Breaking the Efficiency Barrier: A Fast and Scalable Factuality Evaluation Framework for LLMs

Anonymous ACL submission

Abstract

Large language models (LLMs) exhibit remarkable text-generation capabilities yet struggle with factual consistency in knowledge-intensive tasks. Existing fact-checking methods based on the "Decompose-Then-Verify" paradigm improve factual reliability but face scalability issues due to two main limitations: (1) reliance on costly LLM API calls, and (2) quadratic complexity from pairwise verification of decomposed text segments. We present Light-FS, an efficient framework adopting a "Decompose-Embed-Interact" paradigm: (1) a small language model (SLM) based decomposer extracts atomic propositions, (2) a specialized Bi-Encoder module generates semantic embeddings, and (3) a multi-feature interaction module performs embedding-based verification. Our experiments show that Light-FS achieves $14\times$ faster decomposition than GPT-4o within a 3% F1-drop while delivering a $20\times$ efficiency gain over NLI-based fact-checking models with comparable verification performance. Light-FS provides a scalable and efficient solution for evaluating the factuality of LLM-generated content.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in text generation tasks (Mann et al., 2020; Li et al., 2024; Iqbal et al., 2024). However, ensuring the factual reliability of the generated content remains a critical challenge. Recent studies (Ji et al., 2023; Bang et al., 2023; Sadasivan et al., 2023) indicate that LLMs frequently generate hallucinated content, including incorrect dates, numerical errors, and fabricated relationships, which can mislead decision-making and exacerbate misinformation spread. Consequently, automated factuality verification for LLM-generated content has become a critical research problem in NLP (Panchendrarajan and Zubiaga, 2024; Si et al., 2024; Atanasova, 2024).

Existing fact-checking methods predominantly adopt the "Decompose-Then-Verify" paradigm, where generated text is decomposed into atomic factual claims and verified against a reference source (Zhang and Bansal, 2021; Chern et al., 2023; Zhao et al., 2023). FactScore (Min et al., 2023), a representative approach, employs LLMs for atomic fact decomposition and then verifies each fact using either a LLM or a Natural Language Inference (NLI) model. While this paradigm enhances verification granularity, its reliance on costly API calls and quadratic complexity in pairwise fact verification makes it impractical for large-scale applications.

To address these efficiency bottlenecks, we present Light-FS (Light-FactScore), an API-free and computationally efficient fact-checking framework that implements a novel "Decompose-Embed-Interaction" paradigm. Light-FS achieves efficiency-accuracy balance through three key innovations:

First, we adopt a sentence-level decomposition strategy using a supervised fine-tuned small language model (SLM). Compared to conventional paragraph-level LLM decomposition (Min et al., 2023), this strategy reduces inference latency by $15\times$ while mitigating long-context hallucination risks. Second, we introduce a specialized Bi-Encoder architecture that improves the representation quality of atomic fact embeddings. Unlike NLI models, which require premise-hypothesis pairs for verification, our approach encodes the premise and hypothesis independently, eliminating the need for pairwise comparisons. This architecture reduces computational complexity from $O(K^2)$ to $O(K)$, achieving $20\times$ speedup over conventional NLI verification. Third, we design a multi-feature interaction module that strengthens embedding interactions. By integrating pairwise interaction features, discrepancy features, and global similarity features, this module enables embedding-based verification

to achieve accuracy comparable to NLI models while maintaining computational efficiency.

Our contributions can be summarized as:

- We introduce Light-FS, a novel computationally efficient fact-checking framework that resolves quadratic complexity bottlenecks through our "Decompose-Embed-Interaction" paradigm.
- We propose a sentence-level atomic fact decomposition strategy using a SLM, achieving 15× speedup over LLM-based decomposition while maintaining minimal F1 performance degradation.
- We design an efficient fact verification mechanism composed of a specialized Bi-Encoder and a multi-feature interaction module, achieving NLI-level verification performance while improving computational efficiency by 20×.

2 Related Works

2.1 Hallucinations in LLMs

Hallucinations in LLMs, where models generate non-factual content such as temporal inconsistencies, numerical errors, or fabricated relationships, present significant challenges to their reliability, particularly in knowledge-intensive tasks (Huang et al., 2023). Current strategies to mitigate hallucinations include training-phase interventions (e.g., curated datasets and knowledge distillation) (Gekhman et al., 2024; Abbas et al., 2023; McDonald et al., 2024; Huang et al., 2022), retrieval-augmented generation (RAG) approaches that integrate external knowledge during inference (Ram et al., 2023; Gao et al., 2022; Lewis et al., 2020), and post-hoc verification methods to assess factual consistency after text generation (Manakul et al., 2023; Dhuliawala et al., 2023; Maynez et al., 2020). The development of standardized evaluation benchmarks, like TruthfulQA (Lin et al., 2021), REALTIMEQA (Kasai et al., 2024) and HaluEval (Li et al., 2023), has further enabled systematic measurement of hallucination patterns across different models.

2.2 Factuality Evaluation

Fact verification methods are primarily categorized into Factual Hallucination Detection and Faithfulness Hallucination Detection (Huang et al., 2023). Both methods fundamentally rely on comparing the generated content with reference material. However, direct document-level comparisons often fail to pinpoint specific factual inconsistencies when applied to long-text scenarios. The "Decompose-Then-Verify" paradigm, as demonstrated by FactScore (Min et al., 2023), overcomes this limitation by breaking the text into atomic factual claims for more granular verification. While this approach improves precision and interpretability, it introduces significant computational challenges, particularly due to the reliance on iterative LLM API calls for atomic fact decomposition and the quadratic complexity of pairwise verification. Even when replacing LLMs with smaller NLI models like DeBERTa (He et al., 2020), these bottlenecks persist, making the "Decompose-Then-Verify" methodology impractical for large-scale evaluations.

3 Light-FS

Light-FS follows a three-stage "Decompose-Embed-Interact" paradigm, consisting of three core components: Decomposer, Embedder, and Multi-Feature Interaction Module (MFIM). Figure 1 illustrates the overall architecture of Light-FS. The workflow of Light-FS consists of three stages: (1) **Atomic Fact Decomposition**. The Decomposer extracts atomic facts from both generated content and reference material at the sentence level, ensuring each fact is independent, self-contained, and semantically complete. (2) **Embedding Generation**. The Embedder, based on a Bi-Encoder architecture, converts atomic facts into vector representations. (3) **Fact Verification via MFIM**. The MFIM computes the fact score between generated content and reference material based on multi-feature interaction.

The following sections detail the three core components of Light-FS, and its detailed implementation is provided in Appendix A.

3.1 Decomposer

The Decomposer extracts discrete and self-contained atomic facts from the textual content. This decomposition process requires strong reasoning capabilities, typically best handled by LLMs. We employ a supervised fine-tuned SLM to balance efficiency and reasoning capability, significantly reducing computational costs. However, when applied to long-text decomposition, SLMs may gener-

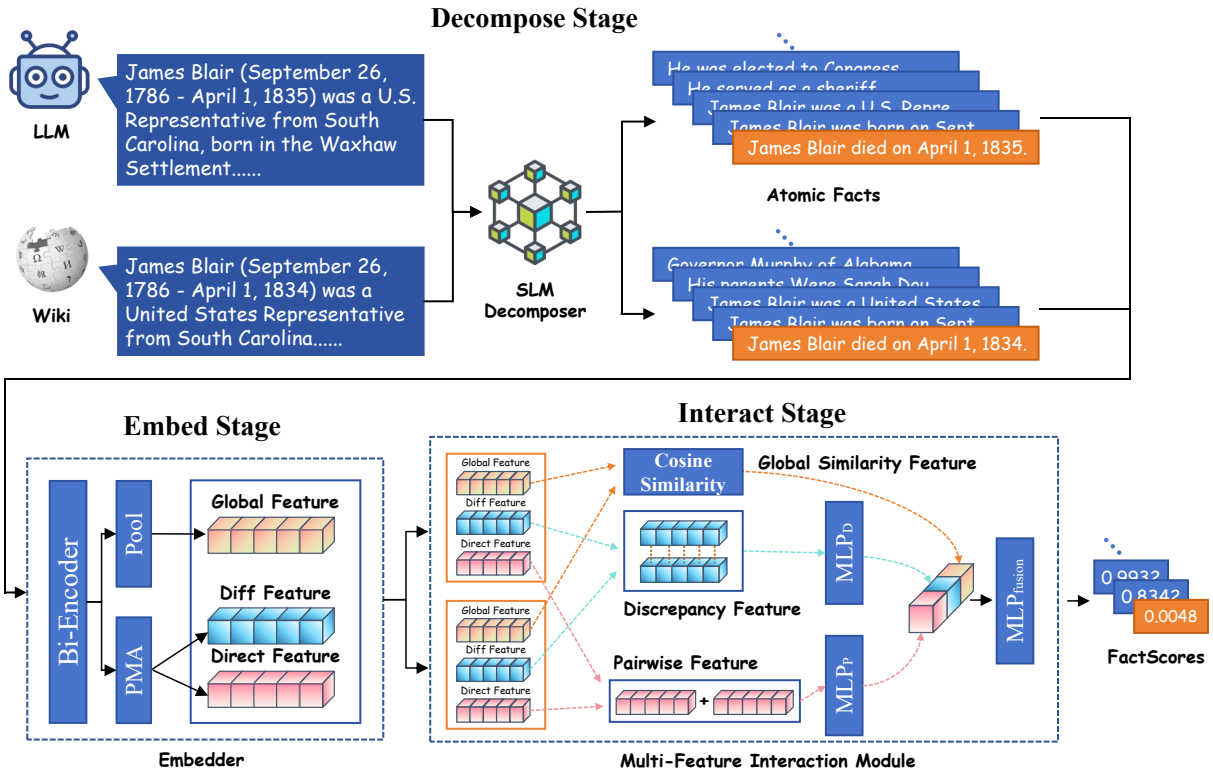

Figure 1: Overview of the **Light-FS** framework for fact verification. The system follows a three-stage process: **Decompose**, **Embed**, and **Interact**. In the **Decompose** stage, the LLM-generated text and the corresponding reference text from Wikipedia are processed using a small language model decomposer. In the **Embed** stage, these atomic facts are encoded using a Bi-Encoder, with the use of PMA and Pool to capture different embedding features. In the **Interact** stage, the embeddings undergo multi-feature interactions through cosine similarity and feature-based processing, producing fact scores to assess the factuality of the content.

ate hallucinated, inaccurate, or incomplete atomic facts, which can compromise the accuracy of subsequent verification processes. To mitigate this, we adopt a sentence-level decomposition strategy instead of a passage-level to minimize factual distortions. Additionally, embedding-based approaches alone may struggle to capture fine-grained semantic nuances in long-text scenarios. We apply atomic fact decomposition to both generated content and reference material, ensuring greater scalability and improved fact verification.

We first segment the input text (both generated content and reference material) into sentences using Stanza (Qi et al., 2020), denoted as $T = \{t_1, t_2, ..., t_n\}$, where each t_i represents the i-th sentence. Each sentence t_i is individually processed by the SLM to extract atomic facts, resulting in a fact set $A_i = \{a_1, a_2, ..., a_m\}$, where a_j represents the j-th atomic fact from sentence t_i. The complete atomic fact set is constructed as $A = \bigcup_{i=1}^{n} A_i$.

By adopting sentence-level decomposition, we reduce the factual complexity per inference step, minimize hallucination risks, and enhance fact decomposition accuracy.

3.2 Embedder

The Embedder converts atomic facts into vector representations for efficient fact verification. Traditional BERT-based embedding models typically use either the [CLS] token or mean pooling for sentence embeddings (Reimers, 2019). However, such methods often fail to capture fine-grained semantic nuances, critical for factuality evaluation. We adopt a Pooling-based Multi-Head Attention (PMA) mechanism to enhance embedding quality, inspired by (Liao et al., 2024; Lee et al., 2019), following the BERT encoder.

Given an atomic fact set $F = \{s_1, s_2, ..., s_n\}$, each fact s_i is tokenized and encoded using BERT, resulting in token embeddings $T_i = \{t_1, t_2, ..., t_{\text{len}}\}$, where t_j is a d-dimensional vector. The PMA module then aggregates token embeddings to produce a multi-view sentence embedding:

$$h = \text{LN}(\text{MHA}(q, T_i, T_i) + q), h_i^{\text{agg}} = \text{LN}(h + \text{FFN}(h))$$

Where LN denotes Layer Normalization, $\text{MHA}(Q, K, V)$ is the Multi-Head Attention mechanism, q is a learnable query vector, dynamically aggregating token-level information. To capture diverse aspects of sentence semantics, h_i^{agg} consists of two embeddings, each representing different aspects of sentence meaning. This design retains richer contextual information than traditional pooling-based methods.

Moreover, to leverage BERT's global semantic representation, we extract a global embedding g_i from either the [CLS] token $t_i[\text{CLS}]$ or the mean-pooled token embeddings $mean(T_i)$, depending on the model's training configuration.

The final sentence representation H_i is formed by stacking both the attention-based and global embeddings:

$$H_i = \text{Stack}(h_i^{\text{agg}}, g_i) \in \mathbb{R}^{3\times d}$$

This multi-view embedding strategy enriches atomic fact representations, providing stronger factual verification signals.

3.3 MFIM

The Multi-Feature Interaction Module (MFIM) computes a fact score between reference and generated atomic fact embeddings. Given embeddings H_r (reference material) and H_g (generated content), we define three interaction features:

Pairwise Interaction Feature (P): P models the direct semantic alignment between reference and generated facts, explicitly capturing their factual overlap. This feature helps detect minor factual distortions, such as incorrect dates, numerical discrepancies, or entity mismatches, by comparing their semantic representations. As the primary factual alignment signal, P enables the model to detect cases where the generated fact is directly entailed by or contradicts the reference fact.

$$P = \text{MLP}_P(\text{Concat}(H_r[0], H_g[0])) \in \mathbb{R}$$

Discrepancy Feature (D): D models fine-grained factual differences, simulating premise-hypothesis entailment in NLI tasks. Errors in generated content sometimes arise from introducing extraneous information rather than direct contradiction. To quantify this, D computes the directional difference between the reference fact and the generated fact, detecting cases where the generated content includes unsupported details that alter factual accuracy. Unlike direct contradiction detection, this feature ensures the model penalizes factual additions while allowing omissions as long as the retained information remains correct.

$$D = \text{MLP}_D(H_r[1] - H_g[1]) \in \mathbb{R}$$

Global Similarity Feature (S): S quantifies overall semantic alignment between reference and generated embeddings using cosine similarity. While pairwise and discrepancy features focus on local fact-level alignment, factual consistency also depends on global semantic coherence. Cosine similarity provides a robust measure of overall contextual consistency, ensuring that the generated content is lexically and semantically aligned with the reference material.

$$S = \frac{H_r[2] \cdot H_g[2]}{||H_r[2]|| \cdot ||H_g[2]||}$$

The final fact score is computed via a fusion network. This fusion mechanism enables the model to jointly leverage direct semantic alignment, information asymmetry, and global contextual consistency, ensuring a more comprehensive factuality assessment.

$$FactScore = \text{Sigmoid}(\text{MLP}_{\text{fusion}}(P, D, S))$$

3.4 Computational Complexity Analysis

In this section, we theoretically analyze the computational efficiency of the Light-FS framework. We divide the analysis into two main components: Decomposer (responsible for atomic fact extraction) and Checker (responsible for embedding and fact verification).

3.4.1 Decomposer Complexity Analysis

Light-FS utilizes a supervised fine-tuned SLM to perform atomic fact decomposition at the sentence level. Sentence segmentation is computationally lightweight, and its cost can be ignored. In contrast, fact decomposition is the primary computational bottleneck, as each sentence must be processed by the decomposer.

Given that the input sequence of T tokens partitioned into N sentences, language models employing self-attention mechanisms (Vaswani, 2017) incur quadratic computational complexity $O(T^2)$. Light-FS addresses this challenge through sentence-level decomposition. By constraining attention computations to individual sentences with average length $\bar{t} = \frac{T}{N} \ll T$, the aggregated complexity reduces to $O(N\bar{t}^2)$. This design drastically reduces

global attention costs by restricting attention computations to shorter text segments, making Light-FS substantially more efficient than conventional passage-level LLM processing.

3.4.2 Checker Complexity Analysis

The Checker module consists of the Embedder and the MFIM. Its complexity is influenced by the following factors: embedding computation and fact verification computation. To facilitate analysis, we assume both the generated content and the reference material contain K atomic facts, and the reference content is segmented into S chunks, where $\bar{N}$ denotes the average number of atomic facts per chunk.

Embedding Computation. Light-FS employs a Bi-Encoder structure, enabling independent encoding of atomic facts before interaction. Assuming that the computational complexity of the BERT-based embedding model is $O(D)$, the embedding process involves encoding $2K$ atomic facts (from both the generated and reference content), resulting in a total embedding complexity of $O(2KD)$.

Fact Verification Computation. The MFIM performs pairwise interaction between atomic fact embeddings. Unlike NLI models, which require cross-encoding each premise-hypothesis pair, Light-FS utilizes a more efficient MLP-based comparison. Given that each generated atomic fact is compared with all K reference atomic facts, the verification complexity is $O(K^2M)$, where M represents the computational complexity of MLP.

Thus, the overall complexity of the Checker is $O(2KD + K^2M)$. For a standard NLI-based model, each atomic fact in the generated content is compared against S chunks of the reference content. Assuming the NLI model has a $O(D)$ complexity per comparison, the total complexity can be expressed as $O(KSD)$. Rewriting S in terms of $\bar{N}$ (the average number of atomic facts per chunk), we obtain $O(\frac{K^2}{\bar{N}}D)$.

The above analysis highlights a key difference: while NLI models require quadratic complexity in D (transformer-based cross-encoding), Light-FS shifts the quadratic term to M, which corresponds to the MLP computation. Since MLPs are significantly more efficient than transformer-based models, Light-FS substantially reduces computational overhead.

4 Experiments

To systematically evaluate the effectiveness of the Light-FS framework, we conduct experiments in four key dimensions: (1) **Decomposition Capability Evaluation**: Compare different models in atomic fact decomposition to identify the most suitable decomposer. (2) **Fact Verification Performance Assessment**: Assess the effectiveness of Checker (consisting of Embedder and MFIM) against traditional NLI models. Then, assess the overall performance of Light-FS, incorporating both the decomposer and checker. (3) **Computational Efficiency Analysis**: Measure inference speed in decomposition and fact verification. (4) **Ablation Study**: Analyze the impact of core components, including Pooling-based Multi-Head Attention and Multi-Feature Interaction Module.

All open-source LLMs used in the experiments are Q4_K_M quantized, executed with llama.cpp[1]. For long-context fact verification, LLM-based approaches are provided with the full reference content as the premise input. Cross-Encoder models receive premise inputs in chunks (500-character length with 100-character overlap). Bi-Encoder-based approaches, including ours, are fed atomic facts as premise inputs. In all cases, the hypothesis input consists of atomic facts.

4.1 Datasets

wiki-en-sentences: A large-scale factuality detection dataset constructed from 500,000 Wikipedia sentences selected from wikipedia-en-sentences[2]. We prompt a LLM to generate both positive and negative samples. Our Light-FS is trained on it.

wiki-bio-hallucination (Manakul et al., 2023): A dataset for evaluating hallucinations in LLM-generated biographies, containing 238 Wikipedia biography articles. We expanded this dataset with both synthetic and real data to enhance its applicability in factuality verification. Synthetic data consists of controlled factual hallucinations generated by GPT-4o (Hurst et al., 2024), while real data includes biographies produced by four closed-source models, GPT-3.5-Turbo, GPT-4o, Claude-3.5-Haiku, Claude-3.5-Sonnet, and four open-source models, Llama-2-7b, Llama-2-13b (Touvron et al., 2023), Qwen2-7B (Bai et al., 2023), Qwen2.5-0.5B (Yang et al., 2024).

[1] https://github.com/ggerganov/llama.cpp

[2] https://huggingface.co/datasets/sentence-transformers/wikipedia-en-sentences

factscore-dataset (Min et al., 2023): A subset of the FactScore (Min et al., 2023) dataset focusing on ChatGPT-generated Wikipedia biographies. We select 105 samples that have matching reference content in the wiki_bio dataset [3](Lebret et al., 2016) to benchmark factuality verification models.

Due to the high cost of manual annotation, both wiki-bio-hallucination and factscore-dataset were annotated using GPT-4o for atomic fact decomposition and factuality labeling to ensure consistency. The prompts used for generation and annotation are provided in the appendix A.3.

4.2 Decomposition Capability Evaluation

We evaluate the decomposition performance of GPT-4o with several open-source models, Qwen2-7B, Qwen2.5-0.5B and Flan-T5 (Chung et al., 2022), on the wiki-bio-hallucination dataset. The evaluation metrics include **Precision** (correctly extracted facts), **Recall** (alignment with GPT-4o's decomposition), and **F1** score. GPT-4o, as the standard reference, performs passage-level decomposition using few-shot prompting, while open-source models undergo supervised fine-tuning and are evaluated under the same conditions. A Qwen2-7B model serves as the evaluator.

Table 1: Performance comparison of different decomposers at various decomposition granularities.

Model	Granularity	F1	Precision	Recall
GPT-4o	Passage	0.9910	0.9830	0.9991
Qwen2-7B	Sentence	0.9797	0.9799	0.9795
Qwen2-7B	Passage	0.9703	0.9875	0.9536
Qwen2.5-0.5B	Sentence	0.9676	0.9628	0.9725
Flan-T5	Sentence	0.9486	0.9512	0.9460
Qwen2.5-0.5B	Passage	0.8837	0.8920	0.8754

As shown in Table 1, GPT-4o demonstrates strong performance in passage-level decomposition. Sentence-level decomposition generally yields higher recall than passage-level decomposition across models. Among open-source models, Qwen2-7B performs strongly at the sentence level, but its passage-level recall declines, suggesting long-text decomposition may introduce factual inconsistencies, especially in smaller models. Qwen2.5-0.5B performs comparably to Qwen2-7B at the sentence level, while Flan-T5 lags slightly. Considering both accuracy and computational efficiency, Qwen2.5-0.5B (Sentence) is selected as the Decomposer for Light-FS, as it achieves a strong balance between decomposition quality and inference speed, making it a practical choice for large-scale fact verification.

4.3 Fact Verification Performance Assessment

In this section, we validate the fact verification ability of Light-FS through two experiments. The first experiment focuses on assessing the effectiveness of the Checker. The second experiment evaluates the full Light-FS framework, incorporating both the Decomposer and Checker.

4.3.1 Experiment on Checker

To evaluate the performance of the Checker, we conducted comparison experiments with several baselines, including Qwen2-7B, two NLI models[4], DeBERTa-v3-base-mnli-fever-anli[5] and nli-deberta-v3-base[6] and two Bi-Encoder models, BERTScore (Zhang et al., 2019) and BGE-en-base-v1.5 (Xiao et al., 2023). The experiment is conducted across four datasets, with evaluation metrics including **Accuracy**, **F1**, **Recall**, and **Precision**.

As shown in Table 2, Light-FS consistently outperforms other non-LLM models across most datasets, excelling in both accuracy and F1 score. Due to targeted training, Light-FS achieves the highest performance on the wiki-en-sentences dataset, demonstrating superior fine-grained fact verification. On the wiki-bio-hallucination (synthesis) dataset, Qwen2-7B leads in accuracy, but Light-FS outperforms NLI models by capturing subtle differences in the generated text. Bi-Encoder models show high recall but lack precision, indicating limitations in handling fine-grained factual discrepancies. For the more challenging wiki-bio-hallucination (GPT-4o), all models show a performance drop due to the diverse and summary-based nature of the content. Despite this, Light-FS maintains competitiveness, outperforming DeBERTa-v3-base-mnli-fever-anli in both precision and recall. On the factscore-dataset, Light-FS achieves high accuracy and recall, surpassing Bi-Encoder baselines and matching or exceeding NLI models. Even our mini model (23M parameters), much smaller than the base model (109M parameters), shows

[3] https://huggingface.co/datasets/michaelauli/wiki_bio

[4] We followed (Jiang et al., 2024) setup by ignoring the neutral label and using only entailment and contradiction as the basis for fact verification.

[5] https://huggingface.co/MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli

[6] https://huggingface.co/cross-encoder/nli-deberta-v3-base

Table 2: Performance comparison of various models across different datasets, where wiki-bio-hallucination (synthesis) consists of LLM-generated biographies with controlled factual hallucinations, while wiki-bio-hallucination (GPT-4o) contains real-world hallucinations from GPT-4o-generated biographies. The table presents **accuracy (Acc)**, **F1 score**, **recall (Recall)**, and **precision (Prec)** for different models, including random, LLM-based models, Cross-Encoders, Bi-Encoders, and our proposed method. The best results are marked in **bold**, and the second-best results are underlined. L stands for LLM, X stands for Cross-Encoder, and B stands for Bi-Encoder.

Types	Models	wiki-en-sentences				wiki-bio-hallucination (synthesis)				wiki-bio-hallucination (GPT-4o)				factscore-dataset			
		Acc	F1	Recall	Prec	Acc	F1	Recall	Prec	Acc	F1	Recall	Prec	Acc	F1	Recall	Prec
	Random	0.5023	0.5062	0.5002	0.5123	0.5058	0.5753	0.5135	0.6541	0.5028	0.5166	0.4962	0.5388	0.5036	0.4005	0.4834	0.3418
L	Qwen2-7B	0.8891	0.9014	0.9933	0.8250	0.9239	0.9442	0.9875	0.9046	0.8297	0.8581	0.9615	0.7747	0.8058	0.7745	0.9724	0.6435
X	DeBERTa-v3-base-mnli-fever-anli	0.8249	0.8535	**1.0000**	0.7444	0.7770	0.8528	0.9914	0.7483	0.5519	0.7038	0.9943	0.5447	0.6367	0.6441	0.9586	0.4850
	nli-deberta-v3-base	0.8556	0.8758	0.9983	0.7801	0.8199	0.8718	0.9394	0.8134	0.6118	**0.7283**	0.9719	0.5824	**0.7252**	**0.6861**	0.8757	**0.5641**
B	BERTScore	0.5100	0.6755	**1.0000**	0.5100	0.6519	0.7893	**1.0000**	0.6519	0.5354	0.6974	**1.0000**	0.5354	0.3430	0.5108	**1.0000**	0.3430
	BGE-en-base-v1.5	0.5753	0.7060	0.9996	0.5457	0.6519	0.7893	**1.0000**	0.6519	0.5383	0.6988	**1.0000**	0.5370	0.3482	0.5127	**1.0000**	0.3448
	Ours	**0.9444**	**0.9476**	0.9841	**0.9136**	**0.8638**	**0.8965**	0.9051	**0.8881**	**0.6423**	0.7043	0.7954	**0.6319**	0.6949	0.6434	0.8025	0.5370
	Ours (mini)	0.9054	0.9132	0.9761	0.8580	0.7920	0.8447	0.8675	0.8230	0.6146	0.6889	0.7971	0.6066	0.6551	0.6099	0.7859	0.4982

competitive results across all datasets, demonstrating that a significant parameter reduction does not come at the large sacrifice of performance.

4.3.2 Experiment on Overall Framework

Table 3: Performance of different Decomposers and Checkers on the wiki-bio-hallucination dataset. **Spearman/Pearson Correlation Coefficient** and **Coefficient of Variation** are used as metrics. For detailed fact score distributions, refer to the Figures 2 in the appendix.

Decomposer	Checker	Spearman	Pearson	CV
GPT-4o	Qwen2-7B	**0.9762**	**0.9972**	**0.4679**
	DeBERTa-v3-base-mnli-fever-anli	0.9286	0.9776	0.1590
	nli-deberta-v3-base	**0.9762**	0.9901	0.2846
	BGE-en-base-v1.5	0.3095	0.4608	0.0063
	Ours	0.9524	0.9749	0.2151
Ours	Qwen2-7B	**0.9762**	0.9910	0.4242
	Ours	**0.9762**	0.9581	0.2378
	Ours(mini)	**0.9762**	0.9523	0.2235

In this section, we evaluate the performance of the full Light-FS framework to assess its reliability in factual verification. The experiment is conducted on the wiki-bio-hallucination dataset, which contains generated content from 8 mainstream closed-source and open-source models. We compute the average fact score for the content generated by each model using different combinations of decomposers and checkers. These scores are then compared against GPT-4o's ground truth annotations, with **Spearman** and **Pearson** correlation coefficients used to assess alignment and the **Coefficient of Variation** to measure discriminative power.

As shown in Table 3, our full Light-FS framework maintains a high correlation with GPT-4o, demonstrating its reliability as an independent fact-checking system. Compared to NLI models, Light-FS achieves similar Spearman and Pearson correlations but exhibits a higher CV than DeBERTa-v3-base-mnli-fever-anli, indicating better differentiation capability in assessing factual inconsistencies. In contrast, Bi-Encoder models struggle with fine-grained fact distinctions, leading to low correlation and poor score variability. Importantly, when replacing the GPT-4o decomposer with ours, the performance of Qwen2-7B checker consistency remains high, further validating our Decomposer's effectiveness. The joint use of our Decomposer and Checker ensures stable and robust performance, maintaining high accuracy, efficient computation, and strong differentiation in factual verification tasks. Notably, our mini model retains competitive performance despite its significantly smaller size.

4.4 Computational Efficiency Analysis

In this section, we evaluate the efficiency of both the Decomposer and Checker in Light-FS. The experiment is conducted on the wiki-bio-hallucination dataset, which contains generated content from 8 mainstream closed-source and open-source models.

As shown in Table 4, Qwen2.5-0.5B(Ours) demonstrates an impressive efficiency advantage. The total decomposition time consists of two components: **Shared Decomposition Time**, which is required by all methods to process the generated content, and **Additional Decomposition Time**,

Table 4: Decomposition time. **Shared Decomposition Time** refers to the time to decompose the generated content. **Additional Decomposition Time** refers to the time spent on decomposing the reference text.

Model	Shared Decomposition Time(seconds)	Additional Decomposition Time(seconds)
GPT-4o (API)	10285.64	1215.22
Qwen2-7B	15950.54	1839.49
Qwen2.5-0.5B (Ours)	**665.29**	**148.71**

Table 5: Fact verification time. **Embedding Time** refers to the time spent on embedding the atomic facts. **Computation Time** refers to the time spent on fact verification. **Total Time** is the sum of embedding and computation time for the complete verification process.

Model	Embedding Time (seconds)	Computation Time (seconds)	Total Time (seconds)
GPT-4o (API)	-	56680.48	56680.48
Qwen2-7B	-	3401.82	3401.82
nli-deberta-v3-base	-	346.76	346.76
Ours	17.33	0.33	17.66
Ours(mini)	3.01	0.32	**3.33**

which accounts for the extra time needed to decompose the reference text (i.e., Wikipedia) and is specific to our method. Our model achieves a 14× speedup over GPT-4o and a 22× speedup over Qwen2-7B in total decomposition time. Additionally, the additional decomposition time only constitutes 10-20% of the total decomposition time, and this proportion further decreases as the volume of generated content increases. This efficiency gain is primarily attributed to sentence-level decomposition strategy and the use of a smaller model, which significantly reduces computational overhead. Importantly, this improvement does not come at the cost of quality, as the F1 score remains within 2.34% of GPT-4o.

As shown in Table 5, our Checker shows a remarkable advantage in computation time in the fact verification phase. Compared to the pairwise inference of NLI models, our base version completes the task in only 17.66 seconds (a 20x speedup), and the mini version achieves 3.33 seconds (a 104x speedup). This efficiency is largely due to our Bi-Encoder architecture, which enables individual processing of embeddings for generated and reference facts, minimizing redundant computation. In contrast, Cross-Encoder models perform inferences for each fact pair, leading to significantly higher computational complexity.

4.5 Ablation Studies

Since decomposition is an integral part of our approach, this ablation study focuses solely on the Checker module, specifically the effects of the PMA and the MFIM.

Table 6: Ablation study results comparing different configurations for fact verification across two datasets.

	wiki-en-sentences		wiki-bio-hallucination (synthesis)	
	Acc	F1	Acc	F1
Light-FS	**0.9444**	**0.9476**	**0.8638**	**0.8965**
-PMA+Pool	0.8835	0.8962	0.7791	0.8499
-MFIM+Cosine	0.8111	0.8269	0.7482	0.8353
-PMA-MFIM +Pool+Cosine	0.5753	0.7060	0.6519	0.7893

As shown in Table 6, replacing PMA with global pooling methods resulted in a significant drop in accuracy and F1 score, highlighting the importance of PMA in capturing fine-grained semantic differences. Replacing the MFIM with cosine similarity caused a notable decline in performance, particularly in precision, emphasizing the necessity of multi-feature interaction for effective fact verification. Finally, removing both PMA and MFIM led to a dramatic performance drop, confirming the essential role of these components in ensuring robust fact verification.

5 Conclusion

This paper proposes Light-FS, an API-free, computationally efficient framework for evaluating the factuality of generated content. By adopting a "Decompose-Embed-Interact" three-stage paradigm, Light-FS significantly improves computational efficiency while maintaining high verification accuracy. Specifically, we replace the traditional passage-level LLM processing with a sentence-level SLM decomposition strategy, achieving a 14x speedup in atomic fact decomposition. Additionally, Light-FS integrates a Bi-Encoder architecture with a multi-feature interaction mechanism, enhancing efficiency and achieving a 20x acceleration over conventional NLI models. Ablation studies further confirm the importance of the Pooling-based Multi-head Attention and Multi-Feature Interaction modules in improving fact verification performance. In the future, we plan to optimize Light-FS further to improve its generality and scalability.

Limitations

Despite the excellent performance of the Light-FS framework in fact verification tasks, the following limitations remain:

Domain Adaptability: Light-FS was trained on Wikipedia data, which may limit its applicability to other domains like news, law, or scientific papers. Adapting to different data distributions may require additional fine-tuning or training.

Inference Limitations: Light-FS uses a Bi-Encoder-based approach that is suitable for surface-level fact matching. It may struggle with complex reasoning tasks, such as causal or temporal relationships, where Cross-Encoder models perform better due to their interactive encoding.

Dependency on Reference Quality: The effectiveness of Light-FS depends on the accuracy and authority of the reference texts. Light-FS may misjudge generated content if the reference material is outdated or erroneous.

Extra Computational Cost: The need for decomposing both reference and generated texts increases computational costs, particularly in large-scale or real-time verification scenarios. Optimizing caching and retrieval mechanisms could address this issue.

Acknowledgments

This work was supported in part by XXX.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *arXiv preprint arXiv:2303.09540*.

Pepa Atanasova. 2024. Generating fact checking explanations. In *Accountable and Explainable Methods for Complex Reasoning over Text*, pages 83–103. Springer.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv preprint*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *ArXiv*, abs/2309.11495.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, et al. 2022. Rarr: Researching and revising what language models say, using language models. *arXiv preprint arXiv:2210.08726*.

Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. Does fine-tuning llms on new knowledge encourage hallucinations? *arXiv preprint arXiv:2405.05904*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*.

Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. 2022. Knowledge distillation from a stronger teacher. *Advances in Neural Information Processing Systems*, 35:33716–33727.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Hasan Iqbal, Yuxia Wang, Minghan Wang, Georgi Georgiev, Jiahui Geng, Iryna Gurevych, and Preslav Nakov. 2024. Openfactcheck: A unified framework for factuality evaluation of llms. *arXiv preprint arXiv:2408.11832*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea

Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Zhengping Jiang, Jingyu Zhang, Nathaniel Weir, Seth Ebner, Miriam Wanner, Kate Sanders, Daniel Khashabi, Anqi Liu, and Benjamin Van Durme. 2024. Core: Robust factual precision scoring with informative sub-claim identification. *arXiv e-prints*, pages arXiv–2407.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. 2024. Realtime qa: what's the answer right now? *Advances in Neural Information Processing Systems*, 36.

Rémi Lebret, David Grangier, and Michael Auli. 2016. Generating text from structured data with application to the biography domain. *CoRR*, abs/1603.07771.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pages 3744–3753. PMLR.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39.

Zihan Liao, Hang Yu, Jianguo Li, Jun Wang, and Wei Zhang. 2024. D2llm: Decomposed and distilled large language models for semantic search. *arXiv preprint arXiv:2406.17262*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Daniel McDonald, Rachael Papadopoulos, and Leslie Benningfield. 2024. Reducing llm hallucination using knowledge distillation: A case study with mistral large and mmlu benchmark. *Authorea Preprints*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Rrubaa Panchendrarajan and Arkaitz Zubiaga. 2024. Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research. *Natural Language Processing Journal*, 7:100066.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Jiasheng Si, Yibo Zhao, Yingjie Zhu, Haiyang Zhu, Wenpeng Lu, and Deyu Zhou. 2024. Checkwhy: Causal fact verification via argument structure. *arXiv preprint arXiv:2408.10918*.

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

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

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*.

Shiyue Zhang and Mohit Bansal. 2021. Finding a balanced degree of automation for summary evaluation. *arXiv preprint arXiv:2109.11503*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yiran Zhao, Jinghan Zhang, I Chern, Siyang Gao, Pengfei Liu, Junxian He, et al. 2023. Felm: Benchmarking factuality evaluation of large language models. *Advances in Neural Information Processing Systems*, 36:44502–44523.

Appendix

A Implementation Details

Our experiments are conducted on a system running Ubuntu 22.04, equipped with an NVIDIA RTX 4090 GPU, an Intel 13th Gen Core i7-13700K CPU, 64GB RAM, and software dependencies, including CUDA 11.8, PyTorch 2.4.0 and Transformers 4.45.2.

A.1 Decomposer Training Settings

Our Decomposer is based on Qwen/Qwen2.5-0.5B-Instruct[7] as the foundation model. The base model undergoes full-parameter supervised fine-tuning using the llama-factory framework[8], with a learning rate of 2.0e-5, batch size of 4, and trained for 3 epochs.

The training data is sourced from michaelauli/wiki_bio[9], from which 5,000 samples are randomly selected as the foundational dataset. The Wiki paragraphs are first split into sentences using Stanza, and each sentence is then decomposed by GPT-4o to generate the training set. The prompt used for decomposition is as follows using few-shot prompt 3.

A.2 Checker Training Settings

We conduct joint training of the embedder and multi-feature interaction module. The embedder is responsible for generating high-quality sentence embeddings. For this purpose, we utilize the BERT model bge-base-en-v1.5[10]. Additionally, to enhance embedding efficiency, we provide a mini version using the BERT model all-MiniLM-L6-v2[11]. The multi-feature interaction module then computes factuality scores based on these embeddings.

We employ two loss functions: triplet loss and binary cross-entropy (BCE) loss. The objective of Triplet Loss is to optimize fact scores through supervised learning of triplets, ensuring that the factual score of the anchor sentence is higher when paired with a highly factual positive sentence while being lower when paired with an unrelated negative sample.

$$\mathcal{L}_{triplet} = \max(0, \alpha + \text{Factscore}(H_a, H_n) - \text{Factscore}(H_a, H_p)) \quad (1)$$

where α denotes the margin, set to 0.5. H_a, H_p, and H_n represent the embeddings of the anchor, positive, and negative samples, respectively, with factual scores computed via the multi-feature interaction module.

Simultaneously, BCELoss is employed for supervised training. The Factscore output by the multi-feature interaction module is a value between $[0, 1]$, indicating the degree of alignment between the generated content g and the reference content c. The objective is to minimize the difference between the predicted score and the ground truth label $y \in \{0, 1\}$.

$$\mathcal{L}_{\text{bce}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot \log(\text{Factscore}_i) + (1 - y_i) \cdot \log(1 - \text{Factscore}_i)] \quad (2)$$

The overall joint training objective function is formulated as the sum of Triplet Loss and BCE Loss:

$$\mathcal{L} = \mathcal{L}_{\text{triplet}} + \mathcal{L}_{\text{bce}}$$

During training, the parameters of the BERT model are **frozen**, and only the PMA module within the embedder and the multi-feature interaction module are updated. The training process uses a learning rate of 5e-5, a batch size of 32, and runs for 8 epochs.

The dataset is derived from wikipedia-en-sentences[12], selecting the top 500,000 Wikipedia sentences. Triplet data is generated using the

[7] https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct

[8] https://github.com/hiyouga/LLaMA-Factory

[9] https://huggingface.co/datasets/michaelauli/wiki_bio

[10] https://huggingface.co/BAAI/bge-base-en-v1.5

[11] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[12] https://huggingface.co/datasets/sentence-transformers/wikipedia-en-sentences

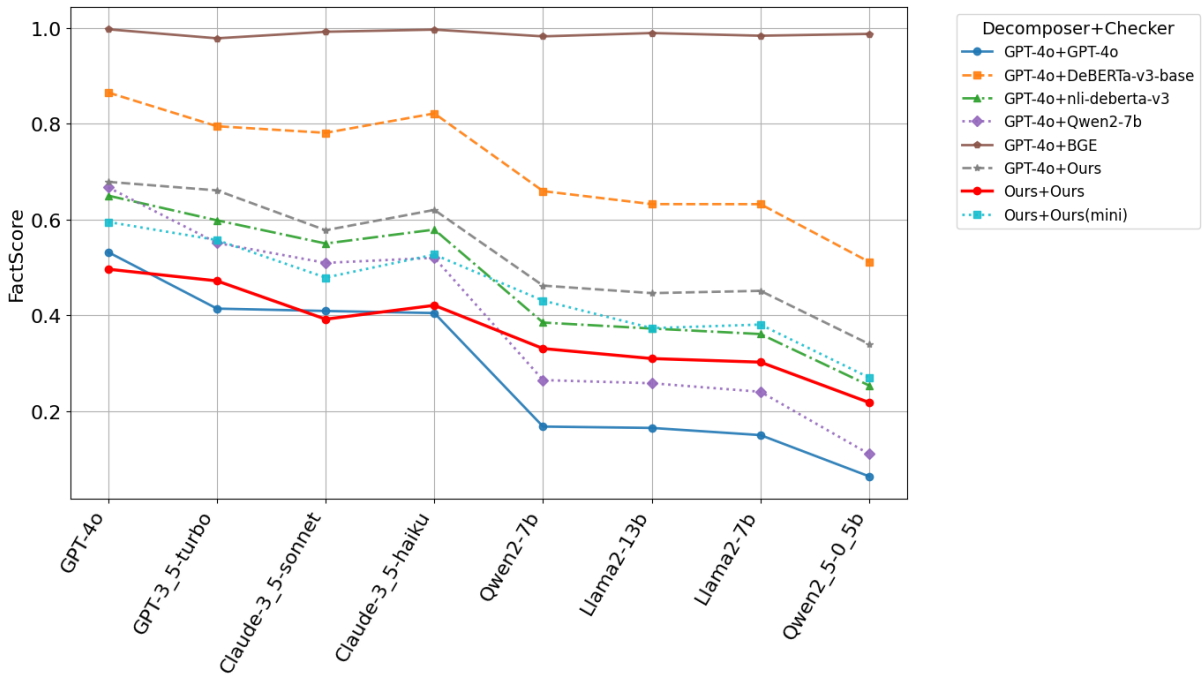

Figure 2: Fact score distribution for different Decomposer + Checker combinations across eight generative models. The x-axis represents the models producing the generated content, while the y-axis shows the corresponding fact scores. Light-FS (Ours+Ours) and its mini variant (Ours+Ours(mini)) maintain strong Spearman and Pearson correlations with GPT-4o's ground truth annotations, ensuring consistent ranking of factual consistency across models. Although Light-FS exhibits lower score variance compared to Qwen2-7b and nli-deberta-v3-base, its ranking of model factuality remains aligned with GPT-4o, demonstrating its reliability as an efficient fact verification framework.

Qwen2-7B model with prompts. The final training set consists of 2,749,030 triplets, with 50,000 pairs allocated for validation and testing, respectively. The prompt used is as follows: 4, 5, 6, 7.

A.3 Experiment Datasets Annotation

We used GPT-4o for decomposition and annotation of the datasets employed in the experiments, with the following prompt used for annotation: 8, 9, 10. Meanwhile, the prompt used for LLM in fact verification is shown in 11.

Figure 3: Few-shot prompt used for sentence-level atomic fact decomposition.

Sentence-level Atomic Fact Decomposition Prompt

Decompose the following sentences into atom facts if possible, response only the decomposition.
Rely solely on the provided text.
Do not infer or assume additional information.
Do not include any additional information.
Just be faithful to the text.
Examples:

Input: Elisha Brown (25 May 1717 - 20 April 1802) served as Deputy Governor of Rhode Island from 1770 to 1772.
Output: Elisha Brown was born on 25 May 1717. Elisha Brown died on 20 April 1802. Elisha Brown served as Deputy Governor of Rhode Island from 1770 to 1772.

Input: George Bovell is currently a professional swimmer and intends to compete in a record fifth Olympiad. Bovell is also respected for his voluntary giving back initiatives such as "The World Swim Against Malaria and Drowning" in Uganda, 2013, with his friend, Ugandan swimmer Max Kanyarezi.
Output: George Bovell is currently a professional swimmer. George Bovell intends to compete in a record fifth Olympiad. George Bovell is respected for his voluntary giving back initiatives such as "The World Swim Against Malaria and Drowning" in Uganda, 2013. George Bovell did this with his friend, Ugandan swimmer Max Kanyarezi.

Input: He now hosts the breakfast slot on 98FM.
Output: He now hosts the breakfast slot on 98FM.

Now expand this biographical statement with the same accuracy and style, ensuring the facts remain unchanged and no additional information is inferred.

Input: `{sentence}`
Output:

Figure 4: Zero-shot prompt used for non-factual sentence.

Non-factual Sentence Generation Prompt

type: type_info
type_dict = {
"time": "Time content: Covers time, dates, periods, etc., related to when events occur.",
"number": "Number content: Includes data, ratios, percentages, etc.",
"entity": "Entity content: Involves specific entities such as names of people, places, organizations, etc.",
"event": "Event content: Describes specific events, activities, actions, etc.",
}

Modify the input sentence by changing only the `{type}` content to make the sentence factually incorrect.
`{type_info}`
Ensure that the sentence structure and meaning remain consistent, but the facts related to the `{type}` content should be altered.
Do not add any new words or use contrasting phrases like 'however' or 'but'.
Just replace the original `{type}` content with a different, incorrect value.
And provide only the modified sentence as a response.
The sentence is: `{sentence}`
Your answer:

Figure 5: Zero-shot prompt used for similar sentence.

Similar Sentence Generation Prompt

Please take the following sentence and rewrite it using various of expressions, but keep the factual information the same.
Do not add any additional information that is not already mentioned in the original sentence.
And provide only the modified sentence as a response.
The sentence is: `{sentence}`
Your answer:

Figure 6: Few-shot prompt used for sentence with extra information.

Sentence with Extra Information Generation Prompt

Given the following sentence, generate a new sentence by adding extra, relevant information, but not too long.
For example:
1. Sentence: Adja Yunkers received a Guggenheim Fellowship.
Answer: Adja Yunkers received a Guggenheim Fellowship in 1956.

2. Sentence: Adja Yunkers was a printmaker.
Answer: Adja Yunkers was an American printmaker.

3. Sentence: Admiral William J. Flanagan, Jr. was born in 1943.
Answer: Admiral William J. Flanagan, Jr. was born in April 3, 1943 in New York City.

4. Sentence: Albert Einstein was awarded the Nobel Prize in Physics in 1921.
Answer: Albert Einstein was awarded the Nobel Prize in Physics in 1921 for his work on the photoelectric effect.

Instructions:
- You may add relevant details such as dates, locations, specific achievements, or additional background information.
- The added details should enrich the meaning of the original sentence, providing more context without overwhelming it.
- Ensure that the sentence remains clear and concise.

Now, consider the following sentence:
Sentence: `{sentence}`
Your answer (please provide only the modified sentence):

Figure 7: Few-shot prompt used for sentence with missing information.

Sentence with Missing Information Generation Prompt

Given the following sentence, generate a new sentence by removing some information.
Ensure the remaining information is still accurate and the core meaning is preserved.

For example:
1. Sentence: Adja Yunkers received a Guggenheim Fellowship in 1956.
Answer: "Adja Yunkers received a Guggenheim Fellowship.

2. Sentence: Adja Yunkers was an American printmaker.
Answer: Adja Yunkers was a printmaker.

3. Sentence: Admiral William J. Flanagan, Jr. was born on April 3, 1943, in New York City.
Answer: Admiral Flanagan was born in 1943 in New York.

4. Sentence: Albert Einstein was awarded the Nobel Prize in Physics in 1921 for his work on the photoelectric effect.
Answer: Albert Einstein was awarded the Nobel Prize in Physics in 1921.

Instructions:
- You may remove specific details such as dates, places, etc.
- Be sure that the modified sentence remains factually correct and conveys the main idea.

Now, consider the following sentence:
Sentence: `{sentence}`
Your answer (please provide only the modified sentence):

Figure 8: Zero-shot prompt used for wiki-bio-hallucination (synthesis).

Wiki-bio-hallucination (synthesis) Generation Prompt

cata = ["time", "number", "event", "entity"]

Based solely on the provided Wikipedia text, write a description of the topic in no more than 60 words.
Use the information mentioned in the text but include two factual inaccuracies about `{cata}` if possible.
Do not alter the main entity (the individual) being described.
Ensure the errors are plausible but intentionally deviate from the provided text.
Wikipedia Text: `{text}`
Response only the description:

Figure 9: Zero-shot prompt used for wiki-bio-hallucination (LLM).

Wiki-bio-hallucination (LLM) Generation Prompt

Write an introduction (within 400 words) in a Wikipedia style about `{name}`.
Provide only the generated introduction.

Figure 10: Few-shot prompt used for passage-level atomic fact decomposition.

Passage-level Atomic Fact Decomposition Prompt

Decompose the following sentences into atom facts, response only the decompositions:
Sentence: Tim Finchem (born August 24, 1947) is an American businessman and former Commissioner of the PGA Tour..... He was inducted into the World Golf Hall of Fame in 2017.
Answer: ["Tim Finchem was born on August 24, 1947.",...,"Tim Finchem was inducted into the World Golf Hall of Fame in 2017."]
Sentence: John Russell Reynolds (1820–1876) was an English lawyer, judge, and author..... He also wrote a biography of the poet John Keats (1848).
Answer: ["John Russell Reynolds was born in 1820.",...,"John Russell Reynolds wrote a biography of John Keats in 1848."]
Sentence: `{sentence}`
Answer:

Figure 11: Prompt used for fact verification.

Fact Verification Prompt

Giving a claim and a paragraph, determine if the claim is supported by the paragraph:

Paragraph: `{paragraph}`

Claim: `{claim}`

Answer (just yes or no):