

Knowledge-Grounded Agentic Large Language Models for Multi-Hazard Understanding from Reconnaissance Reports

Chenchen Kuai^{a,*}, Zihao Li^{a,*}, Braden Rosen^b, Stephanie Paal^a, Navid Jafari^a, Jean-Louis Briaud^a, Yunlong Zhang^a, Youssef M. A. Hashash^{a,c}, Yang Zhou^{a,**}

^a*Zachry Department of Civil & Environmental Engineering, Texas A&M University, 3136 TAMU, College Station, 77843, TX, USA*

^b*Department of Engineering Technology and Industrial Distribution, Texas A&M University, College Station, 77843, TX, USA*

^c*Department of Civil and Environmental Engineering, University of Illinois Urbana-Champaign, Urbana, 61801, IL, USA*

Abstract

Post-disaster reconnaissance reports contain critical evidence for understanding multi-hazard interactions, yet their unstructured narratives make systematic knowledge transfer difficult. Large language models (LLMs) offer new potential for analyzing these reports, but often generate unreliable or hallucinated outputs when domain grounding is absent. This study introduces the Mixture-of-Retrieval Agentic RAG (MoRA-RAG), a knowledge-grounded LLM framework that transforms reconnaissance reports into a structured foundation for multi-hazard reasoning. The framework integrates a Mixture-of-Retrieval mechanism that dynamically routes queries across hazard-specific databases while using agentic chunking to preserve contextual coherence during retrieval. It also includes a verification loop that assesses evidence sufficiency, refines queries, and initiates targeted searches when information remains incomplete. We construct HazardRecQA by deriving question-answer pairs from GEER reconnaissance reports, which document 90 global events across seven major hazard types. MoRA-RAG achieves up to 94.5% accuracy, outperforming zero-shot LLMs by 30% and state-of-the-art RAG systems by 10%, while markedly reducing hallucinations across diverse LLM architectures. MoRA-RAG also enables open-weight LLMs to achieve performance comparable to proprietary models. It establishes a new paradigm for transforming post-disaster documentation into actionable, trustworthy intelligence for hazard resilience.

Keywords: Large Language Models (LLMs), Agentic AI, Retrieval-Augmented Generation (RAG), Multi-Hazard Understanding, Reconnaissance Reports

1. Introduction

Natural hazards such as earthquakes, hurricanes, floods, and wildfires continue to threaten communities, infrastructure, and economies worldwide [1, 2, 3]. While each hazard poses distinct

*These authors contributed equally to this work.

**Corresponding author: Yang Zhou .

Email address: yangzhou295@tamu.edu (Yang Zhou)

risks, many disasters occur in sequence or in combination, amplifying impacts significantly [4]. Earthquakes may destabilize slopes and cause landslides, hurricanes often generate both storm surge and inland flooding, and heavy rainfall can accelerate soil erosion. These complex dynamics make it essential to understand not only individual hazards but also their interactions. A crucial element in building such understanding is the knowledge transfer [5], the ability to learn lessons and knowledge from past events and apply them to strengthen preparedness, response, and resilience against future hazards.

Knowledge in the hazard domain is preserved in many forms. Operational manuals describe best practices, predictive models simulate hazard scenarios, and sensor networks provide continuous environmental monitoring [6, 7, 8]. Yet none of these sources fully capture the depth and contextual nuance of post-disaster field evidence. Reconnaissance reports occupy a unique position in this landscape [9]. Compiled by interdisciplinary teams after major disasters, they document hazard characteristics, infrastructure performance, environmental conditions, and social impacts. These reports integrate measurements, photographs, maps, and expert analyses, often capturing cascading or compound hazards that unfold within a single disaster event. They represent one of the most authoritative sources of hazard knowledge [10].

However, the very richness that gives these reports their scientific and practical value also makes them challenging to exploit systematically. Reconnaissance documents often span hundreds of pages and encompass a diverse blend of qualitative and quantitative content, including narrative accounts, measurements, maps, tables, and photographic evidence. Although this depth of information offers unparalleled insight into disaster impacts, it is typically presented in highly specialized language filled with technical jargon, abbreviations, and situational references that demand expert interpretation [11]. Critical findings, such as the relationships between hazard intensity, structural response, and cascading failures, are frequently fragmented between sections and embedded within unstructured narrative text. The absence of uniform standards among the reporting teams further amplifies this challenge, making it difficult to systematically synthesize, compare, and transfer the knowledge contained in these reports [12].

Recent advances in large language models (LLMs) offer new opportunities to extract, integrate, and reason with this wealth of knowledge. LLMs have demonstrated strong performance on tasks such as summarization, answering questions, and contextual reasoning [13]. Yet, because they rely heavily on broad world knowledge, their output often suffers from hallucinations, producing fluent but unsupported claims. Retrieval-augmented generation (RAG) seeks to address this issue by grounding responses in external documents [14]. While effective for general-purpose texts, existing RAG frameworks are not well-suited for long and highly technical hazard reports. Once these reports are divided into smaller chunks (i.e., short text segments split for embedding and retrieval) for vector databases, their rich contextual dependencies become fragmented across numerous segments. Retrieving truly relevant evidence from such a vast and heterogeneous hazard corpus remains difficult [15]. In addition, most retrieval pipelines rely solely on semantic similarity, tending to retrieve textual information within a single topical dimension [16, 17]. In hazard domains, this corresponds to retrieving information from one hazard type while neglecting multiple or cascading hazards. Consequently, the retrieved information can be incomplete or misleading, and the LLMs built upon it often fail to produce trustworthy, well-grounded responses. Moreover, conventional RAG systems lack mechanisms to assess whether the retrieved evidence

is sufficient or to recognize when a reliable answer cannot be provided [18].

These limitations reveal a fundamental gap between current retrieval architectures and the demands of multi-hazard reasoning. Addressing this gap requires a framework capable of both precision retrieval and iterative validation across diverse hazard domains. To this end, this paper presents the first systematic effort to adapt RAG for transforming hazard reports, such as reconnaissance documents, into structured and reliable world knowledge that supports multi-hazard understanding and reasoning with LLMs. A dedicated dataset, *HazardRecQA*, is developed from reconnaissance reports to benchmark the quality of knowledge transfer. Building upon this foundation, we propose the *Mixture-of-Retrieval Agentic RAG* (MoRA-RAG) framework, which overcomes the limitations of existing approaches. MoRA-RAG incorporates a *Mixture-of-Retrieval* (MoR) mechanism that dynamically selects relevant hazard databases through a routing strategy, enhancing retrieval precision and highlighting cross-hazard relationships. Its agentic architecture actively verifies the sufficiency and consistency of retrieved evidence, performs targeted online searches to fill knowledge gaps, and synthesizes external information to strengthen reasoning robustness. By combining these capabilities, MoRA-RAG establishes a new paradigm for trustworthy and context-aware knowledge transfer in the hazard domain, enabling LLMs to reason across interconnected disaster processes and support more informed decision-making.

2. Related Literature

2.1. Knowledge transfer in natural hazards

Knowledge transfer from past events helps understand how hazards evolve, interact, and impact communities as new events unfold [3, 19]. By drawing on prior experiences, researchers and practitioners are able to identify patterns of vulnerability, recognize emerging risks, and anticipate cascading effects across interconnected systems [20, 21, 22]. Forecasting and predictive modeling leverage data from historical events to identify recurring patterns and prepare in advance of upcoming hazards, forming the foundation of modern early warning systems [23, 24]. Similarly, resilience analyses based on past disasters reveal how infrastructure, logistics, and social systems respond and recover under disruption, offering insights into system fragility and redundancy [25]. All together, knowledge transfer remains a cornerstone of hazard impact analysis and an indispensable component of disaster preparedness and response.

However, much of this knowledge transfer still relies primarily on structured data (e.g., tabular records and quantitative indicators). At the same time, a substantial portion of valuable information contained in heterogeneous and unstructured sources remains underutilized [26, 27, 28]. Reconnaissance reports, field observations, and expert narratives often provide rich contextual details about on-the-ground conditions, response actions, and cascading impacts that are difficult to capture through structured datasets alone [29]. Yet, processing such heterogeneous data is inherently challenging. Only a limited number of studies have attempted to bridge these unstructured sources using advanced natural language processing (NLP) and related analytical techniques. For instance, Ma et al. [30] adopted text mining and extracted key information from dense geological hazard documents, while He et al. [31] integrated past textual information into models of social response during disasters through event-embedding approaches. While effective, existing efforts

remain largely task-specific and domain-bounded. A recent review paper calls for scalable disaster information management and highlights the importance of transforming knowledge from these rich yet heterogeneous sources; however, still no practical framework has been developed [32].

2.2. Evolution of LLMs and context engineering

The rapid development of LLMs has drastically reshaped NLP capabilities and opened new opportunities to bridge existing gaps in hazard knowledge transfer from unstructured reports. Transformer-based foundation models such as ChatGPT, Claude, Gemini, and LLaMA exhibit strong general abilities in comprehension, summarization, and question answering [33, 34, 35]. Models such as DeepSeek have further enhanced reasoning performance through reinforcement learning-based optimization [36, 37]. However, these foundation models are primarily trained on broad, general-purpose text and remain insufficiently grounded in domain knowledge (e.g. hazard knowledge) [38]. As a result, they often generate hallucinated or incomplete responses when applied to domain-specific tasks. To mitigate this issue, context engineering has emerged as a promising solution. Context engineering refers to designing an information-rich environment, such as incorporating domain-specific documents, structured metadata, or tailored prompts, so that LLMs are better grounded when generating answers [39, 40].

In practice, the context engineering is often realized through RAGs. RAG augments the context of LLMs with retrieved text passages, leading to large improvements in knowledge-intensive tasks [41]. Recent studies have also applied RAG for hazard and climate resilience, such as ChatClimate [42], which enables question answering and reasoning about climate-related information, and other systems that assist analysis and decision-making during natural hazards and extreme weather events [43]. While promising, RAG faces critical challenges: when irrelevant or low-quality evidence is retrieved, model performance can degrade rather than improve [44, 45]. This “garbage-in, garbage-out” issue underscores persistent limitations in retrieval accuracy and evidence reliability. Recent models such as C-RAG [46] and MAIN-RAG [47] introduce correction and validation mechanisms to refine retrieved knowledge, but they mainly focus on re-actively correcting retrieval errors. Mechanisms for sufficiency and reliability of the information to answer the query remain unexplored.

3. QA Dataset Construction

HazardRecQA is developed to evaluate an LLMs’ ability to transfer knowledge from post-disaster reconnaissance documents through question–answer (QA) pairs. These QA pairs provide the foundation for assessing model performance (e.g., LLM and RAG) [48], where the answering accuracy indicates how effectively a model captures and applies hazard knowledge from the reports. The ground-truth hazard knowledge is obtained from post-disaster reconnaissance reports (i.e., GEER Reports [9]) and organized into QA form for systematic evaluation.

GEER reconnaissance reports provide comprehensive post-disaster documentation, including site observations, photographs, instrumentation data, geospatial records, and analyses of failure mechanisms. These reports span seven categories of natural hazards¹ observed worldwide from

¹Hurricanes and typhoons are considered the same hazard type, as they are regional names for the same phenomenon. In this study, both are collectively referred to as hurricanes.

1989 to 2025 (Figure 1). As the world’s largest and most comprehensive open-access collection of post-disaster field reconnaissance, the GEER database encompasses 58 earthquakes, 9 floods, 2 tsunamis, 1 wildfire, 4 storms, 5 landslides, and 11 hurricanes, totaling 90 hazard events. Building on the knowledge preserved in these reports, the questions in HazardRecQA are designed to evaluate models across four categories of hazard-related knowledge: (1) *Hazard Characteristics*: describing the event itself, including what happened, when and where it occurred, its magnitude, setting, or physical triggers; (2) *Analysis Approach*: focusing on how the event was examined through data collection, instruments, surveys, models, or analytical methods; (3) *Impacts and Damage*: addressing the consequences or losses caused by the event, such as structural failures, economic damage, or human impacts; and (4) *Response and Recovery*: covering the actions taken after the event, including emergency response, evacuation, repair, or long-term recovery.



Figure 1: Coverage and event distribution of GEER reconnaissance reports [9].

The QA generation process employs GPT-5 nano to automatically construct factual QA pairs from reconnaissance report paragraphs, following recent advances showing that LLMs can comprehend complex domain text and autonomously generate coherent, fact-based questions [49, 50]. Each paragraph from the reconnaissance reports undergoes a data cleansing step to remove non-informative components (e.g., tables of contents, acknowledgments, references) and retain semantically rich content for question construction. GPT-5 is then guided by a structured instruction with strict output constraints to generate verifiable QA pairs in either True/False or Multiple-Choice (A–D) formats. These formats are adopted because they provide clear factual grounding and allow objective evaluation through accuracy-based metrics [51]. Each QA item is further classified into the four question categories introduced above. The generated outputs undergo post-validation through schema checking to ensure consistency with the predefined QA formats.

The HazardRecQA dataset contains a total of 5,776 QA pairs generated from post-disaster reconnaissance reports. As shown in Table 1 and Figure 2, each QA instance is categorized along two dimensions: (1) task category, including Hazard Characteristics, Response and Recovery, Analysis Approach, and Impacts and Damage, and (2) hazard type, such as floods, landslides, storms,

Task Category	TF	MC	ALL
Hazard Characteristics	1652	692	2344
Response and Recovery	123	83	206
Analysis Approach	666	844	1510
Impacts and Damage	1175	541	1716
Total	3616	2160	5776

Table 1: Question type distribution across task categories, TF (True/False) and MC(Multiple-Choice).

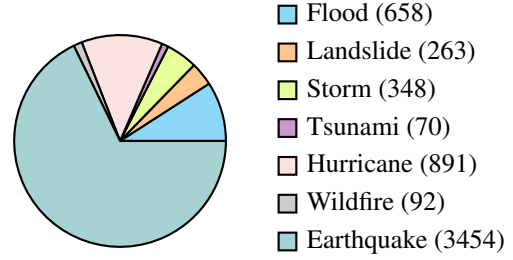


Figure 2: Proportional distribution of total valid QA across hazard types (total QA).

tsunamis, hurricanes, wildfires, and earthquakes. The dataset comprises both True/False and Multiple-Choice formats, enabling diverse factual tasks and straightforward evaluation through answer accuracy. The complete QA construction prompts are presented in [Appendix A](#).

4. Methodology

This section will first introduce the framework of RAG, explaining how external documents are processed into vector databases and how the retrieval and generation modules function to produce grounded answers. We then extend this framework through the proposed Mixture-of-Retriever Agentic RAG (MoRA-RAG), which enhances both database construction and retrieval precision. Furthermore, an agentic LLM framework is incorporated to validate, reflect on, and refine the generated responses for improved reliability.

4.1. Retriever-Argumented Generation

The RAG framework operates in two main stages: (1) constructing a knowledge database from external documents, and (2) performing retrieval and answer generation based on a given query. As illustrated in Figure 3, consider an example question asking about the engineering consequence of erosion on the upstream side of an abutment during a hurricane. When relying solely on its general world knowledge, the LLM produces a generic response that lacks domain-specific evidence. In contrast, RAG retrieves relevant information from multiple reconnaissance reports and identifies a specific segment in the GEER-032 Hurricane Sandy Report, which documents abutment damage and T-wall settlement. By grounding the generation in such retrieved evidence, RAG enables the model to effectively transfer knowledge from the documents.

4.1.1. Database chunking

In RAG, the first step is to construct a vector database suitable for retrieval. The original documents (e.g., reconnaissance reports) are often lengthy and cannot be directly fed into a model’s context window. Therefore, the documents are required to be segmented into smaller text units; this step is known as chunking. These chunks are later converted into vector representations and are jointly preserved in the vector database.

We denote the chunking process the documents \mathcal{D} as:

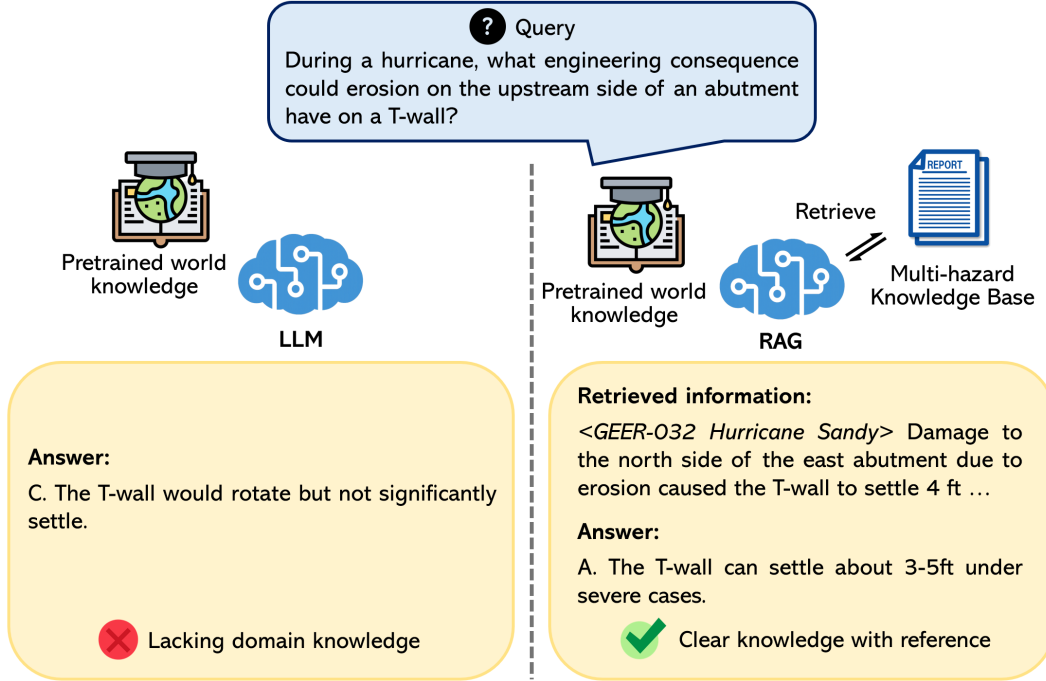


Figure 3: RAG systems explained, it is able to extract and transfer hazard knowledge from external databases.

$$\{d_1, d_2, \dots, d_n\} = \varphi(\mathcal{D}) \quad (1)$$

where $\{d_1, d_2, \dots, d_n\}$ represents the set of chunks produced by the chunker φ from document \mathcal{D} . Common strategies include fixed-token and paragraph-based chunking [52, 53]. While widely used, these methods can break the logical structure of the text, causing key contextual relationships, such as cause-and-effect connections or hazard impact descriptions, to be split across segments [54]. This fragmentation often weakens retrieval relevance and reduces the completeness of the retrieved information.

Each chunked text is then converted into a numerical vector representation using an embedding function $f(\cdot)$:

$$e = f(d) \quad (2a)$$

$$f(\cdot) : \mathcal{X} \rightarrow \mathbb{R}^m \quad (2b)$$

where \mathcal{X} denotes the text space and \mathbb{R}^m represents the m -dimensional embedding space determined by the language embedding model. `text-embedding-3-small` [55], an embedding model from OpenAI adopted in this study, produces 1,536-dimensional embeddings.

Finally, each chunk and its corresponding embedding are stored in the vector database $\hat{\mathcal{D}}$:

$$\hat{\mathcal{D}} = \{(d_1, e_1), (d_2, e_2), \dots, (d_n, e_n)\} \quad (3)$$

This database serves as the external knowledge source that supports later retrieval and generation within the RAG framework.

4.1.2. Retriever & Generation

The retrieval component in RAG is responsible for locating the most relevant information from the external vector database $\hat{\mathcal{D}}$, while the generation component integrates this retrieved knowledge into the LLM’s context to produce grounded and reliable answers. Formally, the RAG framework, denoted as \mathcal{M} , can be expressed as:

$$\mathcal{M} = (\mathcal{G}, \mathcal{R}), \quad \mathcal{M}(q \mid \hat{\mathcal{D}}) = \mathcal{G}(q, \mathcal{R}(\hat{\mathcal{D}} \mid q)) \quad (4)$$

where \mathcal{G} is the generation module (i.e., the backbone LLM that generates the response) and \mathcal{R} is the retrieval module. Given a user query q and an external knowledge base $\hat{\mathcal{D}}$, the retrieval module identifies and returns the most relevant evidence $\mathcal{R}(\hat{\mathcal{D}} \mid q)$, which is then used by \mathcal{G} to generate the final answer.

The retriever first identifies the most relevant evidence from the database with respect to the query q through a two-stage process: a coarse-grained retrieval followed by a fine-grained reranking. These consist of a fast, ‘coarse-grained’ search to find potential matches, followed by a slower, ‘fine-grained’ reranking to find the best matches. The coarse-grained stage is performed by a bi-encoder, which encodes the query and each document independently and measures their similarity efficiently, though at lower precision. The fine-grained stage then uses a cross-encoder to refine the ranking of retrieved candidates with higher accuracy but greater computational cost. The bi-encoder and cross-encoder will be introduced in detail. The retriever therefore, balances the performance and computational cost. It first selects the top- L candidate chunks using the bi-encoder, where L controls the number of chunks passed to the cross-encoder for reranking. The value $L = 50$ is adopted following prior work [52]. The cross-encoder then reranks these L candidates and selects the top- K most relevant chunks, with $K = 5$ adopted [56].

Formally, the bi-encoder computes the cosine similarity between the embeddings of the query and each chunk, and selects top- L chunks with highest similarity score:

$$\{d_1, d_2, \dots, d_L\} = \arg \max_{d \in \hat{\mathcal{D}}}^L \mathcal{L}_b(d, q), \quad (5a)$$

$$\mathcal{L}_b(d, q) = \frac{e \cdot f(q)}{\|e\| \|f(q)\|}, \quad (5b)$$

where $f(q)$ and $e \in \hat{\mathcal{D}}$ are the embedding vectors of the query and chunk, respectively. The cosine similarity function $\mathcal{L}_b(\cdot)$ measures the normalized inner product between the two embeddings, with higher values indicating stronger semantic relevance.

In the fine-grained, reranking stage, the cross-encoder takes both the query and each candidate chunk as a single combined input, allowing direct interaction between the two. This model jointly encodes the pair into a contextual representation $g(q, d)$, and the relevance score is computed as:

$$\{d_1, d_2, \dots, d_K\} = \arg \max_{d \in \{d_1, d_2, \dots, d_L\}}^K \mathcal{L}_c(d, q), \quad (6a)$$

$$\mathcal{L}_c(d, q) = \sigma(\mathbf{w}^\top g(q, d)), \quad (6b)$$

where $\sigma(\cdot)$ denotes the sigmoid activation and \mathbf{w} is a learnable scoring vector. The resulting top- k chunks represent the most contextually relevant evidence retrieved for the query.

The final retrieval output concatenates these selected chunks to form the context used for generation:

$$\mathcal{R}(\hat{\mathcal{D}} | q) = d_1 \oplus d_2 \oplus \dots \oplus d_k \quad (7)$$

This retrieved evidence $\mathcal{R}(\hat{\mathcal{D}} | q)$ is then added to the original query, enabling the generation module \mathcal{G} to produce the evidence-grounded answer.

4.2. Mixture-of-Retriever Agentic RAG

We introduce the Mixture-of-Retriever Agentic RAG (MoRA-RAG) framework (Figure 4), which extends the standard RAG by integrating two key enhancements. First, it incorporates an improved chunking strategy and a mixture-of-retriever mechanism to increase retrieval precision. Second, it introduces an agentic LLM framework designed to address the validation gap in traditional RAG systems. This agentic framework enables the model to validate, reflect upon, and refine its retrieved evidence through coordinated specialized agents. The detailed design and function of each component are presented in the following sections.

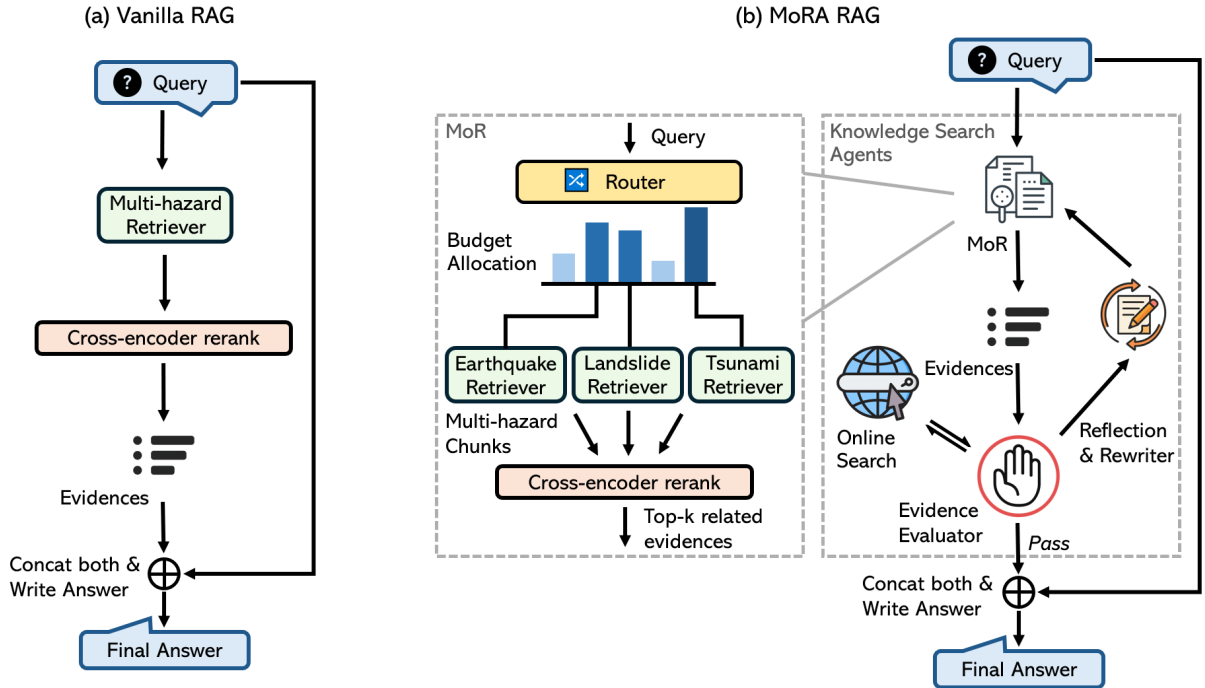


Figure 4: Architecture of the proposed MoRA-RAG framework compared with Vanilla RAG.

4.2.1. Agentic Chunk

In MoRA-RAG, the chunking process is improved by incorporating LLM agents as the document chunker φ , allowing the segmentation of reconnaissance reports into knowledge-preserving

units. Instead of relying on fixed-token or paragraph-based segmentation, which may fragment the logical flow of hazard observations, the agentic chunker ensures that each chunk maintains contextual completeness and factual clarity.

The process begins with the extraction of propositions, where the LLM identifies concise factual statements from long paragraphs. Each proposition is rewritten into a standalone and explicit sentence that includes both subject and predicate, for example, transforming “Observed liquefaction along quay walls” into “Investigators observed liquefaction along quay walls.” Next, the LLM groups related propositions into compact and thematically consistent chunks. Each chunk typically combines statements describing hazard conditions, observed impacts, and contextual factors, followed by a short LLM-generated summary capturing the key idea. To maintain focus and internal consistency, each chunk contains fewer than ten propositions, avoiding mixed or overly long segments. This structure reflects how domain experts summarize multiple related observations in post-disaster reports.

The resulting chunks are further paired with their embedding and produce the vector database, denoted as $\hat{\mathcal{D}}$. This serves as the enhanced knowledge database for retrieval and generation in MoRA-RAG. Implementation details and comparisons with basic chunking methods are detailed in [Appendix B](#).

4.2.2. Mixture of Retriever

The MoR module enhances the retrieval process by enabling hazard-aware routing and adaptive evidence selection from the database. Instead of retrieving from a single unified knowledge base, the query is first analyzed by a router agent that estimates the relevance distribution over multi-hazard domains. This design allows the framework to allocate retrieval budgets, focusing on the hazards most likely related to the query, while maintaining coverage over potentially interacting hazard types.

Formally, let \mathcal{H} denote the complete set of hazard categories (e.g., earthquake, flood, storm, landslide, wildfire, tsunami). The router agent estimates a probability vector $\mathbf{p} = \mathcal{T}(q)$, where each component p_h represents the likelihood that the query q is associated with hazard h , and all probabilities sum to 1. A threshold τ is then applied to filter out unrelated hazards, yielding the subset of relevant hazards $\mathcal{S} = \{h \mid p_h \geq \tau, h \in \mathcal{H}\}$. In this study, a threshold of $\tau = 0.2$ is adopted to strike the balance of potential information loss and computational demand. Given the fixed bi-encoder retrieval budget L (same as introduced in RAG), the router proportionally assigns the chunk budget l_h to each selected hazard category h according to its probability:

$$l_h = \frac{p_h}{\sum_{j \in \mathcal{S}} p_j} L, \quad h \in \mathcal{S}. \quad (8)$$

With the router, the retrieval budget is dynamically distributed across hazards according to their relevance scores. Each hazard-specific database $\hat{\mathcal{D}}_h$ then performs retrieval independently under its allocated quota.

During the coarse-grained retrieval stage, each hazard-specific retriever selects the top- l_h chunks based on the cosine similarity between the query and the chunk:

$$\{d_{h1}, d_{h2}, \dots, d_{hl_h}\} = \arg \max_{d \in \hat{\mathcal{D}}_h}^{l_h} \mathcal{L}_b(d, q), \quad (9)$$

The retrieved chunks from all relevant hazards are then aggregated into the candidate chunks set $C = \bigcup_{h \in S} \{d_{h1}, \dots, d_{hl_h}\}$, representing the multi-hazard evidence set for refinement.

A fine-grained reranking stage uses a cross-encoder that takes the query and a candidate chunk as a single combined input and encodes them together. The final retrieval output selects the top- k chunks across the multi-hazard candidate chunks:

$$\mathcal{R}(\mathcal{D} \mid q) = \arg \max_{d \in C} \mathcal{L}_c(d, q). \quad (10)$$

4.2.3. Agentic LLM Structure

The agentic LLM structure functions as the information validation and enhancement layer beyond the Mixture-of-Retrieval (MoR) module. Its primary goal is to verify the quality of retrieved evidence and continuously refine it through iterative loops until trustworthy information is obtained.

This framework consists of five cooperative agents that collectively manage evidence retrieval, validation, reflection, and external data search. The overall workflow is illustrated in Algorithm 1, where the left column outlines the procedural pipeline and the right column highlights the agents operating at each stage. In each iteration, the MoR retriever first gathers hazard-aware evidence from the relevant databases. The Evidence Evaluator then examines whether this evidence is sufficient to answer the user’s query. If not, the Online Search Agent supplements the evidence with external information sources. When both in-domain and external evidence remain inadequate, the Reflection and Question Rewriter agent revises the query, prompting another retrieval cycle. The process iterates until reliable evidence is obtained or the iteration limit is reached. We set the limit to 5 to balance accuracy and computational cost. Finally, the Answer Writer synthesizes the validated information into a grounded response.

The agents serve distinct yet complementary purposes: (1) *MoR Router Agent* routes hazard-related queries by estimating a probability distribution over all hazard categories, determining which knowledge bases or retrievers should be prioritized; (2) *Online Search Agent* performs lightweight web-based retrieval using external search APIs (i.e., qdrant-client [57]) to supplement online knowledge when local retrieval is insufficient; (3) *Evidence Evaluator Agent* examines the sufficiency and relevance of retrieved evidence, deciding whether it is sufficient to answer the question (output “1”) or if more information is required (output “0”); (4) *Reflection & Question Rewriter Agent* reviews the retrieved content and reformulates ambiguous or ineffective queries to improve retrieval precision through iterative reflection and rewriting; and (5) *Answer Writer Agent* synthesizes validated evidence into final grounded responses, ensuring factual accuracy and coherence across multi-hazard contexts. To ensure reproducibility, detailed prompt designs for all agents are provided in [Appendix C](#).

5. Experiments

This section outlines the experimental design used to evaluate the proposed framework. We first test model performance in the zero-shot setting, where LLMs rely solely on pretrained knowledge without external input. This serves as a baseline for understanding their inherent reasoning capacity in multi-hazard contexts. We then introduce the RAG setting to examine how adding

Algorithm 1 Agentic MoRA-RAG Inference

Require: Generation module \mathcal{G} ; MoR retriever module \mathcal{R} ; Online search agent O ; Evidence evaluator E ; Reflection & Question rewriter Q

- 1: **Input:** User query q , Multi-hazard databases $\{\hat{\mathcal{D}}_h\}$
- 2: **Output:** Grounded answer y
- 3: **for** iteration $t = 1$ to 5 **do**
- 4: MoR module retrieves hazard-aware evidence $C = \mathcal{R}(q, \{\hat{\mathcal{D}}_h\})$ ▷ MoR Router
- 5: Evaluator checks Sufficiency of C to answer q ▷ Evidence Evaluator
- 6: **if** Sufficiency == Yes **then**
- 7: Evidence $E = C$; **break**
- 8: **else if** Sufficiency == No **then**
- 9: Online Search Agent retrieves external evidence $C_o = O(q)$ ▷ Online Search
- 10: Evaluator checks Sufficiency of C_o to answer q ▷ Evidence Evaluator
- 11: **if** Sufficiency == Yes **then**
- 12: Evidence $E = C_o$; **break**
- 13: **else**
- 14: Reflect and rewrite query $q = Q(q, C)$ ▷ Reflection & Question Rewriter
- 15: **end if**
- 16: **end if**
- 17: **end for**
- 18: **if** Evidence E exists **then**
- 19: $y = \mathcal{G}(q, E)$ ▷ Answer Writer
- 20: **else**
- 21: $y = \mathcal{G}(q)$ ▷ Answer Writer (Fallback)
- 22: **end if**
- 23: **Return:** final answer y

report-based knowledge improves factual grounding and contextual understanding. While this enhances overall accuracy, limitations remain when retrieved evidence is incomplete or unrelated to the query. Building on these observations, we evaluate the proposed MoRA-RAG with state-of-the-art RAG frameworks. These benchmarks enable a direct comparison with the proposed approach, allowing us to assess whether its mixture-of-retriever design and agentic verification loop lead to measurable improvements in retrieval quality and answer accuracy.

An ablation study is also conducted to examine the contribution of key components within the proposed framework. The analysis is structured around two research questions (RQs):

- **RQ1:** How does the agentic structure contribute to more effective hazard understanding, and what is the role of each module in improving knowledge extraction and generation?
- **RQ2:** How do different chunking strategies affect the reliability and efficiency of knowledge retrieval across multi-hazard databases?

5.1. Baseline Models and Metric

The evaluation includes two settings: zero-shot and RAG-based. In the zero-shot setting, LLMs generate answers using only their pretrained knowledge without external input. We tested two open-weight models (Gemma series and GPT-oss) and three proprietary models (Gemini-2.5-Flash [34], GPT-5-Nano [58], and Claude-sonnet-4 [59]). The Gemma series [60], ranging from 1B to 27B parameters (1B, 4B, 12B, and 27B), was selected to examine how model complexity influences performance. In the RAG-based setting, three retrieval frameworks were included for comparison: Vanilla RAG, CRAG (Corrective Retrieval-Augmented Generation [46]), and MAIN-RAG (Multi-Agent Filtering Retrieval-Augmented Generation [47]). These models serve as benchmarks for evaluating the performance of the proposed MoRA-RAG. All experiments were conducted using an NVIDIA A100 GPU for open-weight LLMs and via official APIs for proprietary models.

All questions in the *HazardRecQA* dataset are true/false or multiple-choice. Model performance is evaluated using accuracy [61], defined as

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \mathbb{I}(\hat{y}_i = y_i) \quad (11)$$

where N is the total number of questions, y_i is the ground-truth answer, \hat{y}_i is the LLMs' answer, and $\mathbb{I}(\cdot)$ equals 1 if the prediction is correct and 0 otherwise.

5.2. Model performances

In the zero-shot setting, pretrained knowledge alone was insufficient for reliable hazard-specific understanding (Figure 5). This limitation stems from the fact that general-purpose LLMs are not trained on detailed post-disaster observations or the mechanistic knowledge required to interpret hazard interactions, damage mechanisms, or cascading effects. The task in this study is highly specialized, requiring an understanding of multi-hazard processes and engineering context that typical pretraining corpora do not contain. Across different hazard categories, the highest accuracy reached only about 60%, with noticeable variation among categories. The performance disparity suggests model bias, particularly for underrepresented hazards such as tsunamis and wildfires, which exhibited relatively lower accuracy.

When Vanilla RAG was introduced, model performance improved steadily across all LLMs. This gain reflects the advantage of grounding responses in domain material drawn from reconnaissance reports, which provide factual and contextual details missing from pretraining data. Nevertheless, the overall accuracy remained limited, reaching roughly 80% at best. Although accuracy improved, some inconsistencies remained because standard retrieval approaches do not always extract complete or contextually aligned information needed for precise reasoning.

Under the proposed MoRA-RAG framework, all evaluated models showed significant improvement compared with their zero-shot counterparts. Accuracy increased to around 90%, indicating that the framework effectively strengthened factual grounding and reasoning stability. In general, proprietary LLMs tend to outperform open-weight models due to their larger training datasets and extended optimization pipelines [58, 61]. However, within our experiments, the open-weight GPT-oss-20B achieved performance comparable to the proprietary GPT-5-Nano,

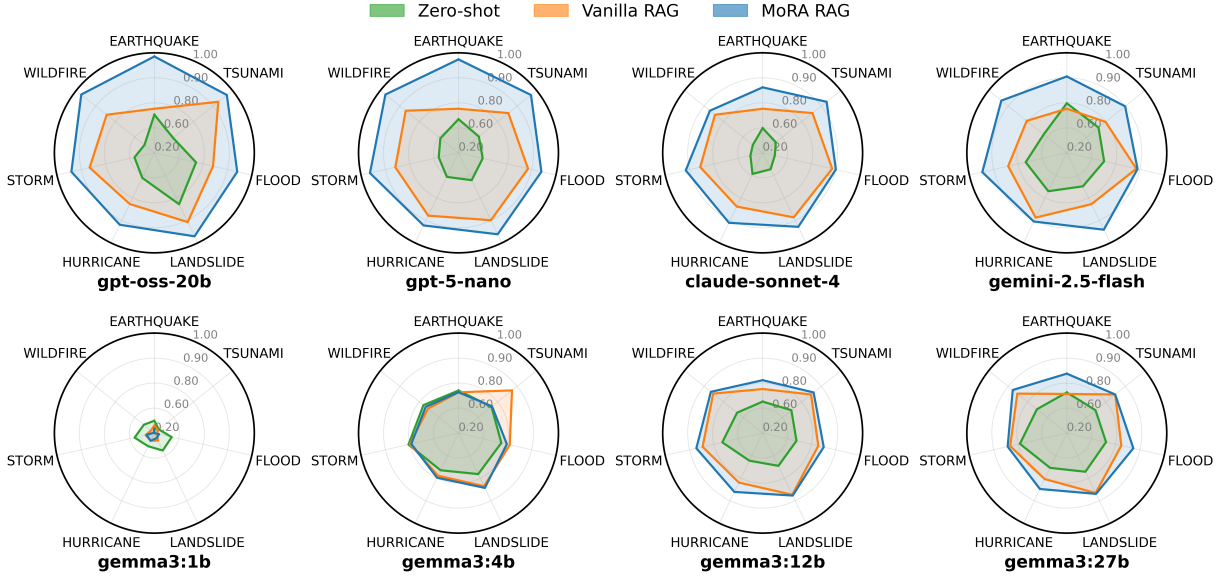


Figure 5: Model performance comparison across different backbone LLMs. First row: different models, second row: same model with different parameter sizes.

demonstrating that well-designed retrieval and verification mechanisms can substantially reduce this performance gap. This result underscores the potential of open-weight models to serve as reliable and transparent alternatives for domain-grounded hazard reasoning. Moreover, MoRA-RAG mitigated bias across hazard categories, producing more balanced performance even for under-represented hazards such as tsunamis and wildfires. These consistent gains across architectures of varying scale highlight the framework’s robustness and its adaptability across both open and proprietary model families.

In the second row of Figure 5, the results show that model complexity strongly influences how effectively the framework integrates retrieved knowledge. Smaller models (i.e., Gemma3:1b) displayed irregular trends, with performance decreasing from the zero-shot setting to Vanilla RAG and further under MoRA-RAG. This pattern suggests that models with limited capacity struggle to process the expanded context and multi-agent reasoning steps, leading to information overload and reduced generation quality. Mid-sized models (i.e., Gemma3:4b) achieved partial improvement, where Vanilla RAG performed better than MoRA-RAG, implying that while retrieval provides useful context, the complete agentic workflow may exceed their optimal reasoning capacity. In contrast, larger models (Gemma3:12B and Gemma3:27B) showed a consistent improvement from the zero-shot setting to Vanilla RAG and then to MoRA-RAG. This pattern indicates that higher-capacity architectures can better utilize and integrate additional information provided by the agentic framework. Overall, these results suggest that the effectiveness of MoRA-RAG scales with model size, as larger models are more capable of handling multi-step retrieval and verification while maintaining stable reasoning performance.

In Table 2, we compare model performance across different question types. Under the zero-shot setting, the results show a consistent pattern across question categories, which aligns with

the performance observed across hazard categories in Figure 5. The accuracy range across the four question types, including analysis approach, hazard characteristics, impacts and damage, and response and recovery, was about 8% across models. This range indicates that without hazard-specific knowledge, the LLMs struggle to maintain consistent reasoning across different tasks. For the RAG-based evaluation, we selected gpt-oss-20b as the backbone LLM because of its strong zero-shot performance and open-weight accessibility, which ensures transparency and reproducibility. All retrieval-based frameworks, including Vanilla RAG, CRAG, MAIN-RAG, and the proposed MoRA-RAG, were implemented on this backbone for consistency. Among them, MoRA-RAG achieved the highest overall accuracy of about 94.5% and showed only around a 1% range across question types (Table 2), indicating more stable and reliable performance.

Table 2: Model performances across models and QA categories (adopted model: gpt-oss-20b as the agent).

Model	Analysis Approach	Hazard characteristics	Impacts & Damage	Response & Recovery	Overall
<i>Zero-Shot</i>					
Gemma3-27B	72.29%	73.08%	75.78%	70.23%	73.54%
gpt-oss-20b	68.14%	69.10%	61.57%	60.00%	65.10%
gemini-2.5-flash	70.10%	74.02%	75.57%	69.32%	73.13%
gpt-5-nano	63.88%	56.20%	61.54%	65.91%	60.38%
claude-sonnet-4	51.45%	42.31%	50.00%	53.93%	49.51%
<i>RAG-based</i> ^a					
Vanilla RAG	87.32%	80.65%	<u>83.03%</u>	88.64%	83.60%
CRAG ^b	<u>88.99%</u>	<u>81.04%</u>	<u>80.23%</u>	<u>89.77%</u>	<u>85.20%</u>
MAIN-RAG	84.45%	79.08%	77.83%	88.64%	81.96%
MoRA-RAG (Proposed)^b	93.78%	94.74%	95.02%	94.32%	94.53%

a. We adopt gpt-oss-20b as the backbone LLM for retriever-based models due to its strong performance in open-weight models (see Figure 5).

b. **Bold** values indicate the best-performing model per category; underlined values indicate the second-best.

5.3. Ablation Studies

5.3.1. RQ1: Agentic Structure and Module Contribution

The ablation study in Table 3 shows that each agentic module contributes incrementally to performance improvement. Compared with the Vanilla RAG baseline (83.6%), adding the mixture-of-retriever module increased accuracy by about 1%, reflecting more targeted retrieval across hazard domains with slightly reduced latency. Incorporating the online search agent yielded a larger improvement of nearly 6%, demonstrating the value of supplementing domain data with external knowledge when in-domain evidence is limited. The reflection and rewriter module provided an additional 8% gain by refining ambiguous queries and strengthening context alignment. When all modules were combined, the complete MoRA-RAG framework reached an overall accuracy of 94.5%, an improvement of almost 11% over the baseline. Although latency increased due to the multi-step reasoning loop, the overall trade-off favors reliability and factual completeness. These results confirm that the agentic structure meaningfully improves reasoning robustness and sup-

ports the conclusion of RQ1 that coordinated agent design is critical for enhancing multi-hazard understanding.

Table 3: Ablation study of model components

Module Setting	Overall Accuracy	Average Latency
Vanilla RAG	83.60%	4.13s
RAG + MoR	84.65% (+1.05%)	3.40s (-0.73s)
RAG + Online Search	89.46% (+5.86%)	7.32s (+3.19s)
RAG + Reflection & Rewriter	91.23% (+7.63%)	18.56s (+14.43s)
MoRA RAG (All components)	94.53% (+10.93%)	20.41s (+16.28s)

5.3.2. RQ2: Chunking and Retrieval Strategy

The comparison in Table 4 highlights the strong influence of chunking strategies on retrieval accuracy. Fixed-token and paragraph-based chunking achieved comparable performance, both reaching around 91% overall accuracy. These methods retain general context but still introduce fragmented semantics and redundant overlaps. The proposition-based method performed notably worse, with an average accuracy of about 72%, indicating that excessive granularity weakens contextual continuity and makes retrieval more sensitive to phrasing. In contrast, the agentic chunking method achieved the highest overall accuracy of 94.5%, improving by nearly 4% over fixed-token and paragraph-based approaches. This method uses an LLM to combine semantically related propositions, summarize contents, and form coherent retrieval units that preserve local context while avoiding information fragmentation. The results confirm RQ2, demonstrating that adaptive, LLM-assisted chunking significantly enhances retrieval precision and semantic consistency in multi-hazard knowledge extraction.

Table 4: Accuracy comparison of different chunking methods for RAG.

Chunking Method	Analysis Approach	Hazard Characteristics	Impacts & Damage	Response & Recovery	Overall
Fixed-token	93.65%	89.79%	91.82%	87.95%	90.80%
Paragraph-based	91.96%	88.25%	91.14%	91.36%	90.68%
Proposition-based	75.77%	70.64%	71.36%	70.91%	71.92%
Agentic chunk (adopted)	93.78%	94.74%	95.02%	94.32%	94.53%

Note: **Bold** values indicate the best-performing method per category.

6. Conclusion

This study introduced an agentic, knowledge-grounded LLM framework (MoRA-RAG) for improving multi-hazard understanding from reconnaissance reports. The framework was designed to address the limitations of conventional retrieval-augmented generation systems, which often suffer from fragmented retrieval, insufficient evidence grounding, and unreliable reasoning in

complex hazard scenarios. Through comprehensive evaluations, MoRA-RAG demonstrated consistent performance improvements across model families, hazard categories, and question types. By integrating a mixture-of-retriever structure and an agentic verification loop, the framework enhanced retrieval precision, reasoning reliability, and factual consistency, achieving an overall accuracy of 94.5%. The ablation study further confirmed that each agentic module (i.e., retrieval routing, online search, and reflection–rewriting) contributed to the overall gain, with coordinated operation yielding the highest performance. Additionally, results across models of varying scale revealed that open-weight LLMs such as GPT-oss-20B can reach performance levels comparable to proprietary models like GPT-5-Nano when equipped with effective grounding and verification mechanisms, narrowing a long-standing performance gap in the field [58].

The findings highlight the potential of domain-grounded LLMs to transform post-disaster reconnaissance data into structured, interpretable knowledge. By enabling more reliable reasoning across heterogeneous hazard information, MoRA-RAG contributes to the growing integration of AI and disaster risk reduction research. Future work will focus on extending the framework to multi-modal datasets that combine textual, visual, and geospatial sources, allowing the model to reason jointly over imagery, maps, and sensor data [27, 28]. Expanding agentic mechanisms toward adaptive learning and multi-hazards generalization also represents a promising direction for advancing data-driven resilience analysis and management.

Appendix A. Details for HazardRecQA Construction

The HazardRecQA dataset is constructed by automatically generating QA pairs from paragraphs in reconnaissance reports. Each paragraph serves as a factual grounding source, and the generation process follows a structured system prompt designed to ensure both accuracy and exam-level clarity. Below are the implementation details:

QA Construction Prompt from Reconnaissance Reports

System Prompt:

Act as a hazard expert, generate an exam-ready QA item grounded in the paragraph.

Context: The **{disaster_type}** occurred in **{year}** at **{location}**.

A paragraph from the reconnaissance report describes the hazard: **{paragraph}**.

Rules:

- 1) Each question must be strictly grounded in factual evidence from the paragraph and reflect hazard-specific or multi-hazard knowledge.
- 2) The question statement must always be written as a general exam-style question, without mentioning the paragraph or the source text.
- 3) If the text is unsuitable (e.g., TOC, Acknowledgements, References):
 - Return { } only.
- 4) Otherwise, create one QA item following the JSON format:
 - True/False → {"statement": <string>, "answer": "true"|"false"}
 - MC (A-D, one correct) → {"question": <string>,
"options": ["A. ...", "B. ...", "C. ...", "D. ..."],
"correct": "A"|"B"|"C"|"D"}
- 5) Categorize the question under the categories: Hazard Characteristics, Analysis Approach, Impacts and Damage, Response and Recovery, Invalid
- 6) No explanations or extra text.

The generated QA items are categorized according to the type of information captured in each paragraph, ensuring multi-dimensional evaluation based on the QA dataset. The four categories, and an invalid filter are summarized below:

- **Hazard Characteristics:** The question describes the event itself, what happened, when and where it occurred, its magnitude, setting, or physical triggers.
- **Analysis Approach:** The question describes how the event was examined through data collection, instruments, surveys, models, or analytical methods.
- **Impacts and Damage:** The question describes the consequences or losses caused by the event, such as structural failures, economic damage, or human impacts.
- **Response and Recovery:** The question describes actions taken after the event, including emergency response, evacuation, repair, or long-term recovery.
- **Invalid:** The question is unusable, presenting vague, contextless, or referring only to figures, tables, or non-substantive content.

Appendix B. Implementation Details for Chunking Methods

In this research, four chunking methods are compared in the ablation study: (1) fixed-token (2) paragraph-based (3) proposition-based and (4) agentic chunking. Each method reflects a different strategy for segmenting long and information-dense reconnaissance reports into coherent units for RAG retrieval.

The fixed-token chunking is implemented with a window size of 200 tokens, supplemented by 50 overlapping tokens before and after each segment, following standard practices in professional document processing [53]. In other words, each 200-token segment forms the core of a chunk, while the additional 50 tokens on both sides provide surrounding context for better continuity.

The paragraph-based method retains the document’s original narrative flow. Reconnaissance reports are usually organized by themes such as event overview, site observation, damage documentation, and analytical discussion; therefore, paragraph segmentation often aligns with meaningful topic boundaries.

The proposition-based chunking targets finer semantic units by utilizing an LLM for chunking. It extracts standalone factual statements from complex paragraphs, rewriting incomplete phrases into full propositions (e.g., “Observed liquefaction along quay walls” to “Investigators observed liquefaction along quay walls”). This approach captures explicit knowledge statement and produce self-contained chunks.

Proposition-based Chunking Prompt

System Prompt:

Act as a hazard domain expert. Decompose the given paragraph into clear, self-contained propositions that can be understood independently of the source text.

Context: The {disaster_type} occurred in {year} at {location}.

A paragraph from the reconnaissance report describes the hazard: {paragraph}.

Rules:

- 1) Split complex or compound sentences into minimal, simple statements.
- 2) Keep original wording whenever possible; ensure each statement stand alone.
- 3) Replace pronouns (it, they, this, that) with full entity names.
- 4) Preserve explicit details such as dates, times (timezone/Z), locations, agencies.
- 5) Output the results in JSON format as a list of propositions:

{**"Prop"**: [**"<proposition_1>"**, **"<proposition_2>"**, ...]}

Example:

Input: The Los Angeles storm occurred in 2015 at Southern California .. at 22:02Z, a funnel cloud was sighted near Lake Hughes and flash flooding with a car stuck in rock and mudslide. .."

Output: {"Prop": [
"During the Los Angeles storm in 2015, at 22:02 Z on October 15, the NWS reported flash flooding near Lake Hughes with a car stuck in a rock and mudslide."]}

The agentic chunking method builds aggregated representations from multiple propositions.

It groups semantically related statements, such as hazard conditions, observed impacts, and contextual factors, under a concise summary that captures the main idea. This approach produces compact, context-aware chunks, resembling how experts synthesize multi-sentence observations in post-disaster analyses. Each chunk contains fewer than ten propositions to maintain focus and ensure internal consistency, avoiding overly long or heterogeneous segments.

This method employs two collaborating agents. The first agent performs proposition extraction, as described in the proposition-based chunking method, to identify and refine standalone factual statements. The second agent performs grouping and summarization, combining semantically related propositions and generating a short summary that encapsulates the shared context. The implementation format is illustrated below.

Agentic Chunk Prompt (Grouping and Summarization)

System Prompt:

Act as a hazard domain expert. Given a list of factual propositions, group semantically related ones and generate concise summaries to form aggregated, context-aware chunks optimized for RAG.

Context: The propositions are provided as: `{list_of_prop}`.

Rules:

- 1) Process propositions sequentially in the given order.
- 2) Group consecutive propositions that are semantically related into one chunk.
When a proposition is unrelated, finalize the chunk and start a new one.
- 3) Each chunk must contain no more than ten propositions.
- 4) For each chunk, write a short summary that captures the shared meaning.
 - Summaries must be concise and precise.
 - Preserve hazard or event details (dates, units, measurements).
 - Use consistent terminology across summaries.
- 5) Output the results in JSON format:


```
{
  "Chunks": [
    {
      "Summary": "<string>",
      "List of Propositions": ["<prop1>", "<prop2>", ...]
    }
  ]
}
```
- 6) Do not alter or invent facts; keep all proposition text unchanged.
- 7) Return only the JSON object, no extra explanations or commentary.

For all final experiments, the agentic chunking strategy is employed as the default setting, due to its performance demonstrated in ablation studies.

Appendix C. Implementation Details for Agents in MoRA RAG

This section details all agent prompts or workflow, adopted agents include MoR Router, Online Search, Evidence Evaluator, Reflection & Question Rewriter and Answer Writer.

MoR Router Agent Prompt

System Prompt:

Act as a routing expert agent. Classify a user's question into probabilities over natural hazard categories. These probabilities determine which hazard-specific RAG agent(s) should be activated.

Question: {**Question**}

Rules:

- 1) Output only a valid JSON object exactly matching the output example below.
- 2) Use the following hazard categories as keys.
- 3) Assign a normalized probability (0-1) to each category so that the total sums to 1.
- 4) Categories with probability ≥ 0.2 indicate active agents.
- 5) Do not include explanations, text descriptions, or additional fields.

Output Example:

```
{
  "Wildfire": 0.01,
  "Storm": 0.10,
  "Landslide": 0.05,
  "Hurricane": 0.61,
  "Flood": 0.21,
  "Earthquake": 0.01,
  "Tsunami": 0.01
}
```

Online Search Agent Workflow

Description:

The Online Search Agent performs real-time factual retrieval from external web sources using a online search API (i.e., DDGS). The agent is non-generative.

Workflow:

- 1) Receive the query and submit to the external search API (i.e., DDGS.text()).
- 3) Collect the top-N textual snippets ($N \leq 5$).

Output Example:

```
{
  "SearchResults": ["Harvey (2017) reached Category 4 before landfall.", ...]
}
```

Evidence Evaluator Agent Prompt

System Prompt:

You are an expert in hazards and resilience. Decide whether the related excerpts contain sufficient information to answer the question or evaluate the statement.

Question: {question}

Related Evidence: {evidence}

Rules:

- 1) Judge the factual sufficiency of the excerpts with respect to the question.
- 2) Do not infer or assume information beyond what is provided.
- 3) Output only a single value:
 - '1' if the evidence provide enough information.
 - '0' if the evidence are insufficient.

Reflection & Question Rewriter Agent Prompt

System Prompt:

You are an expert in hazards and resilience with strong knowledge of information retrieval. Rewrite the given question to improve retrieval accuracy.

Original Question: {question}

Retrieved Insufficient Information: {evidence}

Rules:

- 1) Replace vague expressions with precise, domain-relevant language.
- 2) If the question is too broad, narrow it to highlight key entities or events.
- 3) If the question is too narrow, generalize slightly for broader information.
- 4) Maintain the original intent and semantics of the question.
- 5) Output only the rewritten question, no explanations or commentary.

Answer Writer Agent Prompt

System Prompt:

You are an expert in hazards and resilience. Determine the correct answer based on the retrieved evidence provided by reports or online resources.

Question: {question}

Trustworthy Evidence (if applicable): {evidence}

Rules:

- 1) Judge strictly on the factual information in the evidence.
- 2) Do not infer or assume information beyond what is given.
- 3) Output format:
 - If the question is True/False, answer with one word: true or false.
 - If the question is Multiple Choice, answer with one letter: A, B, C, or D.
- 4) Do not include punctuation, explanations, or additional text.

References

- [1] C. Wu, H. V. Burton, A. Zsarnóczy, S. Chen, Y. Xie, V. Terzić, S. Günay, J. E. Padgett, M. Mieler, I. Almufti, Modeling post-earthquake functional recovery of bridges, *Earthquake Spectra* 41 (3) (2025) 2089–? doi:[10.1177/87552930251321301](https://doi.org/10.1177/87552930251321301).
- [2] J. Verschuur, E. E. Koks, S. Li, J. W. Hall, Multi-hazard risk to global port infrastructure and resulting trade and logistics losses, *Communications Earth & Environment* 4 (1) (2023) 5.
- [3] Y. Zhou, Z. Li, Y. Meng, Z. Li, M. Zhong, Analyzing spatio-temporal impacts of extreme rainfall events on metro ridership characteristics, *Physica A: Statistical Mechanics and its Applications* 577 (2021) 126053.
- [4] S. Zhang, B. Wang, L. Zhang, S. Lacasse, F. Nadim, Y. Chen, Increased human risk caused by cascading hazards — a framework, *Science of The Total Environment* 857 (Part 1) (2023) 159308. doi:[10.1016/j.scitotenv.2022.159308](https://doi.org/10.1016/j.scitotenv.2022.159308).
- [5] M. Rydstedt Nyman, M. Johansson, E. Liljegren, Systematic knowledge sharing in a natural hazard damage context: How organizational borders limit lessons learned, *Risk, Hazards and Crisis in Public Policy* 8 (4) (2017) 356–380. doi:[10.1002/rhc3.12119](https://doi.org/10.1002/rhc3.12119).
- [6] X. Feng, S. Hu, W. Gu, X. Jin, Y. Lu, A simulation-based approach for assessing seaside infrastructure improvement measures for large marine crude oil terminals, *Transportation Research Part E: Logistics and Transportation Review* 142 (2020) 102051. doi:[10.1016/j.tre.2020.102051](https://doi.org/10.1016/j.tre.2020.102051).
- [7] J. K. Hart, K. Martinez, Environmental sensor networks: A revolution in the earth-system science?, *Earth-Science Reviews* 78 (3-4) (2006) 177–191. doi:[10.1016/j.earscirev.2006.05.001](https://doi.org/10.1016/j.earscirev.2006.05.001).
- [8] P. Houston, Hurricane procedures manuals, <https://www.porthouston.com/infrastructure/safety-security/hurricane-procedures/> (2020).
- [9] GEER Association, Geotechnical extreme events reconnaissance (geer) reports database, <https://geerassociation.org/>, accessed on November 2025.
- [10] I. Tomac, B. K. Zelic, D. Perić, D. Domitrović, N. Štambuk Cvitanović, H. Vučenović, J. Parlov, J. Stipčević, D. Matesić, B. Matos, I. Vlahović, Geotechnical reconnaissance of an extensive cover-collapse sinkhole phenomena of 2020–2021 petrinja earthquake sequence (central croatia), *Earthquake Spectra* 39 (28) (2023) 1–34. doi:[10.1177/87552930221115759](https://doi.org/10.1177/87552930221115759).
- [11] S. H. Ro, Y. Li, J. Gong, A machine learning approach for post-disaster data curation, *Advanced Engineering Informatics* 60 (2024) 102427. doi:[10.1016/j.aei.2024.102427](https://doi.org/10.1016/j.aei.2024.102427).
- [12] A. Lenjani, S. J. Dyke, I. Billionis, C. M. Yeum, K. Kamiya, J. Choi, X. Liu, A. G. Chowdhury, Towards fully automated post-event data collection and analysis: Pre-event and post-event information fusion, *Engineering Structures* 208 (2020) 109884.
- [13] E. Markowitz, K. Galiya, G. Ver Steeg, A. Galstyan, Kg-llm-bench: A scalable benchmark for evaluating llm reasoning on textualized knowledge graphs, *arXiv preprintArXiv:2504.07087* (2025). doi:[10.48550/arXiv.2504.07087](https://doi.org/10.48550/arXiv.2504.07087).
- [14] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, *arXiv preprint-ArXiv:2005.11401* (2020). doi:[10.48550/arXiv.2005.11401](https://doi.org/10.48550/arXiv.2005.11401).
- [15] M. Li, M. Luo, T. Lv, Y. Zhang, S. Zhao, E. Nie, G. Zhou, A survey of long-document retrieval in the plm and llm era, *arXiv preprintArXiv:2509.07759* (2025).
- [16] H. Wu, Y. Zhang, C. Ma, F. Lyu, B. He, B. Mitra, X. Liu, Result diversification in search and recommendation: A survey, *IEEE Transactions on Knowledge and Data Engineering* 36 (2024) 5354–5373. doi:[10.1109/TKDE.2024.3382262](https://doi.org/10.1109/TKDE.2024.3382262).
- [17] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Online, 2020, pp. 6769–6781. doi:[10.18653/v1/2020.emnlp-main.550](https://doi.org/10.18653/v1/2020.emnlp-main.550).
- [18] H. Joren, J. Zhang, C.-S. Ferng, D.-C. Juan, A. Taly, C. Rashtchian, Sufficient context: A new lens on retrieval augmented generation systems, *arXiv preprintArXiv:2411.06037* (2024).
- [19] C. Kuai, Z. Li, Y. Zhang, X. B. Wang, D. Lord, Y. Zhou, Us port disruptions under tropical cyclones: Resilience analysis by harnessing multiple-source dataset, *arXiv preprint arXiv:2509.22656* (2025).

- [20] L. G. Brunner, R. Peer, C. Zorn, R. Paulik, T. Logan, Understanding cascading risks through real-world interdependent urban infrastructure, *Reliability Engineering & System Safety* 241 (2024) 109653.
- [21] R. Spiekermann, S. Kienberger, J. Norton, F. Briones, The disaster-knowledge matrix – reframing and evaluating the knowledge challenges in disaster risk reduction, *International Journal of Disaster Risk Reduction* 13 (2015) 96–108. doi:10.1016/j.ijdrr.2015.05.002.
- [22] G. Quitana, M. Molinos-Senante, A. Chamorro, Resilience of critical infrastructure to natural hazards: A review focused on drinking water systems, *International Journal of Disaster Risk Reduction* 48 (2020) 101575. doi:10.1016/j.ijdrr.2020.101575.
- [23] P. K. Paudel, R. R. C. Timilsina, D. Bhusal, H. P. Huntington, Predicting and forecasting disasters: A global scan of traditional and local knowledge, *International Journal of Disaster Risk Reduction* 125 (2025). doi:10.1016/j.ijdrr.2025.014145.
- [24] M. Crawford, K. Crowley, S. Potter, W. Saunders, D. Johnston, Risk modelling as a tool to support natural hazard risk management in new zealand local government, *International Journal of Disaster Risk Reduction* 28 (2018) 610–619. doi:10.1016/j.ijdrr.2018.01.011.
- [25] D. Purwar, J. Flacke, R. Sliuzas, Improving community understanding of cascading effects of critical infrastructure service failure: An experimental interactive learning process, *Progress in Disaster Science* 24 (2024) 100383. doi:10.1016/j.pdisas.2024.100383.
- [26] Y. Nuwara, F. Gynnild, K. Chawshin, J. Barbier, Unlocking insights from unstructured database of petrophysical reports using retrieval augmented generations and large language models, in: 86th EAGE Annual Conference & Exhibition, Vol. 2025, European Association of Geoscientists & Engineers, 2025, pp. 1–5.
- [27] Z. Li, C. Ma, Y. Zhou, D. Lord, Y. Zhang, Leveraging textual description and structured data for estimating crash risks of traffic violation: A multimodal learning approach, *IEEE Transactions on Intelligent Transportation Systems* (2025).
- [28] G. Zhou, K. M. Mosalam, Automated virtual earthquake reconnaissance reporting using natural language processing, *Natural Hazards Review* 26 (3) (2025) 04025018.
- [29] N. Yeşiller, J. L. Hanson, J. Wartman, B. Turner, A. Gardiner, D. C. Manheim, J. Choi, Disaster reconnaissance framework for sustainable post-disaster materials management, *Waste Management* 169 (2023) 392–398. doi:10.1016/j.wasman.2023.07.010.
- [30] Y. Ma, Z. Xie, G. Li, K. Ma, Z. Huang, Q. Qiu, H. Liu, Text visualization for geological hazard documents via text mining and natural language processing, *Earth Science Informatics* 15 (2022) 439–454. doi:10.1007/s12145-021-00732-0.
- [31] J. He, W. Duan, Y. Zhou, Y. Su, Impact of media information on social response in disasters: A case study of the freezing-rain and snowstorm disasters in southern china in 2008, *International Journal of Disaster Risk Science* 15 (2024) 73–87. doi:10.1007/s13753-024-00539-9.
- [32] F. Xu, J. Ma, N. Li, J. C. P. Cheng, Large language model applications in disaster management: An interdisciplinary review, *International Journal of Disaster Risk Reduction* 127 (2025) 105642. doi:10.1016/j.ijdrr.2025.105642.
URL <https://www.sciencedirect.com/science/article/pii/S2212420925004662>
- [33] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).
- [34] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al., Gemini: a family of highly capable multimodal models, arXiv preprint arXiv:2312.11805 (2023).
- [35] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971ArXiv:2302.13971 (2023).
- [36] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, et al., Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, arXiv preprint arXiv:2501.12948 (2025).
- [37] E. Gibney, Secrets of deepseek ai model revealed in landmark paper, *Nature* Published online 17 September 2025 (2025). doi:10.1038/d41586-025-03015-6.
- [38] C.-C. Hung, W. Ben Rim, L. Frost, L. Bruckner, C. Lawrence, Limitations of llms for high-risk domains despite domain-specific instruction tuning, arXiv preprint arXiv:2311.14966ArXiv:2311.14966 [cs.CL] (2023).

- [39] L. Mei, J. Yao, Y. Ge, Y. Wang, B. Bi, Y. Cai, J. Liu, M. Li, Z.-Z. Li, D. Zhang, C. Zhou, J. Mao, T. Xia, J. Guo, S. Liu, A survey of context engineering for large language models, arXiv preprint arXiv:2507.13334ArXiv:2507.13334 [cs.CL] (2025).
- [40] D. Zheng, L. Du, J. Su, Y. Tian, Y. Zhu, J. Zhang, L. Wei, N. Zhang, H. Chen, Knowledge augmented complex problem solving with large language models: A survey, arXiv preprint arXiv:2505.03418ArXiv:2505.03418 [cs.LG] (2025).
- [41] A. Asai, Z. Wu, Y. Wang, A. Sil, H. Hajishirzi, Self-rag: Learning to retrieve, generate, and critique through self-reflection, arXiv preprint arXiv:2310.11511ArXiv:2310.11511 [cs.CL] (2023).
- [42] S. A. Vaghefi, D. Stambach, V. Muccione, J. Bingler, J. Ni, M. Kraus, S. Allen, C. Colesanti-Senni, T. Wekhof, T. Schimanski, et al., Chatclimate: Grounding conversational ai in climate science, *Communications Earth & Environment* 4 (1) (2023) 480.
- [43] M. Juhasz, K. Dutia, H. Franks, C. Delahunty, P. Fawbert Mills, H. Pim, Responsible retrieval augmented generation for climate decision making from documents, arXiv preprint arXiv:2410.23902ArXiv:2410.23902 [cs.CL] (2024).
- [44] T. Merth, Q. Fu, M. Rastegari, M. Najibi, Superposition prompting: Improving and accelerating retrieval-augmented generation, arXiv preprint arXiv:2404.06910 (2024).
- [45] J. Ouyang, T. Pan, M. Cheng, R. Yan, Y. Luo, J. Lin, Q. Liu, Hoh: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation, arXiv preprint arXiv:2503.04800 (2025).
- [46] S.-Q. Yan, J.-C. Gu, Y. Zhu, Z.-H. Ling, Corrective retrieval augmented generation, arXiv preprint arXiv:2401.15884 (2024).
- [47] C.-Y. Chang, Z. Jiang, V. Rakesh, M. Pan, C.-C. M. Yeh, G. Wang, M. Hu, Z. Xu, Y. Zheng, M. Das, et al., Main-rag: Multi-agent filtering retrieval-augmented generation, arXiv preprint arXiv:2501.00332 (2024).
- [48] C. Wang, S. Cheng, Q. Guo, Y. Yue, B. Ding, Z. Xu, Y. Wang, X. Hu, Z. Zhang, Y. Zhang, [Evaluating open-qa evaluation](#), in: *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL <https://neurips.cc/virtual/2023/poster/73580>
- [49] M. A. Ehsan, A. M. Hasan, K. B. Shahnoor, S. S. Tasneem, Automatic question & answer generation using generative large language model (llm), arXiv preprint arXiv:2508.19475ArXiv:2508.19475 [cs.CL] (2025).
- [50] K. Li, Y. Zhang, Planning first, question second: An llm-guided method for controllable question generation, in: *Findings of the Association for Computational Linguistics: ACL 2024*, Association for Computational Linguistics, Singapore (Hybrid), 2024, pp. 4715–4729, paper 280 in the Findings volume.
- [51] W. Chan, A. An, H. Davoudi, A case study on chatgpt question generation, in: *2023 IEEE International Conference on Big Data (Big Data)*, 2023, p. –. doi:10.1109/BigData57512.2023.xxx.
- [52] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, in: *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.
- [53] S. R. Bhat, M. Rudat, J. Spiekermann, N. Flores-Herr, [Rethinking chunk size for long-document retrieval: A multi-dataset analysis](#), arXiv preprint arXiv:2505.21700 (2025). doi:10.48550/arXiv.2505.21700. URL <https://arxiv.org/abs/2505.21700>
- [54] A. J. Jimeno Yepes, Y. You, J. Milczek, S. Laverde, R. Li, [Financial report chunking for effective retrieval augmented generation](#), arXiv preprint arXiv:2402.05131 (2024). URL <https://arxiv.org/abs/2402.05131>
- [55] OpenAI, [New embedding models and api updates](#), accessed: 2025-11-06 (2025). URL <https://openai.com/index/new-embedding-models-and-api-updates/>
- [56] Y. Yu, W. Ping, Z. Liu, B. Wang, J. You, C. Zhang, M. Shoenybi, B. Catanzaro, Rankrag: Unifying context ranking with retrieval-augmented generation in llms, in: *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, 2024.
- [57] Qdrant Team, Qdrant: Vector search engine for the next generation of ai applications, <https://qdrant.tech>, accessed: 2025-10-25 (2025).
- [58] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).
- [59] Anthropic, Claude sonnet 4 [large language model], retrieved August 2025, from conversation with Claude

- (2024).
- [60] T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, P. G. Sessa, et al., [Gemma: Open models based on gemini research and technology](#), arXiv preprint arXiv:2403.08295 (2024).
URL <https://arxiv.org/abs/2403.08295>
- [61] C. Kuai, C. Wu, Y. Zhou, X. B. Wang, T. Yang, Z. Tu, Z. Li, Y. Zhang, Cyportqa: Benchmarking multimodal large language models for cyclone preparedness in port operation, arXiv preprint arXiv:2508.15846 (2025).