SynClaimEval: A Framework for Evaluating the Utility of Synthetic Data in Long-Context Claim Verification

Mohamed Elaraby[*]
University of Pittsburgh
mse30@pitt.edu

Jyoti Prakash Maheswari
Zillow Inc.
jyotip@zillowgroup.com

Abstract

Large Language Models (LLMs) with extended context windows promise direct reasoning over long documents, reducing the need for chunking or retrieval. Constructing annotated resources for training and evaluation, however, remains costly. Synthetic data offers a scalable alternative, and we introduce **SynClaimEval**, a framework for evaluating synthetic data utility in *long-context claim verification*—a task central to hallucination detection and fact-checking. Our framework examines three dimensions: (i) *input characteristics*, by varying context length and testing generalization to out-of-domain benchmarks; (ii) *synthesis logic*, by controlling claim complexity and error type variation; and (iii) *explanation quality*, measuring the degree to which model explanations provide evidence consistent with predictions. Experiments across benchmarks show that long-context synthesis can improve verification in base instruction-tuned models, particularly when augmenting existing human-written datasets. Moreover, synthesis enhances explanation quality, even when verification scores don't improve, underscoring its potential to strengthen both performance and explainability.

1 Introduction

Extending the context window of large language models (LLMs) to process thousands and millions of tokens is a promising step toward building systems capable of comprehending long, complex documents without relying on aggressive chunking or retrieval-based pipelines (Liu et al., 2025). However, constructing datasets for both fine-tuning and evaluating long-context LLMs remains labor-intensive and costly, limiting scalability. Synthetic datasets have emerged as a promising alternative to manual annotation, enabling large-scale, low-cost generation of training and evaluation data (Viswanathan et al., 2025). Yet, in the long-context setting, empirical findings remain mixed: some studies report diminished or even negative effects from synthetic long-context training (Gao et al., 2024), while others demonstrate substantial gains over weak long-context baselines (Pham et al., 2025). These discrepancies highlight the need for a systematic evaluation of synthetic data's utility in improving long-context reasoning. In this work, we focus on **evaluating long-context synthesis for long-context claim verification task**.

We pose the following research questions (RQs), addressing both verification performance and explanation quality. **RQ1: How does synthetic long-context training data affect downstream claim?** We study this question along two dimensions: (*i*) the effect of context length on verification accuracy, and (*ii*) the impact of the source domain of the synthetic data on out-of-domain verification benchmarks. **RQ2: How does synthesis logic affect downstream claim verification?** We study this by varying *error types* in unverifiable claims and *claim complexity* in verifiable ones. **RQ3: Does synthetic training improve the quality of model-generated explanations?** We examine whether synthetic tuning improves explanation quality by encouraging rationales that more consistently cite relevant evidence from the input context.

We introduce **SynClaimEval**, an evaluation framework for systematically evaluating the utility of synthetic data in long-context claim verification across the dimensions outlined in our research questions. Figure 1 provides an overview of the framework. For **RQ1**, we vary training context length by truncating source articles, while keeping evaluation benchmarks untruncated as reference, and test both within-domain and out-of-domain settings to assess generalization. For **RQ2**, we manipulate the logic of synthesis along two dimensions: *complexity*, by conditioning on structured representations that induce multi-hop reasoning, and *error type*,

[*]Work done during an internship with Zillow.

Proceedings of the 5th Workshop on Evaluation and Comparison of NLP Systems, pages 91–108

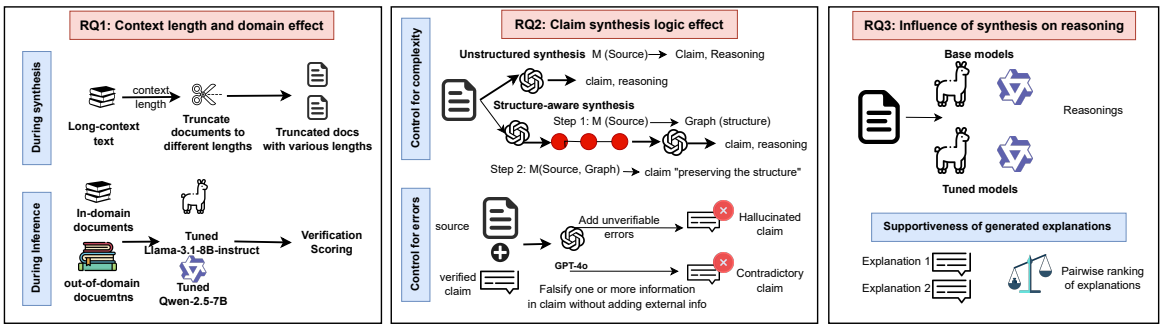

Figure 1: Overview of the **SynClaimEval** pipeline. The framework is designed to evaluate synthetic data along three dimensions: (1) *context length and domain effects*, (2) *claim generation logic*, and (3) *explanation quality*.

by contrasting hallucinated (unverifiable) claims with contradictory ones. For **RQ3**, we evaluate explanation quality through pairwise ranking, asking whether rationales generated under different synthesis strategies offer more support to the same predicted label.

Our study yields five key insights: (i) long-context synthesis enables base instruction-following models to narrow the gap with stronger models, though gains are not always consistent; (ii) extending training contexts improves verification performance; (iii) balancing contradictory and unverifiable (hallucinated) errors yields larger improvements than relying solely on unverifiable errors; (iv) structured synthesis (e.g., multi-hop reasoning) improves performance and generalizes more effectively than unstructured approaches; and (v) although verification gains are modest, synthesis consistently improves explanation quality, independent of verification accuracy improvements.

2 Related Work

Long-context Claim Verification Early work on claim verification largely relied on natural language inference (NLI) models such as `BERT` (Devlin et al., 2019), `RoBERTa` (Liu et al., 2019), and `DeBERTa` (He et al.), which were limited to short contexts (Kryscinski et al., 2020). To adapt these models for longer inputs, prior approaches typically truncated documents (Zha et al., 2023; Zhang et al., 2024) or used retrieval-based strategies (Bishop et al., 2024). More recently, advances in position interpolation and extrapolation have enabled LLMs to process extended contexts directly (Press et al.; Peng et al.), motivating the development of long-context verification benchmarks. For example, Zhao et al. (2024) introduced a financial benchmark where even state-of-the-art models (e.g., Claude-3.5) fall far behind human experts, while Karpinska et al. (2024) proposed a benchmark for verifying claims across fictional books. *In this work, we address long-context claim verification from a broader perspective: rather than targeting a specific domain, we study how synthetic data derived from public benchmarks can serve as effective tuning resources that generalize across diverse long-context settings.*

Synthetic Data in Claim Verification Claim verification can be framed as an entailment task, where most widely used datasets are short-context and human-authored across diverse domains (Bowman et al., 2015; Williams et al., 2018). In contrast, human-written long-context resources are scarce and often domain-specific, such as legal contracts Synthetic data has shown promise in extending verification tasks: for short contexts, Tang et al. (2024) proposed two synthesis pipelines that augmented existing NLI benchmarks, yielding performance comparable to `GPT-4o`. Building on this, Lei et al. (2025) demonstrated that generating claims from context graphs improves over direct prompting, especially for multi-hop reasoning. Results in long-context settings, however, remain mixed. Some studies suggest that short-context synthesis is sufficient for generalization to longer documents (Gao et al., 2024; Bai et al., 2024), while others show that in claim verification—particularly narrative domains—long-context synthesis, often from compressed document representations, yields stronger results (Pham et al., 2025). *In this work, we systematically explore long-context claim synthesis with a strong LLM, evaluating unexplored dimensions such as the effect of error types, varying claim complexity, cross-domain generalization, and the impact of synthesis on explanation quality.*

Row	Content (verifiable-only examples)
Summary (sinppet)	*The report examines the **Senators' Official Personnel and Office Expense Account (SOPOEA)**, which funds staff salaries, travel, supplies, and other office costs.* ***The largest expenditure category** is **personnel compensation**, which accounts for approximately **90%** of total SOPOEA spending . Across selected fiscal years (**2007**, **2008**, **2011**, **2012**), spending categories are largely consistent and overall trends remain relatively stable . There is still variation across spending categories and overall funding levels have decreased or remained flat in recent years . The allocation formula depends on **population** and **distance from Washington, DC**, and the **Senate Appropriations Committee** periodically adjusts SOPOEA limits to emphasize transparency and prudent spending.*
Unstructured claim	**Claim:** ***Personnel compensation** accounts for approximately **90%** of total **SOPOEA** spending.*
Context-graph (entities & path)	**3-Hop Path:** **SOPOEA** $\xrightarrow{\texttt{has\_category}}$ **personnel_compensation** $\xrightarrow{\texttt{accounts\_for}}$ **90%** $\xrightarrow{\texttt{implies}}$ **largest_category** **Claim:** *Within **SOPOEA**, **personnel compensation** constitutes about **90%** of total spending making it the **largest category**.*
Argument-graph (roles & polarity)	**Chain:** Claim ← Premise (opposes) **Generated Claim:** Personnel compensation consistently represents the largest expenditure category in SOPOEA spending, accounting for approximately **90%** of total expenditures, despite variations in other spending categories and overall funding levels.

Table 1: Verifiable claims examples. Entities are **bolded**. Arguments are highlighted: **Claim** , **Supporting Premise** , **Opposing Premise** .

3 SynClaimEval

In this section, we describe the components of our evaluation framework.

3.1 Preparing Claim Sources

Document Truncation For **RQ1**, we examine how context length affects continual supervised fine-tuning (SFT) with synthetic claims. To simulate different source configurations, each document is truncated to a maximum length $T \in \{4{,}096, 8{,}192, 16{,}384\}$ tokens. This design allows us to directly compare models trained on shorter versus longer contexts under identical evaluation conditions, while preserving the integrity of the source.

Compression-based Claim Synthesis. Following CLIPPER (Pham et al., 2025), we synthesize claims from compressed document representations (summaries), which produce less noisy and more cost-effective claims than generating directly from full long-context inputs. We leverage GPT-4o to generate a summary of no more than 1,000 words by instructing the model to produce a concise version of the truncated document. This compressed summary then serves as the source for claim synthesis. To account for domain-specific characteristics in our synthesis sources, we design a dedicated summarization prompt for each domain type[1].

3.2 Claim Synthesis Strategies [2]

We design a synthetic data generation pipeline that produces claims varying along two key axes. First, we control *complexity*: unstructured claims are generated directly from the source text (summaries), while structured claims require multi-hop reasoning either across entities or across discourse/argument units in the context. Second, we vary the *error type*, generating both unverifiable claims that introduce hallucinated content and contradictory claims that embed factual errors. Algorithm 1 outlines the generic synthesis framework.

Unstructured Synthesis. We directly prompt the LLM with (S, D) to generate verifiable claims $C^+ \leftarrow f_{\text{claim}}(S, D)$. To generate error vari-

[1] Summarization prompts are provided in Appendix A

[2] We use GPT-4o as the synthesizer. All prompts are in B

Algorithm 1 Generic Claim Synthesis Framework

Input: (Document D , summary S)
Extract structured representation $I \leftarrow f_{\text{struct}}(S)$
if Unstructured mode **then**
  $I \leftarrow S$
else
  $I \leftarrow f_{\text{struct}}(S)$
  extract structure from text
end if
Generate verifiable claims: $C^+ \leftarrow f_{\text{claim}}(I, S)$
Generate unverifiable variants: $C^u \leftarrow f_{\text{unverif}}(I, S, C^+)$
Generate contradictory variants: $C^c \leftarrow f_{\text{contrad}}(I, S, C^+)$
Output: Synthetic set $\mathcal{S} = \{(D, C^+), (D, C^u), (D, C^c)\}$

ants, we obtain unverifiable claims by $C^u \leftarrow f_{\text{unverif}}(C^+, D)$, which takes the verifiable claim C^+ and inserts plausible but unsupported facts that are not grounded in D. Contradictory claims are obtained by: $C^c \leftarrow f_{\text{contrad}}(C^+, D)$, where f_{contrad} applies common error transformations obtained from the error taxonomy in (Mishra et al.; Devaraj et al., 2022; Pagnoni et al., 2021). Namely we include negation, entity errors, or discourse polarity reversal [3]. Table 1, second row, shows an example of generated unstructured verifiable claim synthesized from the summary.

Context-graph Synthesis. Many claims in long contexts require reasoning over entity relations spanning multiple document segments. To simulate this, we follow the method in (Lei et al., 2025) by constructing a *context graph* $G = (V, E)$ by prompting an LLM to extract entity–relation triplets from summary S. We normalize triplets and form non-branching connected components. From G, we sample multi-hop paths π_{entity} of length up to $k = 3$ [4]. Verifiable claims C^+ are generated by $f_{\text{claim}} : (S, \pi_{\text{entity}}) \mapsto C^+$. Unverifiable claims C^u are obtained by inserting unsupported relations, while contradictory claims C^c are created by corrupting existing edges (e.g., reversing relation types). Table 1, third row, shows an example of an extracted 3-hop path from the entities and how they are aggregated into one single claim.

[3] Appendix C includes error types definitions and examples
[4] More hops do not yield further improvement

Argument-graph Synthesis. Building on prior work in claim verification that leverages composite evidence roles (Habernal et al., 2018), and recent advances in argumentative LLMs that demonstrate improvements in the explainability of verifiable claims (Freedman et al., 2025), we extend these insights to structured synthesis for long-context verification. We introduce a synthesis strategy that leverages *argument graphs* to capture multi-hop argumentative reasoning. In this formulation, we construct an argument graph $A = (V, E)$, where nodes V represent argumentative units (claims or premises) and edges E encode polarity relations (*supports*, *opposes*). Argument roles are extracted from S using an LLM-based argument-mining prompt. From A, we then sample coherent chains π_{arg} that connect a central claim to its supporting and/or opposing premises. This design simulates claim synthesis that relies on reasoning across multiple argumentative evidence, rather than purely entity-based links, exposing models to more discourse-level verification challenges. The remainder of the synthesis pipeline mirrors the context-graph setup: given an extracted chain, we first generate a verifiable claim, which is then perturbed to produce its unverifiable and contradictory variants. Table 1, final row, shows an example of a generated claim based on two rhetorical roles where the premise opposes the claim. The synthesized claim is controlled to capture the relation between them, yielding more complex claims at the sentence level.

3.3 Evaluating Explanations (RQ3)

We assess *justification strength*, i.e., how well an explanation provides valid and sufficient evidence from the context to support the predicted label. Following Elaraby et al. (2024), we frame this as a pairwise ranking task, comparing explanations from different models or tuning strategies against the untuned baseline. Given two explanations (e_i, e_j) for the same claim and predicted label $l \in \{True, False\}$, we use `GPT-4o` to judge which better supports the decision. Each explanation earns 1 point per win and 0.5 per tie:

$$s_i = \sum_{\substack{j=1 \\ j \neq i}}^{M} \mathbb{I}[e_i > e_j] + 0.5 \sum_{\substack{j=1 \\ j \neq i}}^{M} \mathbb{I}[e_i = e_j],$$

where $\mathbb{I}$ denotes the judge's preference. We report average ranking scores across benchmarks.

Variant	Truncation	Total Claims	Verified (n)	Unverified (n)	Claim len (min/mean/max)	Reasoning len (min/mean/max)
Unstructured	4k	14,074	2,815	11,259	6 / 23.17 / 187	13 / 31.83 / 100
	8k	14,072	2,815	11,257	4 / 20.70 / 90	11 / 29.77 / 80
	16k	14,072	2,815	11,257	5 / 23.41 / 102	12 / 31.78 / 87
Context-graph Synthesis	4k	8,403	2,793	5,610	7 / 32.46 / 124	17 / 46.85 / 99
	8k	7,882	2,420	5,462	7 / 32.63 / 111	16 / 43.65 / 118
	16k	8,421	2,803	5,618	7 / 32.71 / 148	16 / 46.82 / 110
Argument-graph Synthesis	4k	7,977	2,672	5,305	6 / 44.95 / 259	16 / 65.86 / 198
	8k	6,156	2,048	4,108	5 / 44.06 / 208	10 / 58.64 / 140
	16k	7,970	2,687	5,283	6 / 45.04 / 473	12 / 65.64 / 221

Table 2: Claim distribution and claim length statistics (in words) across all training synthesis strategies.

4 Datasets

4.1 Synthetic Sources

We construct our synthetic data from widely used, publicly available long-context benchmarks: `PubMed` (Cohan et al., 2018), `GovReports` (Huang et al., 2021), `MeetingBank` (Hu et al., 2023), and `SQuality` (Wang et al., 2022). These datasets were selected to provide a diverse set of domains, enabling us to evaluate the utility of synthesis across varied and openly accessible benchmarks. We uniformly sampled 900 documents from the four datasets, ensuring no overlap with those included in our test benchmarks. Of these, 600 [5] serve as training sources, while the remaining 300 are reserved to construct an in-domain synthetic test set.

Filtration and Truncation. For both training and testing sources, we exclude documents < 1024 tokens. We then apply the pipeline in §3.1. Truncation is applied only to training sources to simulate the effect of context length on benchmarks, while test documents are preserved in their full length.

Obtaining Synthetic Training. We apply both unstructured and structured synthesis strategies as described in §3.2. For each strategy, we sample an equal number of *verified* and *unverified* claims to ensure balanced supervision. To study the impact of error type, we construct two parallel training sets for each synthesis strategy: (1) an *unverified-only* set, where all negative pairs correspond to unverified errors, and (2) a *diverse-error* set, where negative pairs are evenly split between unverified errors (hallucinations) and contradictory errors (balanced across contradiction types). This design allows us to isolate the effect of different error distributions on model training. Table 2 summarizes statistics for the synthetic training datasets across synthesis strategies. Unstructured synthesis yields the largest number of claims, since generating contradictory variants naturally increases error diversity. Truncation has only a minor effect on claim counts and lengths, reducing the risk of confounds when analyzing truncation during fine-tuning. In contrast, structured synthesis produces longer claims and reasoning spans, reflecting our design choice to encourage more complex, multi-faceted examples.

Quality of Generated Claims [6] We employed three annotators to validate the quality of synthetic claims, ensuring no confounding errors from the synthesis process. From the $4k$ unstructured-context set (avoiding longer contexts for efficiency), we sampled 540 claims evenly across types (180 verifiable, 180 unverifiable, 180 contradictory) [7]. Annotators checked each claim's assigned label against its source context, yielding agreement rates of 97.22%, 97.77%, and 99.16% for verifiable, unverifiable, and contradictory claims, respectively—demonstrating the high purity of our synthetic pipeline.

4.2 Evaluation Benchmarks

We evaluate fine-tuning on both synthetic test sets from `SynClaimEval`, aligned with the training distributions, and on publicly available long-document benchmarks with claim- or statement-level support annotations.

`SynClaimEval` We applied the unstructured synthesis pipeline to 300 source documents that were not part of training or any publicly available benchmark. We deliberately avoided constructing a structured synthesis test set in order to assess whether models trained on structured claims can generalize to unstructured settings, where the error distribution differs. In total, we generated 2,500 claims evenly distributed across the labels: verified, unverified, negation, entity error, and discourse error.

[5] Comparable training source sizes are also used in (Pham et al., 2025)

[6] Automatic quality evaluation of synthetic claims is in D and of synthetic explanations in E

[7] Annotators only disagreed on 14 samples out of the 540 IAA $= 0.991\%$

UniSummEval [8] (Wang et al., 2022)is a summarization evaluation benchmark constructed from widely used long-context datasets: PubMed, GovReports, MeetingBank, SQuality, and MediaSumm. Each summary sentence is annotated with a binary label indicating whether it is fully supported by the input context. The benchmark covers both short- and long-context documents; in this work, we focus exclusively on the "long" subset, yielding 5,299 sentence–document pairs. Our motivation for using UniSummEval is to evaluate models tuned on SynClaimEval against a large, multi-domain benchmark that shares the same document characteristics as training, but differs in downstream task framing.

Benchmark	# Pos.	# Neg.	Claim len.	Context len.
SynClaimEval (Test)	500	2000	6/22/76	54/4921/31923
UniSummEval	4897	402	2/23/97	293/3903/10462
FinDver	350	350	11/38/87	4160/39866/69724

Table 3: Statistics of included test benchmarks.

FinDVer [9] (Zhao et al., 2024)is a long-context financial document benchmark in which claim verification requires reasoning across multiple sections of a document. Verifying these claims often entails identifying and correctly interpreting the relevant evidence within the text. We use the *test-mini* split, which contains 700 long financial reports paired with annotated claims and their corresponding reasoning. Our motivation for including FinDVer is to test SynClaimEval on more complex and out-of-domain long-context benchmarks where long context LLMs are known to struggle to verify the claims against them.

Table 3 summarizes the overall statistics of the included test beds. For our in-domain synthetic test set, the average claim length is comparable to that of the UniSummEval benchmark, which is expected given the shared source domains used for synthesis. Among the public benchmarks, FinDver contains the longest documents on average, a characteristic that is reflected in its relatively longer claims. In contrast, UniSummEval shows a strong skew toward positive claims, which is unsurprising since its claims are derived from sentences in generated summaries—a task where LLMs have been shown to perform strongly (Chang et al.).

[8] https://github.com/DISL-Lab/UniSumEval-v1.0

[9] https://github.com/yilunzhao/FinDVer

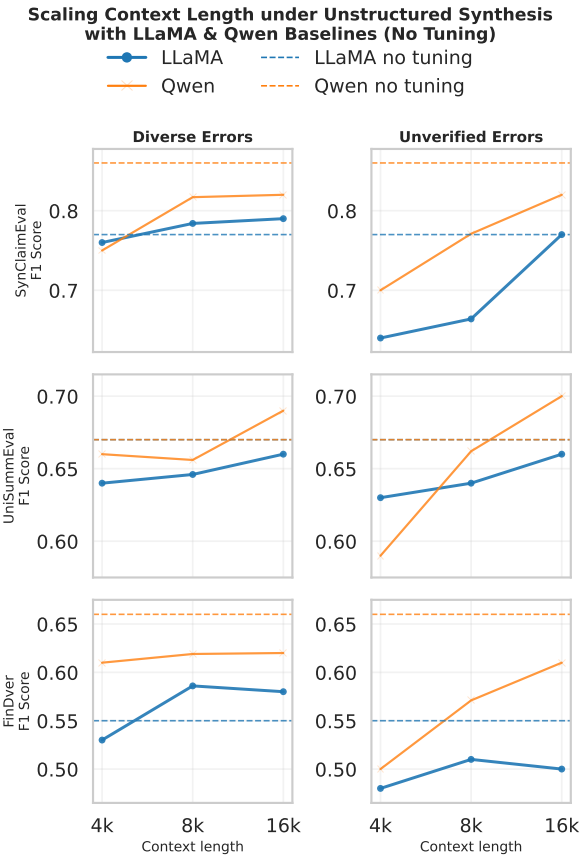

Figure 2: Context length effect on scoring

5 Experimental Setup

5.1 Models and Prompting

We evaluate long-context LLMs with >120k token capacity, including proprietary (GPT-4o, GPT-4o-mini) and open-weight (LLaMA-3.1-8B-Instruct (LLaMa) (Grattafiori et al., 2024), Qwen-2.5-7B-Instruct (Qwen) (Yang et al., 2024), interpolated linearly from 32k→128k). For both inference and tuning, we use the BeSpoke prompt from MiniCheck (Tang et al., 2024), which requires a binary decision (yes/no) and a free-text explanation; decoding temperature is fixed to 0.

5.2 Continual Fine-tuning

Continual SFT is performed with QLoRA (Dettmers et al., 2023) (4-bit, rank=16, $\alpha = 32$), training each model for two epochs.[10] As a baseline, we fine-tune on 16892 human-written samples from ANLI (Nie et al., 2020), following prior work showing short-context tuning may transfer to long contexts (Grattafiori et al., 2024; Gao et al., 2024) and to

[10] Larger ranks/α offered no gains.

measure utility of synthetic long context datasets against human written short ones. For synthetic tuning, we construct 4k balanced pairs (2k verified, 2k unverified), split 85/15 into train/validation. We also evaluate hybrid settings that augment ANLI with synthetic data, extending strategies effective in short-context verification (Tang et al., 2024).

6 Results and Analysis

6.1 RQ1: Context Length and Domain Generalization

Context Length. We first isolate the effect of input length by truncating source documents, holding synthesis complexity fixed through the unstructured variant. Figure 2 shows that for both LLaMA and Qwen, expanding the context window consistently improves verification performance. This pattern is consistent with prior findings (Pham et al., 2025), which similarly reported that longer contexts yield stronger supervision for claim verification. In subsequent experiments, we therefore fix the training context length at 16k to focus on the effect of synthesis complexity (RQ2).

Generalization Figure 2 On in-domain and near-domain tests (SynClaimEval, UniSummEval), LLaMA shows clear gains at 16k over its non-tuned baseline, whereas Qwen underperforms its already strong baseline, which outperforms LLaMA across all benchmarks. This suggests that unstructured synthesis can help weaker models narrow the gap but provides limited benefit for models that already perform well. We further investigate whether more complex claims improve generalization in RQ2.

6.2 RQ2: Error types and synthesis logic

Effect of Error Types. Figure 3 shows that, across benchmarks and models, incorporating diverse error types generally improves verification scores compared to using only unverifiable errors, with the sole exception of SynClaimEval on Qwen. This underscores the value of error-type variation during tuning for enhancing model robustness.

Complexity of claims Table 4 shows that introducing structure into synthesis further shapes model behavior. For LLaMA, structured variants outperform unstructured ones: context-graph synthesis yields moderate improvements, while argument-graph synthesis delivers the strongest results, atleast at lower context sizes. This ordering—*argument-graph* > *context-graph* > *unstructured*—highlights the benefit of conditioning on richer discourse and argumentative structure. In contrast, Qwen again shows limited variation across synthesis strategies, suggesting that structural supervision is more valuable for weaker models that lack strong baseline verification ability.

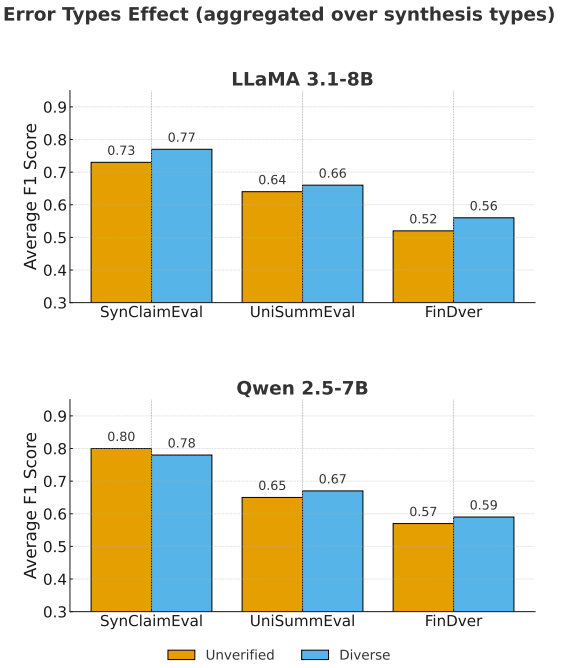

Figure 3: Error types effect

Model / Setting	SynClaimEval F1	UniSummEval F1	FinDver F1
Baselines (Proprietary)			
GPT-4o	**0.97**	**0.71**	**0.81**
GPT-4o-mini	0.93	**0.71**	0.74
Baselines (Open-weight)			
LLaMA-3.1-8B	0.77	0.67	0.55
Qwen-2.5-7B	0.86	0.67	0.66
Unstructured synthesis			
LLaMA-3.1-8B	0.77	0.66	0.50
LLaMA-3.1-8B	*0.79*	0.66	*0.58*
Qwen-2.5-7B	0.82	0.70	0.61
Qwen-2.5-7B	0.82	0.69	0.62
Context-graph (structured)			
LLaMA-3.1-8B	*0.79*	*0.69*	0.52
LLaMA-3.1-8B	0.78	0.68	0.57
Qwen-2.5-7B	0.82	0.70	0.61
Qwen-2.5-7B	0.81	0.70	0.62
Argument-graph (structured)			
LLaMA-3.1-8B	*0.82*	0.62	*0.58*
LLaMA-3.1-8B	0.79	0.66	0.57
Qwen-2.5-7B	0.79	0.69	0.60
Qwen-2.5-7B	0.79	0.70	0.60
Blended Synthetic dataset with and without ANLI			
LLaMA-3.1-8B	0.72	0.65	0.61
LLaMA-3.1-8B	0.81	0.64	0.63
LLaMA-3.1-8B	*0.82*	0.64	*0.65*

Table 4: Performance across benchmarks in F1. Underline = fine-tuned improvements; *Italics* = best among *LLaMA* rows. Diverse errors , ANLI only tuning , ANLI + synthetic mix indicate the type of row.

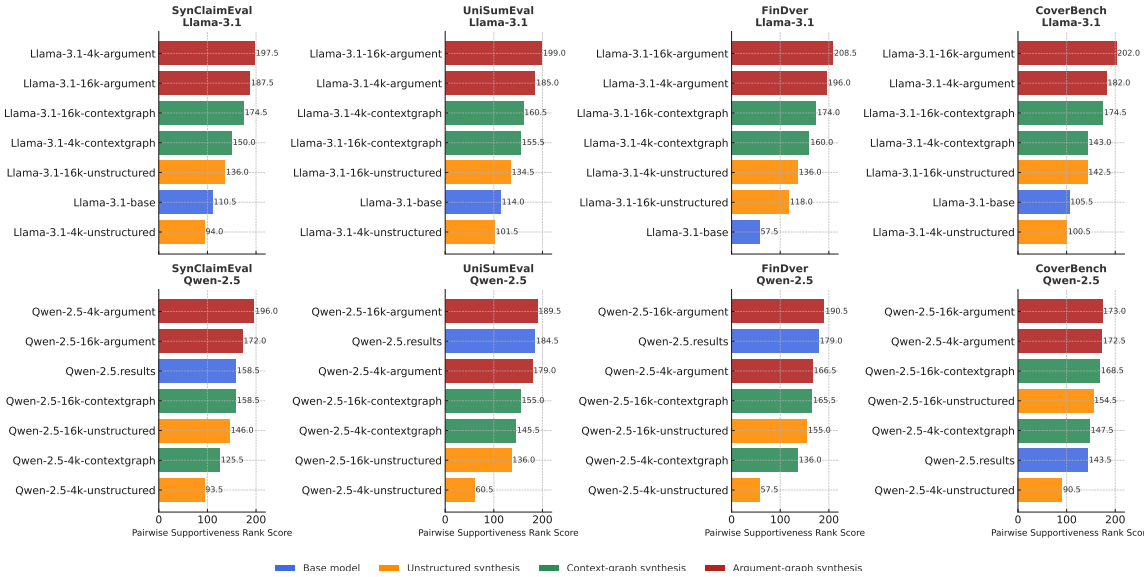

Figure 4: Pairwise supportiveness ranking of explanations across benchmarks. Colors denote synthesis type (Base, Unstructured, Context-graph, Argument-graph). Higher scores indicate stronger judged quality.

Mixing Synthesis Strategies. We evaluate strategy mixing on LLaMA, the model that benefited most from synthesis. Table 4 shows that combining strategies yields higher performance than any single strategy, particularly on FinDver (0.63 F1) and SynClaimEval (0.81), while UniSummEval shows a slight drop. We hypothesize that this decline reflects differences in average context length, as both SynClaimEval and FinDver consist of longer inputs.

Using Synthesis for Augmentation. Table 4 , last 3 rows, shows that augmenting the mixed strategy with ANLI yields the strongest overall results, reaching 0.82 F1 on SynClaimEval and 0.65 on FinDver. These scores surpass tuning with ANLI or synthetic data alone, underscoring the benefits of synthetic claims as complementary augmentation.

6.3 RQ3 Impact on generated explanations

We apply the ranking formula from §3.3 to all synthesis variants. Figure 4 shows that for LLaMa, a consistent ordering emerges across all four benchmarks: *argument-graph* > *context-graph* > *unstructured* > *base model*. The highest ranking scores are obtained by the *argument-graph* variants with $16k$ context length, followed by context-graph based synthesis, while unstructured synthesis trails behind. This ordering mirrors our quantitative results, reinforcing the finding that structured synthesis—particularly when applied with longer contexts—is more beneficial than either unstructured synthesis or no finetuning [11] . By contrast, the trends for Qwen differ. Here, only argument-graph synthesis yields clear improvements over the base model, while context-graph synthesis shows limited gains and unstructured synthesis consistently ranks lowest. This divergence suggests that while synthetic tuning can enhance both prediction scores and explanation quality, its impact depends strongly on the underlying model family. Taken together, these findings highlight both the promise and the limitations of synthetic data: structured synthesis can promote more supportive rationales, but its benefits are not uniformly transferable across architectures.

7 Conclusion and Future Work

We introduced SynClaimEval, a framework for evaluating the utility of synthetic data in long-context claim verification. By disentangling three dimensions—context length, synthesis logic, and explanation quality—we found that synthetic finetuning can improve verification accuracy, particularly under structured synthesis settings that expose models to more complex claims, though these gains are not always consistent. Beyond accuracy, synthetic data proves valuable as an augmentation to human-written claims and more reliably enhances explanation quality, especially with argument-graph synthesis. Looking forward, ap-

[11] Illustrative examples of generated rationales are provided in Appendix F.

plying SynClaimEval to more diverse and domain-specific settings, and combining synthetic with human-annotated data, will be key to understanding the broader impact of synthetic training on long-context reasoning.

Limitations

Our study evaluated several long-context synthesis strategies for claim verification, but important limitations remain. First, we relied on widely available public datasets as synthesis sources. While this choice ensures reproducibility, it also risks overlap with model pretraining corpora. Future work should incorporate more diverse and domain-specific sources to better probe generalization and reduce contamination effects. Second, we restricted training to supervised fine-tuning (SFT). Exploring alternative paradigms—such as reinforcement learning or domain-adaptive pretraining—could reveal different trade-offs between generalization and explanation quality. Third, we limited our experiments to parameter-efficient tuning; extending the framework to full-parameter tuning may yield additional insights. Fourth, scaling synthesis to more challenging domains (e.g., scientific, legal, or financial texts where LLMs often struggle) would clarify how task complexity mediates the benefits of synthetic data. Finally, our explanation-quality assessment relied on LLM-based judges, which, while cost-effective, may introduce biases. Complementing them with human evaluation remains an important direction.

Ethics Statement

This work relies exclusively on publicly available datasets for both synthesis and evaluation, which minimizes risks of handling sensitive or private information. Nevertheless, synthetic data generation may inadvertently amplify biases present in the underlying sources or in the language models used for synthesis. We attempt to mitigate this by sampling from diverse domains and by analyzing multiple synthesis strategies, but acknowledge that residual bias may remain.

Acknowledgment

I would like to sincerely thank my manager during the internship, Matthew Danielson, for his mentorship and steady feedback throughout this work. I am grateful to Amir Rez Rahmani, Bin He, and Winston Quock for their regular feedback and thorough reviews. I also thank the Legal team—especially Abigail Holman—for guidance through the publication process, and the Human-in-the-Loop (HITL) team at Zillow for annotation work and careful reviews. Finally, I thank the LLM Platform (LLMP) team and Taleb Zeghmi for engineering support and access to GPU resources that enabled the experiments.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024. LongAlign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395, Miami, Florida, USA. Association for Computational Linguistics.

Jennifer A Bishop, Sophia Ananiadou, and Qianqian Xie. 2024. Longdocfactscore: Evaluating the factuality of long document abstractive summarisation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10777–10789.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. Booookscore: A systematic exploration of book-length summarization in the era of llms. In *The Twelfth International Conference on Learning Representations*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Ashwin Devaraj, William Sheffield, Byron Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7331–7345,

Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mohamed Elaraby, Diane Litman, Xiang Lorraine Li, and Ahmed Magooda. 2024. Persuasiveness of generated free-text rationales in subjective decisions: A case study on pairwise argument ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14311–14329, Miami, Florida, USA. Association for Computational Linguistics.

Gabriel Freedman, Adam Dejl, Deniz Gorur, Xiang Yin, Antonio Rago, and Francesca Toni. 2025. Argumentative large language models for explainable and contestable claim verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 14930–14939.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. 2024. How to train long-context language models (effectively). *arXiv preprint arXiv:2410.02660*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. Semeval-2018 task 12: The argument reasoning comprehension task. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 763–772.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Yebowen Hu, Timothy Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. MeetingBank: A benchmark dataset for meeting summarization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16409–16423, Toronto, Canada. Association for Computational Linguistics.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Marzena Karpinska, Katherine Thai, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. One thousand and one pairs: A "novel" challenge for long-context language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17048–17085, Miami, Florida, USA. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

Deren Lei, Yaxi Li, Siyao Li, Mengya Hu, Rui Xu, Ken Archer, Mingyu Wang, Emily Ching, and Alex Deng. 2025. FactCG: Enhancing fact checkers with graph-based multi-hop data. 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)*, pages 5002–5020, Albuquerque, New Mexico. Association for Computational Linguistics.

Jiaheng Liu, Dawei Zhu, Zhiqi Bai, Yancheng He, Huanxuan Liao, Haoran Que, Zekun Wang, Chenchen Zhang, Ge Zhang, Jiebin Zhang, et al. 2025. A comprehensive survey on long context language modeling. *arXiv preprint arXiv:2503.17407*.

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

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. Fine-grained hallucination detection and editing for language models. In *First Conference on Language Modeling*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.

Chau Minh Pham, Yapei Chang, and Mohit Iyyer. 2025. Clipper: Compression enables long-context synthetic data generation. *arXiv preprint arXiv:2502.14854*.

Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Vijay Viswanathan, Xiang Yue, Alisa Liu, Yizhong Wang, and Graham Neubig. 2025. Synthetic data in the era of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 5: Tutorial Abstracts)*, pages 11–12, Vienna, Austria. Association for Computational Linguistics.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022. SQuALITY: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1156, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11328–11348, Toronto, Canada. Association for Computational Linguistics.

Huajian Zhang, Yumo Xu, and Laura Perez-Beltrachini. 2024. Fine-grained natural language inference based faithfulness evaluation for diverse summarisation tasks. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1701–1722. Association for Computational Linguistics.

Yilun Zhao, Yitao Long, Tintin Jiang, Chengye Wang, Weiyuan Chen, Hongjun Liu, Xiangru Tang, Yiming Zhang, Chen Zhao, and Arman Cohan. 2024. FinDVer: Explainable claim verification over long and hybrid-content financial documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14739–14752, Miami, Florida, USA. Association for Computational Linguistics.

A Summarization Prompts

Table 5 presents the domain-specific summarization prompts used to compress inputs from various domains to generate synthetic data. Each template is tailored to the conventions of its source domain (e.g., government reports, meeting transcripts, scientific articles, or books), while enforcing common constraints such as conciseness, professional tone, and length limits.

B Claim Synthesis Prompts

B.1 Unstructured Synthesis

Table 6 presents the prompts used to generate verifiable, unverifiable (hallucination-based), and contradictory claims. To ensure a strict 1:1 mapping across verification types, we first synthesize verifiable claims and then apply corruption procedures to derive their unverifiable and contradictory counterparts.

B.2 Context-graph Synthesis Prompts

Table 7 presents the prompt used to extract entity triplets from the input document. Building on these outputs, Table 8 provides the synthesis prompts for generating verifiable, unverifiable, and contradictory claims, each of which consumes the extracted entities as input.

B.3 Argument-graph Synthesis Prompts

Table 9 shows the prompt for extracting argument roles—`claims` and `premises`—along with their support/oppose relations. These roles are assembled into an argument graph, from which connected chains are sampled and passed to the synthesis prompts in Table 10.

C Error types definitions

Table 11 outlines the error granularities considered when synthesizing unverified claims.

D GPT-4o Evaluation of Claim Synthesis

Table 12 captures the quality of synthetic claims across different dataset and context length. We pass the generated claim along with relevant document and leverage GPT-4o as a judge to understand the quality of generated data measured in terms of accuracy

E Evaluating the quality of synthetic explanations

Quality of generated explanations Following (Pham et al., 2025) which evaluated informativeness/faithfulness of the CoT through grounding each step to the input, we evaluate how well generated explanations remain grounded before and after synthesis. We decompose each explanation into atomic facts with `GPT-4.1`, and we compute the proportion of those facts that can be verified against the original context across all synthetic strategies. We sample 100 generated explanations from each synthetic strategy from the verifiable label. At the 4k truncation level, unstructured synthesis achieved 86.12% verified units, context-graph synthesis achieved 80.72%, while argument-graph synthesis attained the highest verification rate at 93.57%. At the 16k truncation level, unstructured (89.39%) and context-graph (88.32%) synthesis improved compared to their 4k counterparts, though argument-graph synthesis remained strong (91.11%). These numbers are in the same range with prior findings of synthetic CoT faithfulness described in (Pham et al., 2025), which showed benefits of synthetic claim generation.

Table 13 shows the prompts for extracting atomic claims from model generated reasoning justifying the final judgment. Once the atomic claims are extracted Table 14 shows the prompts used to evaluate the correctness of the atomic fact and finally evaluated the quality of CoT reasoning used for training the models

F Reasoning Output

Table 16 shows the comparison of model-generated explanation under different synthesis strategies and help understand the impact complex synthesis strategies like Argument-Graph has on model-generated explanations.

Domain	Prompt Template
GovReports	Your task is to write a concise, structured summary for the government report below. Organize your summary into multiple paragraphs. Use a clear, professional tone. Keep the total length under 1000 words. Do not include the full report title in your summary—refer to it generically as "the report." Report {input_text} Summary:
MeetingBank	Your task is to produce a concise, structured "mini" summary of the meeting transcript below (e.g., as in MeetingBank). Treat the summary as a compact representation that captures all essential discussion points and outcomes. Additional requirements: - Keep the summary under 1000 words. - Do **not** include verbatim transcript excerpts—paraphrase in your own words. - Use consistent terminology (e.g., refer to "Project X" the same way throughout). Transcript {input_text} Summary:
PubMed	Your task is to write a concise, structured "mini" version of the scientific document below. Treat the summary as a compact version of the input that retains all critical content. Additional requirements: - Organize the summary into multiple paragraphs. - Use full technical names on first mention, then acronyms thereafter. - Keep the summary under 1000 words. - Do **not** include the document's title or citation details—focus only on content. - Ensure the summary reads as a true "mini" of the input, condensing its essence into a coherent, readable format. Document {input_text} Summary:
SQuALITY / Books	Your task is to write a summary for the book below. Include vital information about key events, backgrounds, settings, characters, their objectives, and motivations. Introduce characters (with full names), places, and other major elements on first mention. The book may feature non-linear narratives (flashbacks, alternate worlds/viewpoints). Organize the summary into a consistent, chronological narrative. The summary must be under 1000 words, span multiple paragraphs, and be written as a single continuous narrative (no bullet lists or outlines). Do **not** include the book name in the summary. Book {input_text} Summary:

Table 5: Summarization prompt templates used for synthetic data generation across four domains. Each template specifies domain-specific constraints and formatting requirements, while maintaining consistency in output length and style. Replace {input_text} with the source document.

Synthesis Type	Prompt Template
Verified	You are given a document. Your task is to extract a list of {num_claims} factual claims from the document. Each claim must: - Be a complete, standalone statement that can be independently verified. - Be factual, atomic, clear, and concise. - Be grounded in the document (no hallucinations). - Be diverse (avoid closely related claims). For each claim, provide reasoning showing why it is factual and supported. Return only the following format: <BEGINFACT>Factual statement<ENDFACT> <BEGINREASONING>Explanation<ENDREASONING> Document: {input}
Unverifiable	You are given a factual claim from a document. Generate a **plausible but unverifiable variant**. It must: - Sound realistic and grammatically correct. - Be related to the topic but include unverifiable information. - Not be explicitly contradictory. Output only: <BEGINUNVERIFIABLE>Unverifiable claim<ENDUNVERIFIABLE> <BEGINUNVERIFIABLEREASON>Reason why unverifiable<ENDUNVERIFIABLEREASON> Document: {document} Claim: {factual_claim}
Contradictory	You are given a factual claim. Generate a **corrupted version** using a specific error type: {error_type}. Error types: - negation (flip polarity) - entity_relation (swap/alter entities or relations) - discourse (flip cause-effect or misattribute support) If not feasible, return <NOT_POSSIBLE>. Output only: <BEGINFALSIFIED>Falsified claim<ENDFALSIFIED> <BEGINFALSEREASON>Reasoning<ENDFALSEREASON> <BEGINERRORTYPE>{error_type}<ENDERRORTYPE> Document: {document} Factual Claim: {factual_claim}

Table 6: Unstructured claim synthesis prompts. Each synthesis type is shaded for clarity: Verified, Unverifiable, and Contradictory. Placeholders {} are replaced with inputs during generation.

Document → Entity Triples Extraction Prompt
Given an article, go over every sentence and extract triples in the form: (entity <TUPLEDELIM> entity <TUPLEDELIM> short description of the relation). Group triples with the same entity together. Separate groups using <GROUPDELIM>. Provided Sentences: {input} Groups of Triples in Provided Document:

Table 7: Prompt for extracting entity–entity–relation triples from a document (**Document → Entities** step).

Context-Graph Synthesis Type	Prompt Template
Verified (uses given entities)	You are given a document. Write a **single factual claim** that **must mention all of the following entities**: **Entities**: {entities} Then provide a **brief explanation** grounded in the document. Output exactly: <BEGINFACT>Your factual claim using all entities.<ENDFACT> <BEGINREASONING>Why the claim is factual and supported by the document.<ENDREASONING> Document: {input}
Unverifiable Variant (same entities)	You are given a factual claim involving the entities {entities}. Generate a **plausible but unverifiable** variant that **introduces at least one relationship not verifiable** from the document (avoid explicit contradiction). Output exactly: <BEGINUNVERIFIABLE>Unverifiable claim with the same entities.<ENDUNVERIFIABLE> <BEGINUNVERIFIABLEREASON>This claim ... (explain why unverifiable without referencing the original claim).<ENDUNVERIFIABLEREASON> **Document:** {document} **Claim:** {factual_claim} **Entities:** {entities}
Contradictory Variant (same entities)	You are given a factual claim involving the entities {entities}. Generate a **contradictory** variant by **flipping or corrupting at least one relationship** among these entities (keep entities unchanged). The new claim must be **contradicted** by the document (not merely unverifiable). Output exactly: <BEGINFALSIFIED>Contradictory claim with the same entities.<ENDFALSIFIED> <BEGINFALSEREASON>This claim ... (explain why contradicted, citing the corrupted relationship).<ENDFALSEREASON> **Document:** {document} **Claim:** {factual_claim} **Entities:** {entities}

Table 8: Context-graph (structured) claim prompts. Row colors indicate type: Verified, Unverifiable, and Contradictory. The triple-extraction step is omitted here for space; this table assumes entities are already provided.

Argument Graph Extraction Prompt (Document → Argument Graph)
Given a passage, extract its argument structure by identifying **claims**, **premises**, and the **relation** between each premise and its claim (supports or opposes). A **claim** is the main assertion. A **premise** is a reason/evidence that supports or opposes the claim. For each claim, list all connected premises with their relation. ### Output Format (repeat per group): <BEGIN_GROUP_CLAIM> <STARTCLAIM>The claim goes here<ENDCLAIM> <STARTPREMISE>Premise text<STARTRELATION>supports or opposes<ENDRELATION><ENDPREMISE> ... (repeat premise blocks as needed) <END_GROUP_CLAIM> Only include relations explicitly inferable from the passage. Do not include general facts, summaries, or hallucinated reasoning. Input: {input_text}

Table 9: Prompt for constructing an **argument graph** from a document (claims, premises, and support/oppose links).

Argument-Graph Synthesis Type	Prompt Template
Verified (from argument chain)	Given an **argument chain** (a central claim with connected premises and their relations: supports/opposes) and the reference document, generate **one concise, overarching factual claim** that synthesizes the core argument. Integrate both supporting and opposing premises faithfully. Provide a brief, document-grounded explanation. Output exactly: <BEGINFACT>Your factual claim synthesizing the chain.<ENDFACT> <BEGINREASONING>Why the claim is factual, grounded in the document.<ENDREASONING> Document: {input} Argument Chain: {argument_chain}
Unverifiable (from argument chain)	Given an **argument chain** and the reference document, generate **one plausible claim** that integrates the chain but introduces an **unverifiable detail** (cannot be confirmed from the document; avoid contradiction). Then explain why it is unverifiable (identify the unconfirmed part). Start reasoning with "This claim...". Output exactly: <BEGINUNVERIFIABLE>Your unverifiable, chain-based claim.<ENDUNVERIFIABLE> <BEGINUNVERIFIABLEREASON>This claim ... (why unverifiable, based on what is missing/uncertain in the document).<ENDUNVERIFIABLEREASON> Document: {document} Argument Chain: {argument_chain}
Contradictory (flip relation in chain)	Given an **argument chain** and the reference document, generate **one concise claim** that **falsifies the original argument** by **incorrectly flipping** at least one premise relation (treat a supporting premise as opposes, or vice versa). The result must be **contradicted** by the document (not merely unverifiable). Then explain why it is falsified, citing the misrepresented relationship. Output exactly: <BEGINFALSIFIED>Your falsified claim that flips a support/oppose relation.<ENDFALSIFIED> <BEGINFALSEREASON>Why this claim is contradicted (what relation was flipped and how the document disagrees).<ENDFALSEREASON> Document: {document} Argument Chain: {argument_chain}

Table 10: Argument-graph (structured) claim prompts spanning two columns. Row colors indicate type: Verified , Unverifiable , Contradictory . This table assumes the argument graph has been extracted using Table 9.

Error Type	Definition / Transformation Strategy
Unverifiable	Produce a claim that sounds plausible but cannot be verified from the source (e.g., by introducing unverifiable details while avoiding explicit contradiction).
Negation	Flip the polarity of the claim to create a false statement (e.g., "X occurred" → "X did not occur").
Entity-Relation	Corrupt entities or their relationships, such as swapping subject/object roles, misattributing actions, or replacing entities with plausible but incorrect ones.
Discourse	Corrupt the logical structure of the claim, e.g., flipping cause–effect, reversing claim and evidence, or misrepresenting support/oppose relations.

Table 11: Error types used in synthetic claim generation. Red rows denote contradictory error types, while unverifiable errors add uncertainty without explicit contradiction.

Dataset	Length	No Error	Unverifiable	Negation	Entity Rel.	Discourse
GovReport	4k	0.88	0.86	0.98	0.82	0.86
	16k	1.00	0.92	1.00	0.80	0.72
SQuALITY	4k	0.92	0.94	0.96	0.88	0.86
	16k	0.96	0.92	1.00	0.86	0.78
MeetingBank	4k	0.96	0.80	0.98	0.76	0.80
	16k	0.92	1.00	1.00	0.84	0.76
PubMed	4k	0.96	0.90	1.00	0.80	0.76
	16k	1.00	0.94	0.98	0.96	0.84

Table 12: GPT-4o evaluation accuracy of synthetic claims under 4k vs 16k unstructured settings, reported per dataset and error type.

Prompt	Content
Atomic Fact Extraction (Split Reasoning)	## **Task Description** You will be given an explanation statement. Your task is to extract a set of **atomic facts**—statements that can be **directly inferred** from this explanation without interpretation, additional assumptions, or redundancy. ## **Guidelines:** - Extract only **explicitly stated** atomic facts in the explanations. - **Do not repeat** facts or include any that require external knowledge. - Maintain **granularity**: Each fact should be **minimal yet complete**. - Structure your output as a valid list of facts, **one fact per line**. **Do not include any additional text or formatting.** - Each summary has at least 1 atomic fact. – ## **Example Output Format** "First atomic fact" "Second atomic fact" "Third atomic fact" – ## **Input** **Explanation:** {explanation} – ## **Output** (**List Only**)

Table 13: Split-reasoning prompt for extracting atomic facts from an explanation. Replace {explanation} with the input text.

Prompt	Content
Atomic Fact Support Evaluation (yes/no)	## **Task Description** You are given an **atomic fact** and a **context**. Your task is to determine whether the fact is **fully supported** by the context. ## **Guidelines:** - A fact is **supported** only if all of its information is explicitly confirmed by the context. - If any part of the fact is missing, contradicted, or not stated in the context, mark it as **not supported**. - Respond with only: **yes** (fully supported) or **no** (not fully supported). – ## **Input** **Context:** {context} **Fact:** {fact} – ## **Output (yes/no only)**

Table 14: Prompt for evaluating whether an atomic fact is fully supported by a given context. Replace {context} and {fact} with inputs.

Prompt	Content
Reasoning Ranking	The true label is '{label}'. Both explanations aim to justify this decision. 1) {r1} 2) {r2} Which explanation is more detailed and supportive? Reply with '1', '2', or 'equal'.

Table 15: Prompt for pairwise reasoning ranking: given a gold label and two competing explanations, the judge selects which explanation is more detailed and supportive.

Model / Setting	Generated Reasoning (Label = No)
Base Model	The claim is inconsistent with the document.
Unstructured	The claim is inconsistent with the document. The document states that OSG is exploring opportunities in transporting liquified CO_2.
Context-Graph	The claim is inconsistent with the document. The document states that OSG is exploring opportunities in transporting liquified CO_2, which is relevant to maritime innovations and aligns with the company's strategy to participate in emerging markets for energy and liquid bulk commodities. However, the claim incorrectly suggests that all revenue from maritime innovations is allocated to sectors outside the maritime industry, which contradicts the document's statement that OSG is actively pursuing such opportunities.
Argument-Graph	The claim is inconsistent with the document. The document states that OSG is exploring opportunities in transporting liquified CO_2, which is a natural next step into an emerging market, consistent with OSG's expertise with liquid cargoes. However, it does not mention that all revenue is allocated to sectors outside maritime innovations. The document highlights that OSG is seeking opportunities to increase scale and drive cost efficiencies through a disciplined approach to investment in core and adjacent asset classes to maximize return on capital across market cycles. This includes the recent acquisition of the Alaskan Frontier and the planned life cycle upgrades on the four Alaska class vessels, which are expected to improve performance and fuel efficiency, leading to a meaningful reduction in carbon output. Additionally, the document mentions that OSG is committed to acting in the best interests of its stockholders and is carefully considering Saltchuk Resources' indication of interest to acquire all outstanding shares of OSG's common stock that Saltchuk does not own for $6.25 per share in cash. The document does not provide any information about revenue allocation outside the maritime sector.

Table 16: Comparison of model-generated explanations under different synthesis strategies (label = No). Structured synthesis produces longer, more detailed rationales that incorporate specific evidence.