a structured scheduling problem, treating heterogeneous models as experts dynamically assigned to generation, evaluation, verification, and synthesis stages. HAMV combines MoE-inspired dynamic routing with evidence-based confidence fusion to balance verification reliability and computational cost without requiring additional model training.

Experiments on Chinese and English hallucination benchmarks show that HAMV's structured scheduling is effective and stable. Ablations indicate that multi-model collaboration, dynamic routing, and evidence fusion contribute complementary gains. Position sensitivity analysis highlights that role assignment within the pipeline affects performance, as different stage allocations can cause substantial variation even with a fixed model pool, showing that static, globally ranked configurations cannot capture stage-dependent behavior.

Although HAMV is limited by shared knowledge gaps and generative uncertainty, modeling multi-model fact verification as a structured cross-model scheduling problem provides a generalizable approach for high-reliability language model systems. Future work could explore learnable schedulers and broader reasoning scenarios.

Limitations

Although HAMV significantly improves factual reliability and reduces hallucinations under a cost–effectiveness trade-off, several limitations remain. First, the framework depends on externally provided model pools and fixed capability profiles; its performance may vary as LLMs evolve or when deployed in domains with limited model diversity. Second, DS-based confidence fusion and MoE routing, while effective, introduce additional latency compared with single-model workflows. Further optimization is needed for real-time or large-scale applications. Third, our evaluation is restricted to two factual QA datasets; broader testing on multi-turn dialogue, long-context reasoning, and domain-specific tasks is required to fully assess generalization. Finally, HAMV is verification-oriented and does not directly address deeper epistemic issues such as model uncertainty calibration or attribution reliability, which remain open challenges.

References

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Qinyuan Cheng, Tianxiang Sun, Wenwei Zhang, Siyin Wang, Xiangyang Liu, Mozhi Zhang, Junliang He, Mianqiu Huang, Zhangyue Yin, Kai Chen, and 1 others. 2023. Evaluating hallucinations in chinese large language models. *arXiv preprint arXiv:2310.03368*.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Representation Learning*, volume 2024, pages 54158–54183.

Arthur P Dempster. 2008. Upper and lower probabilities induced by a multivalued mapping. In *Classic works of the Dempster-Shafer theory of belief functions*, pages 57–72. Springer.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-verification reduces hallucination in large language models. In *Findings of the association for computational linguistics: ACL 2024*, pages 3563–3578.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2023. In-context autoencoder for context compression in a large language model. *arXiv preprint arXiv:2307.06945*.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Yuzhe Gu, Ziwei Ji, Wenwei Zhang, Chengqi Lyu, Dahua Lin, and Kai Chen. 2024. Anah-v2: Scaling analytical hallucination annotation of large language models. *Advances in Neural Information Processing Systems*, 37:60012–60039.

Zheng Yi Ho, Siyuan Liang, Sen Zhang, Yibing Zhan, and Dacheng Tao. 2025. Novo: Norm voting off hallucinations with attention heads in large language models. In *The Thirteenth International Conference on Learning Representations*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and 1 others. 2025a. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. 2025b. RouterEval: A comprehensive benchmark for routing LLMs to explore model-level scaling up in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3860–3887, Suzhou, China. Association for Computational Linguistics.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Zifeng Wang, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. 2025. Universal LLM routing with correctness-based representation. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 9004–9017.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024a. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, and 1 others. 2024b. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. 2024. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 14672–14685.

Jiaxin Zhang, Zhuohang Li, Kamalika Das, Bradley Malin, and Sricharan Kumar. 2023. Sac3: reliable hallucination detection in black-box language models via semantic-aware cross-check consistency. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15445–15458.

A Prompts for Experimental Tasks

The prompts provided in the appendix are intended solely to illustrate the interface constraints and input-output semantics of each stage in HAMV, rather than to optimize performance through specific wording or manual heuristics. In practice, any implementation that satisfies the same structural constraints—such as function calls, templated interfaces, or parameterized controls—can replace the corresponding prompts without affecting the overall logic or conclusions of the framework. Therefore, the core contribution of this work does not depend on prompt design itself, but rather on the structural modeling of the multi-model verification pipeline and the dynamic role-assignment mechanism.

A.1 MoE Routing Prompt

As illustrated in Figure 5, this routing prompt is designed to explicitly model role-assignment decisions within the multi-model verification pipeline, given an input task and a set of models. The prompt maps task features and model profiles to a set of structured scheduling variables, including the assignment of models to the generation, verification, and synthesis stages, stage-level evaluation weights, and the adaptive confidence threshold, thereby forming a complete verification execution plan.

Serving as the unified scheduling interface within the HAMV framework, this prompt captures the positional sensitivity inherent in the multi-model verification pipeline in a structured manner, rather than selecting models based on fixed rules or heuristic rankings.

A.2 Baseline Answer Generation Prompt

As illustrated in Figure 6, this prompt is used to standardize the output format of the generation stage, ensuring that baseline answers are both determinable and minimally ambiguous. By uniformly modeling response style and expression constraints, the prompt aims to reduce vague, speculative, or unverifiable statements, thereby providing stable, low-noise candidate inputs for subsequent evaluation and verification stages.

This prompt does not perform factual judgment itself; its role is solely to ensure that generated outputs can be consistently and reliably processed by downstream modules.

[Task Goal] You are a fact-checker. Provide a concise, fact-based answer to the following question. Avoid speculation, assumptions, and unverified information.
[Inputs] Question: {question}
[Output Requirements]
1. The answer must be based on verifiable facts.
2. The answer must not exceed 100 words.
[Example]
Question: What year did Einstein win the Nobel Prize?
Answer: Einstein won the Nobel Prize in Physics in 1921 for his research on the photoelectric effect.
[Output] {baseline_answer}

Figure 6: Baseline Answer Generation Prompt

A.3 Confidence Assessment Prompt

As shown in Figure 7, this prompt constrains the evaluation model to output a confidence score within the continuous range [0.0, 1.0] and perform a comprehensive assessment along three dimensions: factual consistency, logical coherence, and clarity of expression.

By explicitly standardizing both the output format and evaluation dimensions, the prompt ensures that the confidence signals can be directly parsed and utilized by the subsequent Dempster–Shafer fusion module, thereby avoiding interference from unstructured interpretations in the evidence aggregation process.

[Task Goal] Evaluate the confidence of the following answer (range: 0.0–1.0) based on factual accuracy, logical coherence, and clarity. The evaluation must focus on whether the answer is grounded in scientific facts rather than superstition or pseudoscience. Provide only a single numerical score. A lower score indicates lower confidence.
[Inputs]
Question:{question}
Answer:{baseline_answer}
[Output Requirements] Output a single confidence score between 0.0 and 1.0, with no explanation or extra text.
[Example] 0.92
[Output] {confidence_score}

Figure 7: Confidence Evaluation Prompt

A.4 Verification Question Generation Prompt

As shown in Figure 8, when the fused confidence falls below the adaptive threshold, this prompt is used to generate a structured set of verification questions. The generated questions cover various potential risk points, including factual verification, reasoning chain consistency checks, and counterfactual analysis, aiming to expose possible errors or inconsistencies in the baseline answer from multiple perspectives.

The purpose of this prompt is to systematically expand the verification space, rather than to target specific error patterns.

[Task Goal] Generate verification questions according to the following framework (must include at least one [Fact Check] question):
1. [Fact Check]: Request specific sources such as paper titles or authoritative institutions.
2. [Logic Detection]: Identify hidden assumptions in the reasoning chain.
3. [Counterfactual Test]: Assume a premise is false and ask how the conclusion would change.
4. [Numerical Validation]: Check calculations or statistical ranges.
[Inputs] Question: {question} Baseline Answer: {baseline_answer}
[Output Requirements]

[System Role] Intelligent Scheduling Core (MoE Router)
[Task Goal] Generate a structured and executable model-assignment plan for the HAMV multi-model verification pipeline.
[Inputs]
1.Query_Type (Problem Type) $T \in \{\text{ULP, FCM, LAK, AI, DAU, MFR, LWP}\}$
2.Optimization_Strategy $S \in \{\text{“High-Performance”, “Low-Cost”}\}$
3.Model_Profile P containing capability, cost tier, and ErrDiv attributes.
[Scheduling Logic]
1. Parse T and S.
2. Determine candidate pool P':
   a. If $S =$ Low-Cost $\rightarrow$ keep only models with Economic cost tier.
   b. If $S =$ High-Performance $\rightarrow$ use full model pool P.
3. Assign roles:
   a. Baseline_Generator (m_{Gen}): highest Baseline Generation Ability in P' for type T.
   b. Verification_Q&A (m_{Ver}): strongest Verification Ability in $P' \setminus \{m_{\text{Gen}}\}$.
   c. Confidence_Assessors (m_{Eval1}, m_{Eval2}): top two Assessment Ability models in $P' \setminus \{m_{\text{Gen}}\}$.
   d. Answer_Integrator (m_{Gen}): model maximizing ErrDiv over the team $\{m_{\text{Gen}}, m_{\text{Eval}}, m_{\text{Ver}}, m_{\text{Sum}}\}$.
4. Configure Assessment Group:
   a. assessors = $\{m_{\text{Eval1}}, m_{\text{Eval2}}\}$.
   b. reliability weights = $\{\alpha_1, \alpha_2\}$ from historical accuracies on type T.
   c. adaptive threshold $\tau_c = (\alpha_1 + \alpha_2)/2$.
5. Output all assignments and parameters as a structured JSON object.
[Output (JSON Example)]
{"Query_Type":"FCM","Optimization_Strategy":"Low-Cost","Role_Assignment":{"Baseline_Generator":"Qwen-Plus","Confidence_Assessors":["GLM-4-air", "ERNIE-3.5"], "Verification_QNA": "ERNIE-3.5", "Answer_Integrator": "GLM-4-air" }, "Reliability_Weights": { "GLM-4-air": 0.76, "ERNIE-3.5": 0.82 }, "Threshold": { "τ_c": 0.79 } }

Figure 5: Prompt for MoE-based Intelligent Scheduling in HAMV

1. Generate at least three verification questions.
2. Each question must directly target a potential risk point in the Baseline Answer.
3. Output format must be a JSON list.
[Example]
[{ "type":"Fact Check", "question":"Please provide the Nobel Prize official website link for Einstein’s 1921 award" }]
[Output] {ver_questions}

Figure 8: Verification Question Generation Prompt

A.5 Verification Answering Prompt

As shown in Figure 9, this prompt is used to standardize the response format during the verification stage, requiring the model to answer each verification question individually and output the results in a structured JSON format. This design aims to avoid introducing additional generative noise or hallucinations, ensuring that the verification results can serve as clear and parsable evidence for the subsequent conflict analysis module. The verification answers generated in this manner constitute the core information source for evidence fusion and conflict resolution within HAMV.

[Task Goal] Please answer the following verification questions one by one. Ensure each answer is based on verifiable facts and logical reasoning, avoiding assumptions or speculative statements.

[Inputs]
Question: {question}
Baseline Answer: {baseline_answer}
Verification Questions: {ver_questions}
[Output Requirements]
1. Each answer must be concise and clear.
2. If a question has multiple reasonable interpretations, briefly describe the reasoning process.
3. Output format must be a JSON list.
[Example]
[{ "question":"If the photoelectric effect had not been experimentally confirmed, would the committee still have possibly given the award?", "answer":"Unlikely. The Nobel Prize requires theories to be experimentally verified." }]
[Output] {ver_qna}

Figure 9: Verification Question Answering Prompt

A.6 Summarize Answer Prompt

As shown in Figure 10, this prompt is used in the synthesis stage to uniformly process the baseline answer along with verification evidence. By identifying conflict points, assigning conflict weights, and performing evidence-based consistency corrections, this stage produces the final high-confidence output.

This prompt does not generate new facts; rather, it serves as an evidence-driven synthesis interface, ensuring that the final answer remains consistent and interpretable under multi-source evidence con-

straints.

[Task Goal] Combine the Verification Questions and Answers with the Baseline Answer, and summarize according to the specified structure to produce a final decision.
[Inputs]
Question: {question}
Baseline Answer: {baseline_answer}
Verification Questions and Answers: {ver_qna}
[Output Requirements] Summarize according to the following structure:
1. Conflict List: List all verification results that contradict the Baseline Answer (including external verification mismatches).
2. Credibility Weight: Assign weights based on conflict type (Factual Error $>$ Logical Flaw $>$ Vague Wording).
3. Final Decision: If any high-weight conflict exists $\rightarrow$ the answer must be revised.
[Example]
"The baseline answer states that the Tokyo Olympics took place in 2020, but external verification from authoritative sources indicates it occurred in 2021 (High-Weight Conflict). Therefore, the final answer is revised to 2021."
[Output] {final_answer}

Figure 10: Summarize Answer Prompt