Attr-RAG: Attribution-Guided Retrieval-Augmented Generation for Scientific Experiment Design

Fazle Rahat[1], M Shifat Hossain[1], Arvind Ramanathan[2], Sumit Jha[3], Hao Zheng[1], Rickard Ewetz[3]
[1]*University of Central Florida*, Orlando, FL
[2]*Argonne National Laboratory*, Lemont, IL
[3]*University of Florida*, Gainesville, FL
{fazle.rahat, mshifat.hossain, hao.zheng}@ucf.edu,
ramanathana@anl.gov, {sumit.jha, rewetz}@ufl.edu

***Abstract*—Evidence-based science depends on the iterative integration of experimentation, a process traditionally driven by slow and error-prone human effort. This has inspired the vision of an automated ”robot scientist” capable of conducting end-to-end experimentation. While Large Language Models (LLMs) can generate procedural instructions, they often struggle to accurately describe scientific experiments due to the limited availability of high-quality, domain-specific examples in their training data. Retrieval-Augmented Generation (RAG) helps bridge this gap by allowing LLMs to access up-to-date external information. However, despite being effective for short questions, RAG struggles with long-form scientific experimental queries due to information loss from chunk fragmentation and retrieval of irrelevant information. In this paper, we propose Attr-RAG, an attribution-guided RAG framework to remove irrelevant or misleading context and retaining only complete, relevant information. Unlike traditional RAG methods that rely solely on vector similarity, Attr-RAG introduces a refinement stage using occlusion-based attribution to identify which retrieved chunks truly influence the LLM's response. This attribution-guided filtering ensures that only contextually coherent chunks are used for accurate and grounded final answer generation. Attr-RAG demonstrated superior performance in 9 out of 10 chemistry lab experiment tasks of the ChemEx dataset and outperformed baselines across most quantitative evaluation metrics. In qualitative evaluations conducted by state-of-the-art LLM judges (GPT-4o, Gemini 2.5, and Grok 3), the top mean scores of 27.8, 27.1, and 22.9, respectively, were achieved across six key evaluation criteria.**

***Index Terms*—Attribution guided RAG, automatic scientific lab experiment, complete and coherent context**

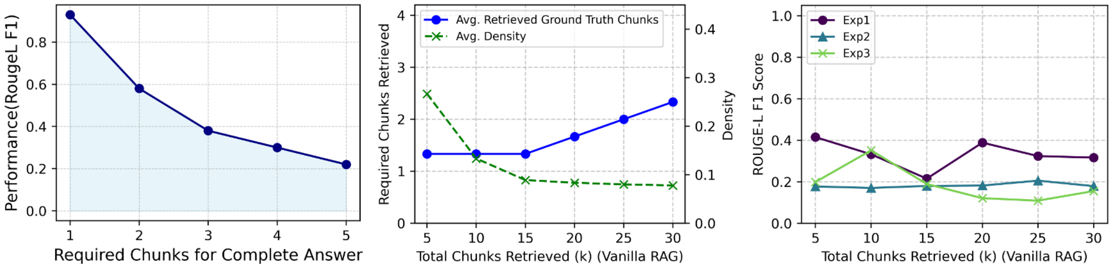
I. Introduction

Evidence-based science depends on an iterative loop of observation, experimentation, analysis, and synthesis, a process still largely constrained by human effort [1]. These challenges are particularly evident in complex domains like biology, chemistry, and materials science, where experimental protocols are intricate, error-prone, and difficult to reproduce. This has led to the growing vision of autonomous AI-driven systems capable of designing, executing, and refining experiments with minimal human intervention [2]. Robotic platforms such as ADAM [3] and EVE [4] have demonstrated early progress toward this vision by automating laboratory execution. Complementing these advances in physical automation, large language models (LLMs) like GPT-4 [5], LLaMA [6], Gemini [7], and Claude [8] have shown remarkable success in natural language processing, reasoning, and task planning [9]. Their ability to translate natural language into procedural instructions makes them promising interfaces for controlling robotic labs. Despite their impressive capabilities, LLMs remain inherently limited by static training data [10, 11]. Consequently, they often fail to provide accurate responses for recent developments or domain-specific queries absent from their training corpus [12, 13].

RAG overcomes these knowledge limitations and improves the factual accuracy by providing additional context to the LLM before generating a response [14]. However, current RAG systems still face challenges in the domain of long descriptive scientific question answering. Generating coherent, long-form responses grounded in source documents is challenging when the retrieved context is fragmented or incomplete. Standard RAG systems typically perform a single round of retrieval, often returning isolated text chunks without maintaining the flow of information [15]. As a result, LLM responses may include relevant facts but lack continuity, omit intermediate steps, or introduce hallucinations due to missing context. Moreover, when multiple document chunks discuss the same topic from different perspectives such as one describing the risks of generating oxygen gas and another outlining the procedure, current RAG systems often struggle to retrieve relevant context. This semantic overlap makes it difficult to select appropriate context, resulting in incomplete or incoherent answers. Recent RAG improvements employ reranking with cross-encoders to better match query–chunk pairs and select more relevant context [16, 17]. Nevertheless, in long scientific tasks, reranking isolated chunks often fails to capture multi-chunk dependencies (Section III).

To address this, we propose a two-stage Attribution-Guided Retrieval-Augmented Generation (Attr-RAG) framework. Our method aims to filter out irrelevant content from the initially retrieved context, enabling the language model to generate responses grounded. In the first stage, a vanilla RAG retrieves a broad set of candidate chunks, which are then used to generate an initial LLM response. In the second stage, we perform an occlusion-based attribution analysis on the LLM's behavior to identify which retrieved chunks (or contiguous sets of chunks) most significantly influence its initial response.

1946-0759/25/$31.00 ©2025 IEEE
DOI 10.1109/ICMLA66185.2025.00064



(a)

(b)

(c)

Fig. 1: Chunk retrieval quality and its impact on LLM performance (a) LLM response quality decreases as the number of required chunks increases. (b) Retrieved ground truth chunks and their density remain low despite increased retrieval. (c) ROUGE-L F1 does not improve with more chunks, showing noise from irrelevant context degrades response quality.

Only these highly influential and contiguous chunk sets are retained as the final context. This refined context is then used to regenerate complete and contextually coherent responses without irrelevant or distracting information. This paper makes the following key contributions:

- We eliminate irrelevant or misleading content from the initial retrieval by using occlusion-based attribution.
- We restore contextual continuity by identifying adjacent influential chunks that enable the model to recover information lost due to fragmentation.
- Our method reduces hallucinations by grounding LLM responses in a well-aligned semantically relevant context.
- Our framework outperforms standard RAG in 9 of 10 ChemEX tasks and achieves an average best LLM judge GPT-4o score of 27.8 across six evaluation criteria.

The remainder of this paper is organized as follows. Section II reviews related work. Section III provides a motivating case study, the proposed methodology in Section IV and experimental results are in Section V. Finally, Section VI concludes the paper and outlines directions for future work.

II. Related Works

RAG integrates external documents as an LLM's context to extend the model's static knowledge base [14]. By retrieving relevant text at query time, RAG improves factual accuracy and provides verifiable evidence for the model's outputs [18, 19]. This paradigm mitigates hallucinations and domain knowledge gaps without requiring expensive model fine-tuning, instead injecting up-to-date information directly into the prompt. Since its introduction for knowledge-intensive NLP tasks [14], RAG has been widely adopted across open-domain QA and specialized domains. For example, RAG-based systems now drive biomedical question answering [20], legal QA with statute and case retrieval[21], and scientific literature assistants [22], demonstrating the approach's versatility in augmenting LLMs with factual grounding in various fields.

To improve the quality of retrieved context in RAG pipelines, researchers have developed reranking methods. In these setups, an initial dense retriever pulls a pool of candidate passages, then a cross-encoder reranker scores each passage in context to select the most relevant chunks [23]. Such rerankers (often trained on large-scale benchmarks like MS MARCO [24]) leverage richer query–passage interactions and yield more relevant results than embedding similarity alone [14]. While reranking helps surface individually relevant passages, it does not guarantee a coherent long-form context. Both standard RAG and rerankers struggle when a question requires step-by-step reasoning across multiple passages. The chunking of documents into separate passages often breaks the logical flow required for detailed answers, yielding fragmented context. As observed in [25], the chunking strategy can disrupt a document's global context, often leaving important information incomplete and forcing the model to fill in the gaps using its parametric memory. Increasing the number of retrieved passages (high top-k) might include all necessary pieces, but at the cost of introducing substantial noise that can confuse generation [25, 26]. Thus, long-form answers that span multiple chunks remain a challenge for traditional RAG.

Feature attribution methods, widely used in NLP, quantify the influence of each input segment on a model's prediction [27, 28]. In particular, occlusion-based attribution [29] measures the importance of input segments by measuring changes in the model's output when specific parts of the input are masked or removed. Inspired by these ideas, we propose Attr-RAG a two-stage retrieval framework that incorporates attribution to guide the selection of relevant context for LLMs. This framework first uses a dense retrieval mechanism [30] to select initial chunks and generate a initial response using LLM. Then, it utilizes occlusion-based attribution to evaluate each chunk set's contribution to the initial response, filtering out irrelevant content and selecting a complete, noise-free context.

III. A Motivating Case Study

We conduct a case study to evaluate the retrieval quality of vanilla RAG using the ChemEX dataset. We use the Qwen2 LLM model to generate responses based on the retrieved context. Our case study is designed from two perspectives: (1) we examine how the LLM's response quality changes when the required context spans one or multiple chunks, and (2) we explore the relevance and completeness of the retrieved context to assess the overall response quality of the vanilla RAG. The implementation details are provided in Section V.

LLM response quality for different number of chunks: To perform this experiment, we design queries whose answers

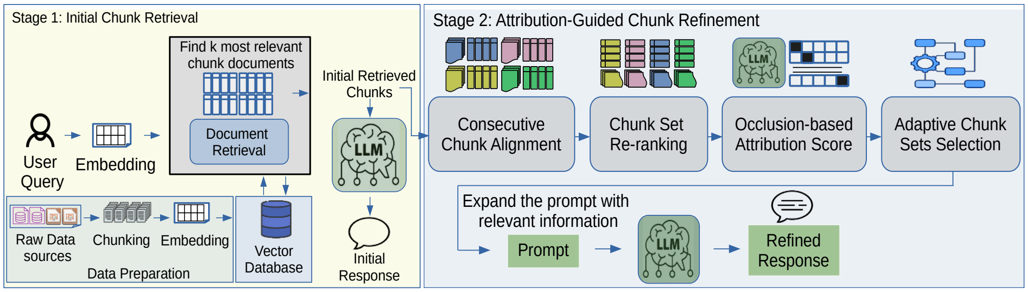

Stage 1: Initial Chunk Retrieval
Find k most relevant chunk documents
Document Retrieval
Initial Retrieved Chunks
User Query
Embedding
LLM
Raw Data sources
Chunking
Embedding
Data Preparation
Vector Database
Initial Response
Stage 2: Attribution-Guided Chunk Refinement
LLM
Consecutive Chunk Alignment
Chunk Set Re-ranking
Occlusion-based Attribution Score
Adaptive Chunk Sets Selection
Expand the prompt with relevant information
Prompt
LLM
Refined Response

Fig. 2: Overview of the Attr-RAG. The framework uses a two-step retrieval-and-refinement process to answer long scientific questions. First, a dense retriever gathers an initial pool of candidate chunks, aligns them into contiguous spans, and re-ranks these grouped contexts. Next, an occlusion-based attribution analysis identifies influential chunk sets and adaptively selects relevant and noise-minimized context for LLM that enables complete and coherent long-form scientific answers.

span from one to five chunks. We then use vanilla RAG to retrieve the context and evaluate the LLM response quality for these queries. As shown in Figure 1(a), the x-axis represents the number of chunks required to fully answer each query, while the y-axis shows the ROUGE-L F1 score comparing the LLM-generated response against the reference answer. We observe in Figure 1(a) when a query requires only a single chunk, the LLM achieves approximately 90% accuracy. However, as the number of required chunks increases to 3, 4, and 5, the performance drops significantly falling below 40% and nearing only 20% for 5-chunk responses. This demonstrates a clear degradation in response quality as the context length increases with number of chunks. For example, short factual questions such as “What is used as the recrystallization solvent for acetaminophen?” can be answered using a single chunk and achieve high ROUGE-L scores. In contrast, descriptive queries like “Explain all the steps of acetaminophen” require multiple chunks to provide a complete response. Here we observe that, for long descriptive scientific questions, vanilla RAG often fails to retrieve intermediate chunks effectively.

Chunk Retrieval Quality: We utilize long descriptive experiments (Exp1, Exp2, Exp3) from the ChemEx dataset. For each experiment-specific query, we identify the ground truth as 3 chunks for Exp1 and 4 chunks each for Exp2 and Exp3. Then we calculate the mean retrieved groud truth chunks and the mean density of ground truth chunks out of all retrival. In Figure 1(b), the x-axis represents the total number of chunks retrieved by vanilla RAG, while the left y-axis shows the average number of ground truth chunks retrieved, and the right y-axis shows the average density of relevant chunks within the retrieved set. We observe in Figure 1(b), when k = 5, vanilla RAG retrieves approximately 1.3 ground truth chunks with a density of 0.27. However, as k increases, the number of ground truth chunks retrieved gradually improves (reaching about 2.3 at k = 30) while the density sharply declines and plateaus around 0.08. This indicates that most additional chunks retrieved are irrelevant or noisy. We also evaluate the LLM response quality for these three experiments. As shown in Figure 1(c), the best performance remains around 40%. In Exp2 and Exp3, vanilla RAG retrieves only 2 out of 4 ground truth chunks, resulting in degraded performance due to fragmentation. In Exp1, even when all 3 ground truth chunks are retrieved within the top 30, the performance drops to 35% due to interference from irrelevant and noisy chunks.

These limitations motivate us to propose the Attr-RAG framework. This framework reduces irrelevant content from the retrieved context and replaces it with new, semantically relevant chunks that are more likely to improve the performance of the LLM’s response.

IV. Methodology

In this section we present the methodology of the Attr-RAG framework for long-form scientific question answering. An overview of this two-stage approach is illustrated in Figure 2. In Stage 1, the system performs an initial retrieval of text chunks using dense embeddings. In Stage 2, it refines this retrieved context through an attribution-driven feedback loop that filters and enriches the chunks provided to the LLM.

A. Step1: Initial Chunk Retrieval

In the first stage of Attr-RAG, we follow a standard dense retrieval pipeline similar to classical RAG frameworks. The system takes a user query q and a set of candidate document chunks $\{c_1, c_2, \ldots, c_n\}$ and maps them into a shared dense vector space using a bi-encoder architecture. The bi-encoder independently encodes the query and document chunk into fixed-size vectors, denoted as:

$$\mathbf{q} = E(q), \quad \mathbf{c}_i = E(c_i) \quad \text{for all } i \in \{1, 2, \ldots, n\},$$

where $E(\cdot)$ is a pre-trained embedding function (e.g., Dense Passage Retriever encoder). We then compute the cosine similarity between the query and each chunk to assess semantic relevance. The similarity score $s(q, c_i)$ for a chunk c_i is computed as:

$$s(q, c_i) = \cos(\mathbf{q}, \mathbf{c}_i) = \frac{\mathbf{q} \cdot \mathbf{c}_i}{\|\mathbf{q}\| \, \|\mathbf{c}_i\|}.$$

The top-k chunks with the highest similarity scores:

$$\mathcal{C}_{\text{top}} = \text{Top-}k\left(\{s(q, c_i)\}_{i=1}^{n}\right).$$

These top-k chunks $\mathcal{C}_{\text{top}}$ are retrieved from the vector database and used as external knowledge to augment the LLM’s input prompt to generate the initial response. However, as the top-k selection is based solely on vector similarity, it may include irrelevant chunks (retrieval noise) or omit

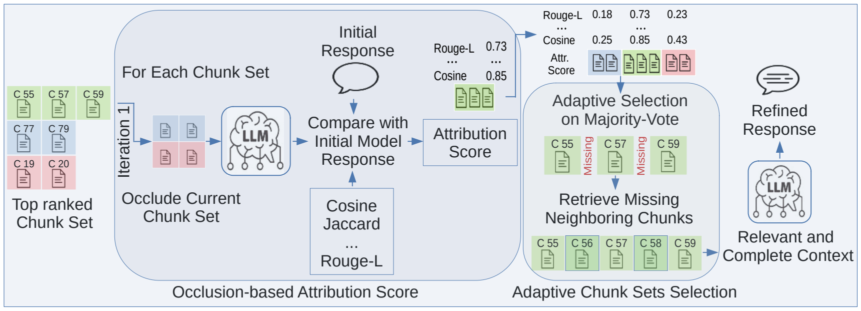


Fig. 3: Detailed view of the final two components in the stage 2 of Attr-RAG. Occlusion-based Attribution Scoring identifies influential chunk sets, while Adaptive Chunk Set Selection constructs a noise-minimized, comprehensive, and complete context.

important information split across multiple chunks of the document (incomplete context). This limitation is particularly problematic for long-form scientific queries that require step-by-step procedural detail and documents discussed closely related topics (shown in Section III). Therefore, we introduce a second refinement stage to filter out noise and retain only the most influential chunks based on their actual contribution to the LLM's response.

B. Step 2: Attribution-Guided Chunk Refinement

In the second stage, we refine the initially retrieved top-k chunks using an attribution-guided feedback loop. The goal is to select a subset of noise free chunk sets that are both highly relevant and influential in generating a high-quality long-form response. This stage consists of four sequential components: (1) consecutive chunk alignment, (2) chunk set re-ranking, (3) occlusion-based chunk attribution, and (4) adaptive chunk selection. The functions of the last two components are further elaborated in Figure 3.

a) Consecutive Chunk Alignment: We first group the retrieved chunks into contiguous sets based on their positions within the original source documents. If two or more of the top-k retrieved chunks originate from the same document and are adjacent in the document's original ordering, they are merged into a single *chunk set*. Each chunk set thus captures a longer, coherent span of context that may encompass multiple related ideas or procedural steps. This grouping helps preserve semantic continuity and mitigates the fragmentation introduced by independent chunking.

To illustrate this, consider a scenario where the retrieval system ranks chunk C_{57} as the first most relevant and C_{59} as the 29th. Although both chunks are adjacent in the document and discuss the same experimental procedure, C_{59} is ranked lower due to slightly lower semantic similarity to the query keywords. If the system truncates the top-k context at $k = 20$, C_{59} which may contain critical continuation of the procedure is excluded, resulting in an incomplete answer. Our chunk alignment step addresses this limitation by grouping semantically and structurally related chunks, ensuring the LLM has access to complete, uninterrupted context.

b) Chunk Set Re-ranking: Next, we re-evaluate the relevance of each chunk set as a whole using a more precise cross-encoder model. Unlike the bi-encoder used for initial retrieval, the cross-encoder jointly encodes both the query and context allowing for finer-grained semantic comparison. The cross-encoder takes the query and the full text of a chunk set as input, and jointly encodes them to predict a relevance score. In our implementation, we employ a pretrained MS MARCO re-ranker (e.g., `ms-marco-MiniLM-L12-v2` [31]), which is a miniature Transformer-based cross-encoder fine-tuned for passage ranking. This model captures detailed semantic alignments between the query and the candidate text that may be missed by the bi-encoder's independent encoding.

c) Occlusion-Based Attribution: After re-ranking, we assess the influence of each chunk set on the generated response using the occlusion-based attribution mechanism shown in Figure 3. The intuition is to measure how the LLM initial response changes when a particular chunk set is omitted from the context. If leaving out a chunk set significantly alters or degrades the initial response, then that chunk set was highly influential. Let C_{all} denote the combined context of all retrieved chunk sets (after re-ranking), and let A_{full} be the LLM's initial response to the query using the full context C_{all}. For each chunk set S_j, we construct an occluded context $C_{\text{all}} \setminus S_j$ (i.e., removing only S_j while keeping all other chunks) and obtain the corresponding occluded response A_{-Sj}. We then compute the attribution score $\alpha(S_j)$ by comparing A_{full} and A_{-Sj} across multiple similarity metrics that capture semantic, lexical, and structural variations in long-form text generation.

Formally, for each metric $M \in \mathcal{M} = \{\text{Cosine}, \text{Jaccard}, \text{ROUGE}_1, \text{ROUGE}_2, \text{ROUGE}_L, \text{ROUGE}_{\text{sum}}\}$, we compute the attribution score $\alpha_M(S_j)$ as:

$$\alpha_M(S_j) = 1 - \text{sim}_M\left(A_{\text{full}}, A_{-j}\right),$$

where $\text{sim}_M(\cdot, \cdot)$ denotes the similarity function defined by metric M. For example, $\text{sim}_{\text{Cosine}}$ represents cosine similarity between sentence embeddings (e.g., Sentence-BERT), while $\text{sim}_{\text{ROUGE}_1}$ refers to the F1 score of unigram overlap. The subtraction from 1 ensures that a higher attribution score reflects greater impact that is, a lower similarity indicates higher reliance on the chunk set S_j.

d) Adaptive Chunk Selection: In the final step, we adaptively select the subset of chunk sets to retain as context for the LLM based on the attribution scores computed across multiple evaluation metrics. Unlike fixed top-k selection, our approach dynamically identifies high-impact context using a combination of majority voting and threshold-aware

TABLE I: Quantitative comparison of Attr-RAG and baseline methods on the ChemEx dataset using seven evaluation criteria: semantic similarity (Cosine), lexical overlap (BLEU, Jaccard), and content fidelity (ROUGE-1, ROUGE-2, ROUGE-L, and ROUGE-Sum F1). We evaluate vanilla RAG, cross-encoder re-ranked RAG with $k \in \{10, 20, 30\}$, and our proposed Attr-RAG. Bolded values indicate the best-performing method per criterion in each task.

Exp	**Eval Metrics**	**Top 10** *LLM vs Lab*	**Top 20** *LLM vs Lab*	**Top 30** *LLM vs Lab*	**Re-rank Top10** *LLM vs Lab*	**Re-rank Top20** *LLM vs Lab*	**Re-rank Top30** *LLM vs Lab*	**Our Method** *LLM vs Lab*
Exp1	Cosine Sim	**0.766**	0.740	0.656	0.358	0.605	0.456	0.707
	BLEU	0.126	**0.331**	0.078	0.012	0.055	0.157	0.278
	Jaccard Sim	0.319	0.465	0.286	0.153	0.226	0.282	**0.489**
	Rouge1 F-measure	0.515	0.668	0.466	0.319	0.377	0.450	**0.746**
	Rouge2 F-measure	0.295	0.424	0.219	0.080	0.186	0.255	**0.474**
	RougeL F-measure	0.371	0.464	0.277	0.153	0.257	0.326	**0.558**
	RougeSum F-measure	0.487	0.650	0.452	0.304	0.368	0.434	**0.707**
Exp2	Cosine Sim	0.709	0.753	0.773	0.596	0.532	0.629	**0.825**
	BLEU	0.233	0.234	0.167	0.033	0.034	0.030	**0.610**
	Jaccard Sim	0.417	0.446	0.437	0.216	0.180	0.207	**0.767**
	Rouge1 F-measure	0.641	0.628	0.574	0.452	0.372	0.418	**0.849**
	Rouge2 F-measure	0.403	0.375	0.375	0.134	0.095	0.108	**0.754**
	RougeL F-measure	0.468	0.460	0.488	0.223	0.192	0.200	**0.784**
	RougeSum F-measure	0.629	0.618	0.568	0.436	0.361	0.399	**0.846**
Exp3	Cosine Sim	0.521	0.416	0.598	0.538	0.495	0.542	**0.860**
	BLEU	0.013	0.014	0.035	0.029	0.013	0.012	**0.234**
	Jaccard Sim	0.197	0.187	0.193	0.237	0.201	0.190	**0.443**
	Rouge1 F-measure	0.354	0.356	0.396	0.450	0.407	0.344	**0.654**
	Rouge2 F-measure	0.091	0.071	0.114	0.095	0.080	0.079	**0.417**
	RougeL F-measure	0.166	0.165	0.205	0.190	0.168	0.168	**0.485**
	RougeSum F-measure	0.320	0.326	0.377	0.408	0.382	0.325	**0.628**
Exp4	Cosine Sim	0.830	0.741	0.739	0.620	0.809	0.555	**0.855**
	BLEU	0.119	0.111	0.116	0.017	0.057	0.005	**0.308**
	Jaccard Sim	0.338	0.356	0.329	0.168	0.275	0.137	**0.554**
	Rouge1 F-measure	0.501	0.562	0.541	0.381	0.504	0.268	**0.726**
	Rouge2 F-measure	0.299	0.267	0.243	0.092	0.192	0.059	**0.498**
	RougeL F-measure	0.362	0.328	0.326	0.171	0.251	0.139	**0.566**
	RougeSum F-measure	0.491	0.540	0.521	0.362	0.482	0.251	**0.707**
Exp5	Cosine Sim	0.742	0.752	0.697	0.479	0.389	0.638	**0.755**
	BLEU	0.034	0.032	0.016	0.004	0.001	0.002	**0.192**
	Jaccard Sim	0.265	0.266	0.182	0.142	0.105	0.148	**0.480**
	Rouge1 F-measure	0.351	0.353	0.278	0.216	0.175	0.184	**0.703**
	Rouge2 F-measure	0.185	0.180	0.117	0.040	0.018	0.081	**0.419**
	RougeL F-measure	0.228	0.239	0.165	0.109	0.095	0.122	**0.516**
	RougeSum F-measure	0.344	0.351	0.272	0.212	0.171	0.182	**0.690**

expansion. We first identify the most influential chunk set S^* via majority voting across the six attribution metrics. Let $\mathcal{M} = \{\text{Cosine}, \text{Jaccard}, \text{ROUGE}_1, \text{ROUGE}_2, \text{ROUGE}_L, \text{ROUGE}_{\text{sum}}\}$ be the set of metrics and $\alpha_M(S_j)$ be the attribution score of chunk set S_j under metric $M \in \mathcal{M}$. We count how many metrics assign S_j the highest attribution among all chunk sets:

$$\text{votes}(S_j) = \sum_{M \in \mathcal{M}} \mathbf{1}\left[S_j = \arg\max_{S} \alpha_M(S)\right]$$

The chunk set S^* with the highest number of votes is selected as the primary influential context. Subsequent chunk sets are selected adaptively based on two criteria: (i) identify the most influential chunk set by majority voting across six attribution metrics, and (ii) expand context selection by dynamically identifying other chunk sets whose attribution scores lie within a small threshold difference from lower-scoring neighbors in multiple metrics, and are consistently selected across at least three metrics. This process captures chunk sets that are not necessarily the top-ranked but exhibit marginal attribution difference from their neighbors.

Once the high-impact chunk sets are selected, we further refine the selection at the chunk level to restore continuity. Scientific questions often require long, contiguous spans of information. Therefore, if the selected chunk sets include non-contiguous chunk indices (e.g., $\{c_{55}, c_{57}, c_{59}\}$), we include intermediate chunks where the gap between two selected chunks is ≤ 2. In the above example, we extend the chunk set to $\{c_{55}, c_{56}, c_{57}, c_{58}, c_{59}\}$. This ensures that fragmented yet contextually important information is not lost due to chunking.

V. Experimental Evaluation

The Attr-RAG framework is implemented in Python, leveraging open-source APIs for LLMs. The experiments are conducted on a system using NVIDIA A100 GPU. In subsequent sections, we present a thorough explanation of the dataset preparation, evaluation procedures, and significant results.

Setup: We introduce the ChemEx dataset comprises ten detailed chemistry lab experiments. For each experiment, we designed queries targeting the step-by-step methodology and construct a corresponding gold-standard answer. To simulate realistic retrieval challenges, we augment each experiment document with additional contextually related and unrelated documents that serve as noise. We introduced two types of noise: (1) contextually similar but non-relevant documents extracted from a chemistry textbook, segmented into smaller parts and inserted around the relevant content; and (2) synthetic documents generated using GPT-4o to simulate diverse perspectives. These synthetic additions include discussions

TABLE II: Qualitative evaluation scores from state-of-the-art LLMs across retrieval methods. We demonstrate the average scores of ten ChemEx experiments evaluated by GPT-4o, Gemini 2.5, and Grok 3 across six criteria: relevance, correctness, completeness, coherence, clarity, and missing steps. Attr-RAG consistently outperforms all vanilla RAG and re-ranking baselines, demonstrating superior quality in long-form scientific answer generation.

Retrieval Method	**GPT-4o Judge**							**Gemini 2.5 Judge**							**Grok 3 Judge**						
	Rel	Cor	Com	Coh	Cla	Mis	Total	Rel	Cor	Com	Coh	Cla	Mis	Total	Rel	Cor	Com	Coh	Cla	Mis	Total
RAG Top 10	4.5	4.0	3.4	4.2	4.3	2.7	23.2	3.5	3.9	2.3	3.5	3.9	2.5	19.6	3.2	2.7	2.1	3.3	2.8	1.6	15.7
RAG Top 20	4.2	3.8	2.9	3.9	4.1	2.8	21.6	3.1	3.3	2.2	3.3	3.6	2.1	17.6	2.9	2.4	1.9	2.9	2.6	1.2	13.9
RAG Top 30	3.9	3.5	3.0	3.9	4.0	2.5	20.6	3.0	3.4	2.5	3.5	3.8	2.3	18.4	2.8	2.1	1.9	2.8	2.4	1.2	13.2
Re-rank Top 10	3.5	3.3	2.7	3.9	3.7	2.6	19.8	2.3	3.1	1.4	3.0	3.5	1.4	14.8	2.1	1.8	1.3	2.5	2.3	0.9	10.9
Re-rank Top 20	2.9	3.3	2.0	3.2	3.9	1.7	17.0	2.5	4.0	1.6	2.5	4.0	2.7	17.2	2.5	2.5	1.8	2.8	2.6	1.2	13.4
Re-rank Top 30	3.1	3.2	2.1	3.2	3.6	1.6	16.8	3.0	3.9	2.1	2.8	3.8	3.0	18.5	2.4	2.1	1.6	2.5	2.4	1.2	12.2
Ours	4.9	4.8	4.3	4.9	4.8	4.0	27.8	5.0	4.7	4.2	4.8	4.8	3.8	27.1	4.3	3.8	3.5	4.4	3.8	3.1	22.9

on potential risks, practical applications, required materials, and the significance of the experiments. For example, for an experiment on generating oxygen gas, distractor texts may discuss the disadvantages of oxygen generation, sourcing of materials, and its real-world applications.

We evaluate the long-form responses generated by LLMs using two complementary approaches: automatic evaluation metrics and LLMs as judges. For the automatic evaluation, we employ standard metrics commonly used for text comparison, including BLEU, cosine similarity, Jaccard similarity, and multiple ROUGE variants. The results of these evaluations are summarized in Table I. However, these automatic metrics are often limited in capturing semantic accuracy and coherence in long-form responses. To address this, we also leverage advanced LLMs specifically GPT-4o, Grok-3, and Gemini 2.5 as evaluators. These models have demonstrated near human performance in evaluating complex text generation tasks and are considered more reliable than traditional metrics for nuanced judgment. We conduct this LLM-based evaluation across six specific criteria, which are defined as follows:

- **Relevance:** Does the generated answer maintain the experimental intent, focus, and components of the reference and align with the scope of the original query?
- **Factual Accuracy:** Are the described procedures, facts, measurements, and terminologies scientifically accurate and consistent with the reference answer?
- **Completeness:** Does the generated response cover all critical steps included in the reference?
- **Coherence and Organization:** Is the response logically structured and organized in a step-by-step manner?
- **Scientific Clarity:** Are chemical concepts, terms, and instructions clearly and precisely articulated in alignment with the scientific rigor of the reference?
- **Missing Step Penalty:** Are any key steps, safety measures, or components omitted in generated answer?

Implementation: In our Python-based implementation, we utilize state-of-the-art open-source models and commercial APIs for embedding, reranking, and response generation. For embedding document chunks and queries, we use the Hugging Face sentence-transformer `all-mpnet-base-v2` [32]. To generate query responses, we employ the instruction-tuned model `Qwen2-7B-Instruct` [33], which demonstrates strong performance in context-aware generation tasks. For reranking retrieved chunks, we use the `ms-marco-MiniLM-L-12-v2` [31] model, which is trained on the MS MARCO passage ranking dataset [24]. While most components of our system rely on open-source models, we also incorporate commercial LLM APIs for qualitative evaluation. We query OpenAI's `GPT-4o` [5], Google DeepMind's `Gemini 2.5` [7], and xAI's `Grok-3` [34] as judges to score the generated responses against gold-standard answers from our ChemEx dataset. The remainder of this section presents the quantitative and qualitative performance of the generated responses.

A. *Quantitative Evaluation using Automatice Metrices*

Evaluating long-form text generation is inherently complex, as no single metric fully captures content relevance, completeness, and coherence [35]. To address this, we adopt a comprehensive evaluation strategy using fifteen automatic metrics, of which we report seven key ones for five experiments in Table I. Our evaluation principle relies on majority metric consensus if most metrics favor a method, we consider its performance superior. Based on this criterion, Attr-RAG outperforms competing approaches in 9 out of 10 experiments.

Across all experiments in Table I, Attr-RAG consistently outperforms standard RAG and re-ranking baselines for top-k chunks (Top-10/20/30) on nearly every metric. Our method achieves higher scores on ROUGE-1, ROUGE-L, and ROUGE-Sum, often with significant margins. For instance, in Experiment 2, Attr-RAG reaches a ROUGE-1 F1 of 0.85 compared to 0.64 with RAG Top-10, and maintains a strong ROUGE-Sum of 0.85, far surpassing re-ranking baselines that rarely exceed 0.43. BLEU and Jaccard scores also show similar trends, indicating that our method generates answers with more precise content and broader lexical overlap. Notably, Cosine similarity scores which measure semantic alignment remain consistently high across tasks, confirming that our method retrieves and integrates more contextually relevant information than all baselines. While re-ranking offers modest improvements over vanilla RAG by filtering some noise, it remains inconsistent across configurations. Adding more documents (Top-30) often introduces excessive noise, which harms generation quality. In contrast, Attr-RAG's attribution-based selection consistently produces cleaner and more informative responses. For example, in Experiment 3, our method achieves a ROUGE-L of 0.49 compared to 0.19 with the best re-ranked baseline. Overall, these quantitative results demonstrate that attribution-guided context refinement allows the model to focus on high-value evidence, yielding more complete, accurate, and semantically aligned responses.

Explain all the required steps of the experimnent of Acetaminophen

Vanilla RAG Response: Synthesizes acetaminophen from 4-aminophenol using acetic anhydride. Steps:

1) Prepare a hot water bath in the fume hood using a 400 mL beaker, thermometer, and hot plate. Heat the water to approximately 90–100°C.
2) Secure an Erlenmeyer ... a clamp above the water bath.
3) In a 125 mL Erlenmeyer flask, ... a stir bar.
4) Set up a vacuum ... a vertical lattice support.
5) Perform the reaction ... within the 90–100°C.
6) Filter the reaction ... solid precipitate.
7) Recrystallize the ... to obtain pure crystals.
8) Determine the melting point ... its identity and purity.
9) Compare the ... desired product.

Re-ranked RAG Response: Key steps:

1) Setting up a hot water bath for heating the reaction mixture.
2) Attaching a vacuum filtration ... a lattice support.
3) Mixing 4-aminophenol, ... in a 125 mL Erlenmeyer flask.
4) Heating the mixture to 90–100°C in the water bath.
5) Stirring the mixture continuously.
6) Cooling the mixture slowly to allow recrystallization.
7) Filtering the product through a Buchner funnel.
8) Purifying the ... melting point range.

Attr-RAG Response: Detailed Synthesizes of acetaminophen steps are:

1) Preparation of hot water bath: A 400 mL beaker, thermometer, and hot plate are used to heat water to around 90–100°C. A clamp is attached to a vertical lattice support to secure an Erlenmeyer flask.
2) Setting up the vacuum filtration: A Buchner ... is pre-weighed.
3) Mixing the reactants: In a 125 mL ... calculations.
4) Heating the reaction mixture: The ... clamp.
5) Cooling the reaction mixture: The ... temperature.
6) Crystallization: An ice bath ... a solid.
7) Filtration**:** The crude ... Buchner.
8) Air drying: The crude ... for 5 minutes.
9) Cleaning and drying the Erlenmeyer flask: The ... recrystallization.
10) Determining the mass of crude product: The mass ... is measured.
11) Obtaining an IR spectrum: The crude ... spectroscopy.
12) Recrystallization**:** The ... crystallization.
13) Vacuum filtering: The ... product.
14) Washing the crystals: Two ... crystals.
15) Air drying: The recrystallized ... minutes.
16) Assembling the TLC development chamber: A 250 ... chamber.
17) Preparing the TLC plate: Three ... product.
18) Spots on the TLC plate: Each ... spot.
19) Preparing the TLC solvent system: Ethyl ... swirling.
20) Developing the TLC plate: The ... UV lamp.
21) Calculating the Rf: The Rf ... calculated.
22) Weighing the recrystallized product: The solid ... reaction.
23) Determining the melting point: The melting point ... determined.
24) Obtaining an IR spectrum: The recrystallized ... spectroscopy.
25) Reporting data: Percent yield, ... reported.
26) Waste disposal: Solid waste is placed in ... glass box.

Fig. 4: Qualitative example responses from Vanilla RAG, Re-ranked RAG, and Attr-RAG. The Attr-RAG response includes ellipses (...) to denote condensed content due to page limits.

B. Qualitative Evaluation using LLM Judge

We conduct human-aligned assessments using three advanced LLM judges GPT-4o, Gemini 2.5, and Grok-3 to evaluate the quality of generated responses. Each response was scored across six critical dimensions: Relevance, Correctness, Completeness, Coherence, Clarity, and Missing Steps Penalty. Due to space constraints, we report the average scores across ten experiments. As shown in Table II, Attr-RAG consistently outperforms both Vanilla RAG and the Re-ranking baseline across all evaluation criteria. It achieves the highest overall mean scores from all three judges, 27.8 from GPT-4o, 27.1 from Gemini 2.5, and 22.9 from Grok-3 demonstrating robust generalization and consistent quality across varied evaluation perspectives. Importantly, Attr-RAG shows strong advantages in categories like completeness and missing steps that matter most for long-form procedural tasks. For instance, under GPT-4o's evaluation, it scores 4.3 in completeness and 4.0 in missing steps penalty, outperforming vanilla RAG (3.4 and 2.8) and Re-ranking (2.7 and 2.6) by significant margins. These gains highlight Attr-RAG's ability to deliver thorough, well-structured answers with minimal omissions.

Figure 4 showcases a representative example from the Acetaminophen synthesis task, highlighting qualitative differences. The Vanilla RAG response omits key stages such as TLC analysis and waste disposal and lacks coherent stepwise structure. The Re-ranking method improves organization but still misses critical procedures like IR analysis and accurate recrystallization. In contrast, Attr-RAG produces a complete and well-structured response, capturing essential steps including mixing reactants, crystallization, vacuum filtration, IR analysis, TLC, and proper waste handling. This example illustrates the strength of our attribution-guided chunk selection by prioritizing context relevance and completeness. Attr-RAG enables LLMs to generate scientifically accurate and procedurally faithful answers aligning closely with the context.

VI. Conclusion and Future Work

We introduced Attr-RAG, an attribution-guided retrieval-augmented generation framework that significantly enhances the quality of long-form scientific question answering. Our method augments the standard RAG pipeline with a second-stage, occlusion-based attribution analysis that selects the context chunks most influential to the model's initial response. This reduces retrieval noise, improves contextual coherence, and preserves critical procedural steps. Our method consistently outperforms standard RAG and re-ranking baselines across ten chemistry experiment tasks in the ChemEx dataset, as measured by both automatic metrics and human-aligned LLM judge evaluations. It achieves superior quantitative results in 9 out of 10 tasks and the highest average qualitative score of 27.8 from GPT-4o. Our findings suggest promising directions for future research in effectively leveraging RAG for long scientific responses with semantically relevant or noisy context. For future work, we plan to generalize the framework to other scientific and instructional domains, explore fine-grained cross-modal attribution for visual reasoning tasks.

Acknowledgement

This material is based upon work supported by the U.S. Department of Energy under grant DE-SC0024428, and by the Defense Advanced Research Projects Agency (DARPA) under Cooperative agreement FA8750-23-2-0501. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied of those supporting this work.

References

[1] R. D. King, J. Rowland, S. G. Oliver, M. Young, W. Aubrey, E. Byrne, M. Liakata, M. Markham, P. Pir, L. N. Soldatova *et al.*, "The automation of science," *Science*, vol. 324, no. 5923, pp. 85–89, 2009.

[2] F. Häse, L. M. Roch, and A. Aspuru-Guzik, "Next-generation experimentation with self-driving laboratories," *Trends in Chemistry*, vol. 1, no. 3, pp. 282–291, 2019.

[3] R. D. King, J. Rowland, W. Aubrey, M. Liakata, M. Markham, L. N. Soldatova, K. E. Whelan, A. Clare, M. Young, A. Sparkes *et al.*, "The robot scientist adam," *Computer*, vol. 42, no. 8, pp. 46–54, 2009.

[4] K. Williams, E. Bilsland, A. Sparkes, W. Aubrey, M. Young, L. N. Soldatova, K. De Grave, J. Ramon, M. De Clare, W. Sirawaraporn *et al.*, "Cheaper faster drug development validated by the repositioning of drugs against neglected tropical diseases," *Journal of the Royal society Interface*, vol. 12, no. 104, p. 20141289, 2015.

[5] OpenAI, "Gpt-4o technical report," https://openai.com/index/gpt-4o, 2024.

[6] H. Touvron, T. Lavril, G. Izacard, and et al., "LLaMA: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023, version 1. [Online]. Available: https://arxiv.org/abs/2302.13971

[7] G. DeepMind, "Gemini 2.5 model card," https://deepmind.google/technologies/gemini, 2024.

[8] Anthropic, "Claude model card and evaluations," https://www.anthropic.com/index/model-card-claude, 2023, white paper.

[9] M. U. Hadi, Q. A. Tashi, R. Qureshi, and et al., "Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects," *TechRxiv*, Nov. 2023. [Online]. Available: https://doi.org/10.36227/techrxiv.23589741.v4

[10] Milvus Blog Team. (2024) How does retrieval-augmented generation help with the issue of an llm's static knowledge cutoff or memory limitations? Accessed 07 May 2025. [Online]. Available: https://blog.milvus.io/ai-quick-reference/

[11] M. Fernández-Pichel, J. C. Pichel, and D. E. Losada, "Evaluating search engines and large language models for answering health questions," *npj Digital Medicine*, vol. 8, no. 153, Mar. 2025. [Online]. Available: https://www.nature.com/articles/s41746-025-01546-w

[12] OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023, sec. 1.3 acknowledges that the model "lacks knowledge of events occurring after its last training data" and may produce outdated or partial responses.

[13] W. Zhang, M. Gomez, and F. Ahmad, "How fresh is your chatgpt? evaluating large language models on post-cutoff and niche knowledge," *arXiv preprint arXiv:2401.01234*, 2024, empirically shows that SOTA LLMs miss or hallucinate 35–50% of facts on news published after their training cutoff.

[14] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, Wen-tauYih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020, pp. 9459–9474. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[15] H. Trivedi, N. Balasubramanian, T. Khot, and A. Sabharwal, "Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, 2023, pp. 18 562–18 578, iRCoT. [Online]. Available: https://arxiv.org/abs/2212.10509

[16] M. Glass, G. Rossiello, M. F. M. Chowdhury, A. R. Naik, P. Cai, and A. Gliozzo, "Re2g: Retrieve, rerank, generate," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Seattle, USA: Association for Computational Linguistics, July 2022, pp. 2701–2715. [Online]. Available: https://aclanthology.org/2022.naacl-main.194

[17] R. Ren, Y. Wang, Y. Qu, W. X. Zhao, J. Liu, H. Tian, H. Wu, J.-R. Wen, and H. Wang, "Investigating the factual knowledge boundary of large language models with retrieval augmentation," *arXiv preprint arXiv:2307.11019*, 2023.

[18] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.

[19] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, P. Merlo, J. Tiedemann, and R. Tsarfaty, Eds. Online: Association for Computational Linguistics, Apr. 2021, pp. 874–880. [Online]. Available: https://aclanthology.org/2021.eacl-main.74/

[20] J. Sohn, Y. Park, C. Yoon, S. Park, H. Hwang, M. Sung, H. Kim, and J. Kang, "Rationale-guided retrieval augmented generation for medical question answering," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds. Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 12 739–12 753. [Online]. Available: https://aclanthology.org/2025.naacl-long.635/

[21] A. Louis, G. van Dijck, and G. Spanakis, "Interpretable long-form legal question answering with retrieval-augmented large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 20, 2024, pp. 22 266–22 275.

[22] J. Lála, O. O'Donoghue, A. Shtedritski, S. Cox, S. Rodriques, and A. White, "Paperqa: Retrieval-augmented generative agent for scientific research. arxiv 2023," *arXiv preprint arXiv:2312.07559*.

[23] M. Glass, G. Rossiello, M. F. M. Chowdhury, A. Naik, P. Cai, and A. Gliozzo, "Re2G: Retrieve, rerank, generate," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, Eds. Seattle, United States: Association for Computational Linguistics, Jul. 2022, pp. 2701–2715. [Online]. Available: https://aclanthology.org/2022.naacl-main.194/

[24] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "Ms marco: A human-generated machine reading comprehension dataset," https://microsoft.github.io/msmarco/, 2016.

[25] Q. Zhao, R. Wang, Y. Cen, D. Zha, S. Tan, Y. Dong, and J. Tang, "Longrag: A dual-perspective retrieval-augmented generation paradigm for long-context question answering," *arXiv preprint arXiv:2410.18050*, 2024.

[26] H. Qian, Z. Liu, K. Mao, Y. Zhou, and Z. Dou, "Grounding language model with chunking-free in-context retrieval," *arXiv preprint arXiv:2402.09760*, 2024.

[27] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[29] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer, 2014, pp. 818–833.

[30] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and 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)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.550

[31] H. F. Contributors, "cross-encoder/ms-marco-minilm-l12-v2," https://huggingface.co/cross-encoder/ms-marco-MiniLM-L12-v2, 2025.

[32] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[33] A. C. Qwen Team, "Qwen2: Open-source large language models," https://huggingface.co/Qwen/Qwen2-7B-Instruct, 2024.

[34] xAI, "Grok-3 model overview," https://x.ai/blog/grok-3, 2024.

[35] K. Krishna, A. Roy, and M. Iyyer, "Hurdles to progress in long-form question answering," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 4940–4957. [Online]. Available: https://aclanthology.org/2021.naacl-main.393/