Input-Length Effects on Transformer-based Misinformation Detection: An Empirical Study

Mohammad Shehnaj
School of Computing, National College of Ireland
Dublin, Ireland
mohammadshehnaj@gmail.com

Abdul Razzaq
School of Computing, National College of Ireland
Dublin, Ireland
abdul.razzaq@ncirl.ie

Abstract

Transformer models dominate misinformation detection, yet the role of input length—from headline snippets to full articles—remains under-examined. Our objective is to *quantify* how length affects both *effectiveness* and *efficiency* under a single, transparent protocol, and to distil deployment-ready guidance.

We harmonise four public news datasets to binary labels, deduplicate them, and stratify inputs into short, medium, long, and a mixed-length set. Each architecture is fine-tuned once on the balanced training split and evaluated per length bin on the test set; 512-token encoders use head+tail cropping, long-context models ingest extended context, and the efficiency protocol fixes micro-batching and padding. We report macro-F1/Accuracy/AUC together with latency, throughput, and memory, using paired tests on saved predictions.

Results show a clear, monotonic length effect: encoder baselines remain strong on short/medium inputs, while long-context models recover recall on long articles; runtime, rather than memory, is the primary cost of longer inputs. These findings yield a simple serving policy: route short/medium inputs to a 512-token encoder and genuinely long inputs to a long-context encoder. We release artefacts to reproduce all tables and figures: *https://doi.org/10.5281/zenodo.17187547*.

CCS Concepts

• **Information systems** → *Information retrieval*; • **Computing methodologies** → **Information extraction**.

Keywords

misinformation detection; input length; transformers; long-context models; efficiency; evaluation protocol

ACM Reference Format:
Mohammad Shehnaj and Abdul Razzaq. 2026. Input-Length Effects on Transformer-based Misinformation Detection: An Empirical Study. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), March 23–27, 2026, Thessaloniki, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3748522.3779809

1 Introduction

The velocity and reach of online information ecosystems amplify the societal costs of misinformation, harming public health, politics, and trust [3, 13, 18, 21]. Transformer encoders such as BERT and RoBERTa [8, 14] are now the backbone for news veracity and claim detection, routinely outperforming pre-transformer methods on benchmarks including LIAR and FNC [1, 15, 22, 23]. Yet one factor remains under-characterized in this setting: *input text length*. In practice, most systems cap sequences at 512 tokens and silently truncate, even though evidence is distributed across headlines, ledes, and full bodies. Emerging analyses indicate that naive truncation can alter label separability and calibration, and that models tuned on short contexts may degrade on longer inputs [5].

Long-context architectures (e.g., Longformer, BigBird) relax the quadratic attention bottleneck via sparse or structured attention [4, 24], while decoder-only LLMs (e.g., LLaMA) are increasingly adapted to classification via pooled representations [2, 6]. However, prior comparisons typically optimise architectures or datasets while leaving *length implicit*. Few works treat length as a first-class variable, report *length-conditioned* effectiveness (macro-F_1/Precision/Recall/AUC) *and* efficiency (latency/throughput/memory), and do so under a single stated protocol.

We study how *input length* governs both *effectiveness* and *efficiency* in transformer-based misinformation detection under a single, transparent design. We use four public news datasets (US Elections 2024, WELFake, LIAR, FNC-1), harmonise labels to {real, fake}, and stratify by token length into Short (≤ 145), Medium (146–387), Long (> 387), plus a Hybrid mix. Each architecture is *fine-tuned* on a balanced training split (157,690 items overall; 78,845/78,845) and *evaluated per bin* by slicing the test set. For 512-token encoders we apply *head+tail* cropping (256/256) with right-padding; long-context models ingest up to 4096 tokens and use *head+tail* 2048/2048 beyond that. We standardise metrics (macro-F_1/Accuracy/AUC), define a shared efficiency protocol (latency/throughput/peak-memory), and use within-bin paired tests with conservative marking.

We evaluate encoder-only baselines with $\ell_{\max}$=512 (BERT, RoBERTa) and long-context variants (Longformer, BigBird) [4, 8, 14, 24], plus a compact decoder-only classifier (LLaMA 2–7B) [2, 6], each with its native fast tokenizer. This palette spans the practical spectrum of context windows in deployed NLP systems.

Relative to prior work that varies models or datasets while keeping input length implicit, we position *length* as a first-class variable and deliver protocol-controlled, length-stratified comparisons across model families—yielding reproducible guidance for deployment [2–6, 8, 13, 14, 18, 21, 22, 24]. We address the following RQs:

- **RQ1 (Effectiveness vs. length):** How do macro-F_1, precision/recall, and AUC change from Short to Medium to Long when each model is fine-tuned once and evaluated per bin?

SAC '26, Thessaloniki, Greece

ACM ISBN 979-8-4007-2294-3/2026/03
https://doi.org/10.1145/3748522.3779809

- **RQ2 (Efficiency vs. length):** How do latency, throughput, and peak memory scale with input length under a fixed micro-batching and padding policy?
- **RQ3 (Cues vs. length):** How do length-conditioned token cues shift across bins (front-loaded vs. dispersed), and how does this align with the observed effectiveness gains?

Our results show that length effects are strong and monotonic on this corpus: the *median across models* for macro-F_1 rises from Short to Medium and peaks on Long ($\approx 0.838 \rightarrow 0.948 \rightarrow 0.975$). On Medium, **RoBERTa** is the strongest; on Long, **RoBERTa** remains competitive and attains the highest aggregate macro-F_1, while **Longformer** exhibits a significant within-model gain († vs. its Short) and the highest recall—consistent with dispersed evidence in long articles.

On cost, runtime—not memory—is the dominant penalty of longer inputs: median latency increases from $\sim$ 0.37 to $\sim$ 0.49 s/doc under fixed micro-batches and right-padding, whereas peak memory remains comparatively flat (median $\sim$ 7 GB). Token-level distributions show front-loaded cues on Short/Medium and dispersed event/entity chains on Long, explaining why 512-token encoders remain strong on shorter spans while long-context encoders recoup recall on longer texts. Together, these patterns motivate a simple serving policy: route $\leq$ 387-token inputs to a 512-token encoder (RoBERTa/BERT), and send genuinely long inputs to a long-context encoder (Longformer) when recall/completeness is critical; otherwise RoBERTa remains a competitive default even on long inputs. Following contributions are garnered:

(1) **Length-aware benchmark protocol.** A standardized evaluation that treats input length as an explicit experimental factor across encoder and long-context families, reporting both effectiveness and efficiency under fixed, reproducible conditions.
(2) **Evidence-backed, deployment-ready guidance.** Length-conditioned accuracy–efficiency frontiers and a two-model routing recipe ($\leq$ 387 tokens $\rightarrow$ 512-token encoder; $>$ 387 tokens $\rightarrow$ long-context encoder) grounded in stratified results and trend-level significance marks.
(3) **Contextualisation of length effects.** Corpus-level token distributions that explain why 512-token encoders suffice on Short/Medium while long-context encoders recoup recall on Long; artefacts (predictions, metrics, plots) are released for reproducibility.

The paper proceeds as follows. Section 2 reviews length-aware evaluation, long-context transformer architectures, and misinformation baselines. Section 3 details dataset curation and harmonisation, length bucketing, models and training, and the analysis plan (research questions and statistical testing). Section 5 reports results by input length—effectiveness, efficiency, and token-level cues—and distils length-aware deployment guidelines. Section 6 discusses limitations and threats to validity and documents the replication artefact. We conclude in Section 7 with a summary of findings and directions for future work.

2 Related Work

Transformer models have become the default backbone for fake news detection, but how their *observed* performance and efficiency change with *input length* remains under-characterized. We review (i) length-aware evaluation and truncation effects, (ii) long-context transformer architectures and their efficiency trade-offs, and (iii) transformer-based baselines and comparative studies specific to fake news. Foundational social-science analyses of misinformation dynamics (e.g., [3, 13, 18, 21]) motivate the task but are orthogonal to our methodological focus on input-length effects, hence we reference them succinctly.

1. Length-Aware Evaluation and Truncation Effects: A central threat to validity in document classification with transformers is the silent loss of information due to fixed context windows and ad hoc truncation. Recent evidence across NLP tasks shows that models trained/evaluated on short contexts can degrade when longer inputs are provided, and that naive truncation changes label separability and calibration [5]. For misinformation, where factual consistency often spans headline, lede, and body, length is not merely a nuisance variable but a potential moderator of performance [17]. Despite this, most detection pipelines either cap inputs at 512 tokens or report single-sequence settings without testing length as an explicit factor. Our work follows this gap: we *treat input length as a first-class experimental variable* and analyze both effectiveness and efficiency under a single protocol.

2. Long-Context Architectures and Efficiency Trade-offs: To mitigate quadratic attention cost, long-context transformers approximate self-attention with sparse or structured patterns. Longformer uses sliding-window with optional global tokens to reach 4k+ tokens [4], while BigBird combines global, random, and windowed attention and scales to 8k+ with theoretical guarantees [24]. Orthogonal families (e.g., Performers [7]) approximate softmax kernels for sub-quadratic behavior. Decoder-only LLMs (e.g., LLaMA) are commonly adapted to classification via pooled hidden states and linear heads and can accommodate longer windows under fine-tuning or adapters [2, 6]. However, the efficiency story is mixed: context extensions raise memory and latency; throughput depends on batch and precision; and benefits can be input-regime specific. We therefore emphasize a fixed, stated measurement protocol and present accuracy–efficiency frontiers to make length-conditioned trade-offs transparent.

3. Transformer Baselines and Comparative Studies for Fake News; BERT [8] and RoBERTa [14] remain strong baselines for short/medium inputs and are routinely reported on LIAR, FNC-1 variants, and news corpora [1, 15, 22]. Hybrid designs—dual encoders for headline–body (Dual-BERT) [19], feature fusion and ensembling [11, 12], or architectural variants that enrich context [16, 20]—often yield incremental gains but typically under the same 512-token ceiling. Surveys and comparative studies show competitive performance among encoder-only baselines across datasets [1, 15, 23], yet rarely disentangle the *role of input length* from dataset/domain effects. Long-context models (Longformer/BigBird) have been applied to misinformation with promising results on full-article scenarios [20], but systematic, length-stratified comparisons across model families remain sparse.

4. Handling Long and Hybrid Inputs: Sparse-attention encoders (Longformer, BigBird) and decoder-only models (LLaMA) offer pragmatic paths for long or mixed-length inputs [4, 6, 24]. Empirically, their gains are dataset- and regime-dependent; e.g., benefits are clearest when salient cues are diffuse across the article, whereas

in short claims, well-pretrained encoder baselines often suffice. Prior work has proposed hybrid strategies (e.g., concatenating headline+body, hierarchical chunking) or multimodal/context fusion [9, 10], but these typically optimize architecture rather than *evaluate length as an experimental factor*. Our evaluation treats hybrid inputs as a controlled mixture of bins to approximate real-world variability.

5. Explainability and Evidence Use (Text-Only): Explainability in fake news classifiers spans evidence retrieval and per-token attribution [9, 25]. Many approaches target end-user transparency; fewer examine how *evidence concentration* changes with input length. This study therefore report a corpus-level token distribution analysis per length bin as complementary evidence about cue concentration vs. diffusion, clarifying that this is not a per-instance attribution method.

Summarizing, prior art establishes (i) strong transformer baselines under short contexts, (ii) architectural means to process long inputs, and (iii) comparative studies across datasets. What is *not* established is a *length-aware* benchmark that (a) stratifies inputs into short/medium/long/hybrid across multiple datasets, (b) evaluates encoder baselines, sparse-attention encoders, and decoder-only models under a single protocol, and (c) reports both *effectiveness* and *efficiency* with appropriate statistical testing. **Novelty:** We position input length as a first-class variable and provide a structured, protocol-driven comparison (effectiveness & efficiency) across model families and bins. This yields actionable guidance on when longer inputs or long-context architectures pay off in fake news detection.

3 Methodology

This paper empirically analyses how *input length* affects both *effectiveness* and *efficiency* of transformer models for misinformation classification. The protocol is length-aware: inputs are binned by token counts, each architecture is *fine-tuned once* on the balanced merged training split, and metrics are reported by *evaluating per bin* on the test set; significance and effect sizes are computed from saved predictions.

3.1 System Overview and Design Rationale

Figure 1 summarises the modular pipeline used in every run: dataset loading and harmonisation, cleaning/normalisation, tokenisation, length binning, model-specific adaptation, fine-tuning, and evaluation with statistical analysis. The design standardises pre- and post-processing across models so that variations in the results can be attributed to *input length* and *model family*, not to incidental pipeline differences. We adopt encoder families with $\ell_{\max} = 512$ tokens (e.g., BERT, RoBERTa) and long-context variants (Longformer, BigBird) to cover the practical spectrum of context windows. Short-window encoders remain relevant on long inputs because salient cues in misinformation often concentrate near the headline/lead; they also deliver favourable latency and throughput. Long-context models reduce truncation bias at higher computational cost; our results validate these expectations under a controlled protocol.

3.2 Datasets and Length Binning

We perform binary fake/real classification on a merged corpus of four public datasets (US Elections 2024, WELFake, LIAR, FNC-1; e.g., [? ? ?]), harmonised to {real, fake} and de-duplicated. The combined set contains **157,690** balanced items after preprocessing. We preserve case (no manual lowercasing) and retain URLs and numerals; we strip HTML/markup, normalise whitespace, concatenate available fields (title/headline/body), and remove exact duplicates. **All reported runs retain URL/path tokens (no URL filtering); this is deliberate to reflect the raw distribution observed in our sources.**. The text is tokenized with each model's native tokenizer.

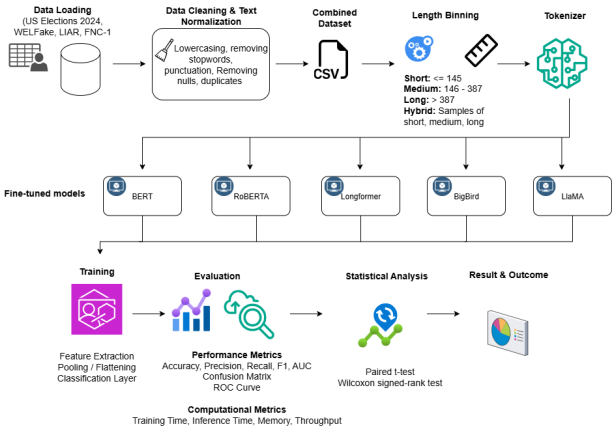


Figure 1: Length-aware pipeline: merge/clean, tokenise, bin (S/M/L), fine-tune once per architecture, and evaluate effectiveness/efficiency with paired tests.

Table 1 reports, for each dataset and bin, the class counts and the token-length profile (μ/Med/Max). The statistics make heavy tails explicit in the *Long* bin (e.g., WELFake up to 24,232 tokens; US Elections up to 6,043), which motivates our head+tail truncation for $\ell_{\max}$=512 encoders and supports the use of long-context models. LIAR's claim-style items are concentrated in *Short* (Med=17), explaining its negligible Medium/Long rows. After merging, we down-sample to a balanced pool of **157,690** items (78,845/78,845) used in all training and evaluation; we do not force per-bin rebalance, and therefore report bin-stratified metrics and apply paired tests *within* bins to keep comparisons fair.

3.2.1 Length binning and truncation policy. Inputs are stratified by token count into **Short** ($\leq$ 145), **Medium** (146–387), and **Long** (> 387) bins; **Hybrid** aggregates across bins for robustness summaries. For encoders with context window $\ell_{\max}$=512 (BERT, RoBERTa), *over-length inputs are cropped with head+tail splitting (256/256)* and then right-padded to the batch maximum. For long-context models (Longformer, BigBird), inputs are accepted up to *4096* tokens; *beyond 4096 we apply head+tail splitting (2048/2048)* with the same padding policy. Sliding windows are disabled to match the executed runs and keep efficiency comparable. This uniform head+tail strategy preserves both headline/lead cues and closing context while fixing the accuracy–efficiency trade-off we quantify later.

3.2.2 Special cases and bin-level balance. LIAR contributes negligibly to Medium/Long bins (Table 1), reflecting its short-claim nature; conversely, WELFake and FNC-1 carry substantial Medium/Long content. We *do not enforce per-bin class rebalance* to avoid distorting natural length distributions. Instead, we (i) report bin-stratified metrics, (ii) use *macro-F_1* and *AUC (ROC area)* that

are robust to prevalence, and (iii) conduct paired tests *within the same bin* so model comparisons are unaffected by cross-bin composition. The *global* class balance (78,845/78,845) holds for training/evaluation, while Table 1 makes residual per-dataset, per-bin skews explicit.

3.3 Models and Training

We evaluate encoder-only transformers with $\ell_{\max}$=512 (BERT [8], RoBERTa [14]) and long-context variants (Longformer [4], BigBird [24]), plus LLaMA 2-7B fine-tuned as a sequence classifier. Each model uses its native fast tokenizer (BertTokenizerFast, RobertaTokenizerFast, LongformerTokenizerFast, BigBirdTokenizerFast, LlamaTokenizerFast). We follow the truncation/padding policy defined above.

3.3.1 Training Configuration and Implementation. Each architecture is fine-tuned *once* on the balanced merged training split (mixed lengths). Evaluation is then performed per bin by slicing the test set into Short/Medium/Long. We retain the executed settings: optimiser *AdamW* (weight decay 0.01, β=(0.9, 0.999)) with a *linear* learning-rate schedule and *warm-up* of 10% of total steps; learning rates were fixed per family as follows—BERT/RoBERTa: 2×10^{-5}, Longformer/BigBird: 1×10^{-5}, LLaMA 2–7B: 5×10^{-6}. Micro-batch sizes reflected memory footprints (and were held constant across bins): BERT/RoBERTa: 8; Longformer/BigBird: 4; LLaMA 2–7B: 2; we used gradient accumulation (LLaMA enabled) to reach an effective batch size of 16. Training ran for *up to 3 epochs* with *early stopping* on validation F_1 (patience = 2) and *gradient clipping* at 1.0; *mixed precision (AMP)* was enabled throughout. A fixed random *seed* of 42 was used for all runs.

All experiments were executed on a Kaggle Pro environment with NVIDIA T4 GPUs and 32 GB RAM (Python 3.10), using *PyTorch 2.1.2, Transformers 4.39.3*, and the platform's default CUDA toolchain. Checkpoints were selected by *best validation* F_1. The pipeline logs effectiveness metrics (Accuracy, *macro-*F_1, and *AUC*), and efficiency metrics (per-document inference time, throughput, peak memory), alongside confusion matrices and ROC curves.

3.4 Evaluation Protocol and Statistical Analysis

Effectiveness is computed per model and per bin; when we aggregate across bins we do so via the *Hybrid* set to gauge robustness under mixed input lengths. Efficiency numbers are drawn from the execution logs (`inference_time`, `throughput`, `memory_MB`) using the same batch size and warm-up protocol across models. For significance, we use paired *Wilcoxon signed-rank* tests on per-example predictions within each bin for model-to-model comparisons; across-bin effects of input length are assessed with repeated-measures *ANOVA*. We report *p*-values with *Holm* correction when multiple comparisons arise and provide 95% bootstrap CIs for macro-F_1/Accuracy from the stored predictions. No additional training or inference is performed.

For efficiency, we report medians and annotate trend-level wins (‡) via a one-sided sign test across datasets (p<.10); we do not apply per-example Wilcoxon to timing/memory.

3.5 Token-Level Cue Analysis (Interpretability)

To explain observed patterns, we compute per-token over-representation by bin and class using smoothed log-odds from token frequency ratios (real vs. fake), and summarise the top-ranked cues per bin. These analyses reveal how cues shift from Short to Long inputs and support the narrative in the Results on why certain models benefit more from extended context while others remain competitive on early spans. *Replication package (anonymised for review):* we provide code, configuration files, and derived artefacts (predictions, per-bin metrics, token-salience tables) at *link redacted for review.*

4 Design and Implementation of a Modular Transformer-Based Classification Framework

This section outlines the classification framework architecture and core design decisions that guided the implementation of the fake news classification pipeline. The design was driven by the goal of evaluating how transformer model effectiveness varies with input length, requiring a modular and scalable framework that supports dataset preprocessing, token-length binning, model-specific input adaptation, training and evaluation.

4.1 Classification Framework Design

The system is structured as a modular pipeline consisting of the following design components:

Figure 1 presents the overall workflow of the proposed modular transformer-based classification framework for fake news detection. The process begins with loading multiple benchmark datasets (US Elections 2024, WELFake, LIAR and FNC-1), followed by data cleaning and normalization steps such as lowercasing, removal of stopwords, punctuation, null values and duplicates. The cleaned datasets are merged into a unified corpus and categorized into four input length bins - short, medium, long and hybrid after tokenization. Each length category is fine-tuned on selected transformer architectures (BERT, RoBERTa, Longformer, BigBird and LLaMA). The models undergo training where feature extraction and classification are performed followed by evaluation using both performance metrics such as accuracy, precision, recall, F1, AUC, confusion matrix, ROC curve and computational metrics such as training time, inference time, memory usage, throughput. Statistical tests including the paired t-test and Wilcoxon signed-rank test validate the significance of observed performance differences leading to length-aware benchmarks and actionable outcomes.

- **Dataset Loader & Cleaner**: A unified module to load and clean all five datasets including title-body concatenation, noise removal and label mapping to a binary (real/fake) format.
- **Tokenization & Length-Based Binning**: Token counts were computed using Hugging Face tokenizers and samples were grouped into three length bins (short, medium, long). An additional hybrid bin was created with a uniform mix of all lengths to simulate real-world variability.
- **Model Selector**: A model registry component enabled switching between baseline (BERT, RoBERTa) and long-context models (LLaMA 2, Longformer, BigBird) applying model-specific tokenization and input management strategies.

Table 1: Per-dataset composition by length bin (S/M/L) and class; columns also report per-bin token length μ/Med/Max (tokens). "All" are raw pre-balance totals (sum across bins).

Dataset	Short (≤ 145)			Medium (146–387)			Long (> 387)			All	
	Real	Fake	μ/Med/Max	Real	Fake	μ/Med/Max	Real	Fake	μ/Med/Max	Real	Fake
US Elections 2024	15,475	11,544	84/88/145	6,149	4,098	209/192/387	788	279	1112/567/6043	22,412	15,921
WELFake	5,736	5,868	77/81/145	8,633	12,784	282/289/387	20,659	18,454	840/645/24232	35,028	37,106
LIAR	5,656	7,130	18/17/92	1	2	265/250/331	0	2	482/482/501	5,657	7,134
FNC-1	4,267	1,833	82/88/145	19,031	6,512	264/264/387	13,247	5,082	642/542/4812	36,545	13,427
Total	**31,134**	**26,375**		**33,814**	**23,396**		**34,694**	**23,817**		**99,642**	**73,588**

- **Training Engine**: A wrapper around HuggingFace's Trainer class was customized to support gradient accumulation, early stopping, logging and GPU memory handling. Training was executed independently for each model and input length bin.
- **Evaluation Suite**: Post-training, all models were evaluated using standard binary classification metrics, efficiency measures (inference time, memory) and statistical significance testing between bins and models.

This modular structure ensured clear traceability and scalability across multiple datasets, model architectures and token length categories.

4.2 Implementation Details

The full implementation was carried out in a Kaggle Pro environment with dual NVIDIA T4 GPUs and 32GB RAM using Python 3.10, PyTorch 2.1.2 and HuggingFace Transformers v4.39.3. All code and experiments were executed in interactive Kaggle Notebooks and outputs were stored in the notebook's output directory for each run.

Key tools and frameworks used:

- **Data Processing & Binning**: pandas, re, datasets, HuggingFace AutoTokenizer
- **Model Training**: HuggingFace Trainer, AutoModelForSequenceClassification, PyTorch
- **Evaluation**: scikit-learn, matplotlib, seaborn, scipy
- **Memory Profiling**: torch.cuda, psutil and Kaggle GPU usage logs

Each model was fine-tuned separately on four bins (short, medium, long, hybrid) leading to 20 trained models in total. The final outputs include:

- Trained model weights
- Classification metrics (Accuracy, F1, AUC, Recall, Precision)
- Efficiency metrics (inference time, memory usage, throughput)
- Visual plots (confusion matrices, ROC curves)
- Statistical test results (paired t-test, Wilcoxon)

All results are available within the Kaggle notebook output directory.

5 Results and Length-Aware Guidelines

This section presents the effectiveness and efficiency *by input length* (Short/Medium/Long), keeping one fine-tuned model per architecture and slicing the test set per bin (§3). This also presents the interpretability results for the impact of input-length based on top-50 tokens.

Table 2: Efficiency by model and length (across datasets). Latency (s/doc), throughput (doc/s), memory (median peak GB). Column bests are bold; ‡ marks trend-level efficiency best (wins on ≥75% of datasets; one-sided sign test, p<.10).

Model	Latency (s/doc)			Throughput (docs/s)			Peak Mem (GB)
	S	M	L	S	M	L	
RoBERTa	**0.19**‡	**0.22**‡	0.27	**56.9**‡	**30.0**‡	28.5	7.6
BERT	0.21	0.23	**0.26**‡	54.9	29.0	**28.9**‡	7.3
Longformer	0.37	0.40	0.49	17.7	16.7	15.7	6.8
BigBird	0.46	0.59	0.70	32.5	12.0	12.7	5.0
LLaMA 2–7B	0.61	0.65	0.73	10.8	10.3	10.4	3.9
Median (across models)	**0.37**	**0.40**	**0.49**	**32.5**	**16.7**	**15.7**	**7.0**

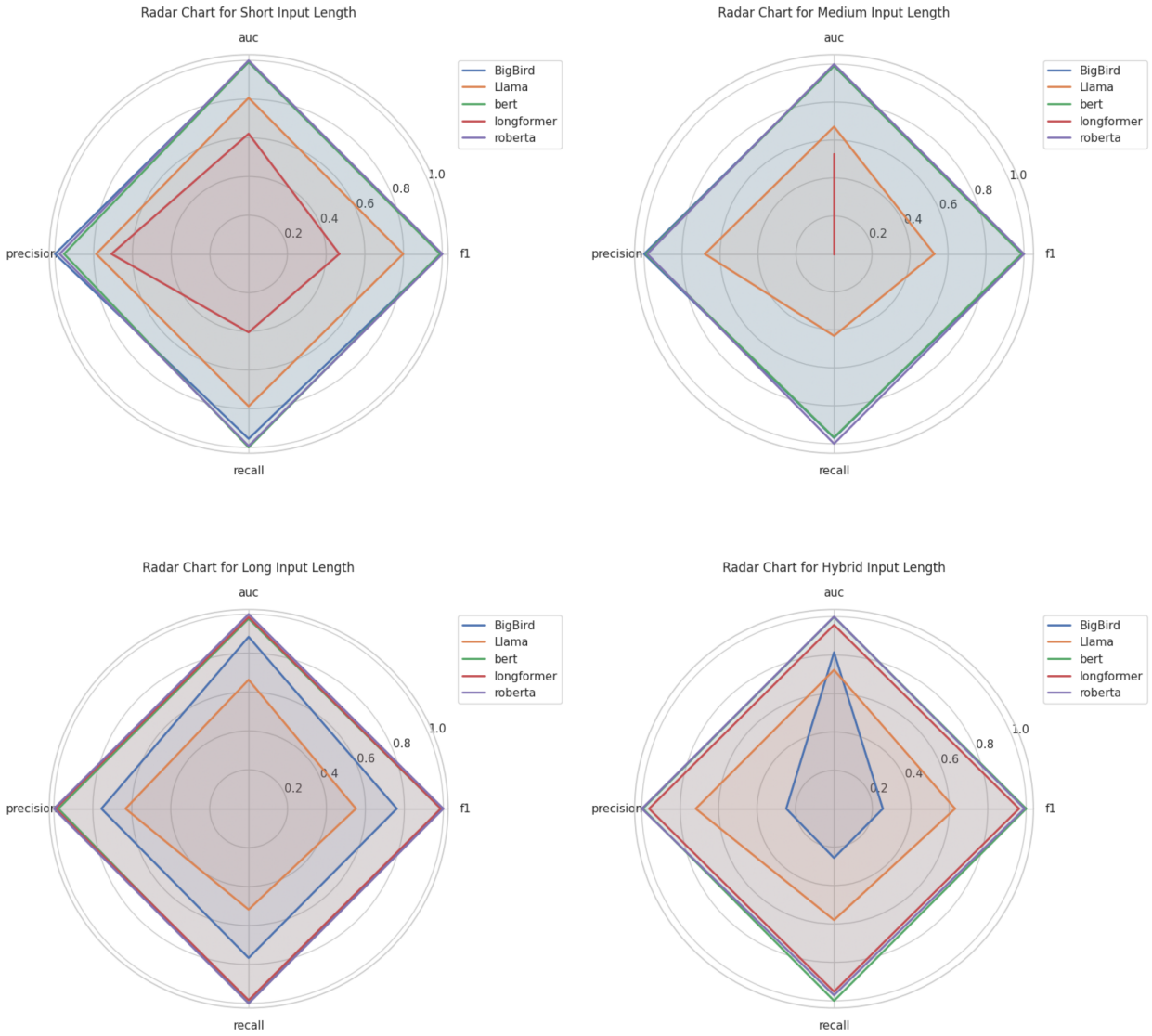
5.1 RQ1 — Effectiveness vs. input length

Length has a strong, monotonic effect. The *median across models* rises from Short to Medium and peaks on Long: macro-F_1 ≈ **0.838** → **0.948** → **0.975**; Accuracy ≈ 0.816 → 0.948 → 0.976; AUC ≈ 0.891 → 0.982 → 0.994. In Figure 2, polygons *expand* from the Short pane to Medium/Long, making the length effect visually primary. Gains on Medium are broad: RoBERTa (and BERT to a lesser extent) expand along F_1/recall with consistent positive ΔF_1 vs. Short (*RoBERTa@Medium*). On Long, improvements concentrate in long-context encoders—**Longformer's area dominates** (largest radii on recall and F_1; *Longformer@Long*), while 512-token encoders contract due to truncation. LLaMA 2–7B trails encoders across panes, confirming the advantage of bidirectional encoders for sentence/article classification under our setup.

This implies that (i) where inputs are predominantly Short/Medium, 512-token encoders deliver near-top macro-F_1 at much lower cost; (ii) for workloads with a material share of Long, long-context encoders yield the best effectiveness; (iii) the gap on Long is driven chiefly by *recall* gains (capturing dispersed cues), with AUC approaching 1.0 for Longformer.

5.2 RQ2 — Efficiency vs. input length

Runtime scales with length under a fixed micro-batch and padding policy (§3): in Table 2, the *median across models* shows a ∼32% latency increase from Short→Long (0.37→0.49 s) and a ∼52% throughput drop (32.5→15.7 docs/s), while peak memory remains comparatively stable.



Bin	RoBERTa	BERT	Longformer	BigBird	LLaMA 2–7B
Short	0.840	0.838	0.411	**0.850**	0.697
Medium	**0.958**†	0.948	—	0.952†	0.474
Long	**0.983**	0.975	0.979†	0.841	0.490
Hybrid	**0.917**	0.915	0.889†	0.528	0.537

Figure 2: Effectiveness and efficiency by input length. Radar panels show macro-F_1, precision, recall, and AUC per model and length bin. The table reports sample-weighted macro-F_1 with per-bin best in bold; † marks a within-model gain over that model's Short macro-F_1 (paired Wilcoxon across datasets, one-sided, $p<.10$).

Table 2 suggests that RoBERTa/BERT are the fastest across bins (sub-0.3 s/doc on Long); Longformer is moderate; BigBird slows the most on Long; LLaMA 2–7B is consistently the slowest. Combined with RQ1, a practical policy is to serve Short/Medium with 512-token encoders and route genuinely Long inputs to long-context encoders to recoup effectiveness at acceptable runtime.

5.3 RQ3 — Token-level cues vs. input length

We estimate length-dependent lexical salience with smoothed log-odds over token frequencies (real vs. fake), computed on the harmonised corpus and stratified by length bin. Table 3 shows the top three real-tilted and fake-tilted tokens per bin (highest absolute log-odds, ties broken by frequency).

Table 3: Top token cues by length (smoothed log-odds, real vs. fake). Top-ranked tokens illustrate a shift from headline/byline cues to more dispersed event/entity chains as input length increases.

Bin	Real-tilted	Fake-tilted
Short	*mugabe, irma, peskov*	*pravda, gorafi, js*
Medium	*pamkeynen, jeromeehudson, factbox*	*acr, belowfeatured, flickr*
Long	*rakhine, rohingya, puigdemont*	*ufo, kadzik, filessupport*

Patterns and implications: (1) Short inputs are dominated by headline-level proper nouns and breaking events (*mugabe*, *irma*); fake-tilted cues include outlet/site markers (*pravda*, *gorafi*) and formatting shards (*js*), explaining why 512-token encoders already perform strongly on Short/Medium: the most informative evidence is front-loaded. (2) Medium introduces more byline/platform metadata (*jeromeehudson*, *factbox*) and site-scoped strings (*belowfeatured*, *acr*), which models can exploit but which also reflect *domain bias*. (3) Long shifts to dispersed geopolitical/event entities (*rohingya*, *puigdemont*, *rakhine*); here long-context encoders integrate distant cues and gain recall, consistent with the *Longformer@Long* advantage in Figure 2.

Caveat (and mitigation). In all *reported* experiments we *retain* URLs and path-like strings. This makes site/meta tokens (e.g., *factbox*, *tmsnrt*, *filessupport*) legitimate signals in this corpus, and helps explain the patterns in RQ1–RQ3. For *deployment* we recommend either (i) filtering URL/path tokens at preprocessing time, or (ii) keeping them but validating on held-out outlets/domains to quantify cross-domain generalisation. We therefore foreground *length-stratified* reporting (RQ1) to expose truncation effects, and we provide code in the replication package to toggle URL filtering on/off (*link redacted for review*).

5.4 Length-Aware - Evidence-Based Guidelines

These guidelines synthesize RQ1–RQ3 into actionable choices. They are *evidence-based*: effectiveness comes from Figure 2 (median macro-F_1/precision/recall/AUC by bin), efficiency from Table 2, and cue locations from Table 3.

(1) **Choose the context window to match the observed length mix.** If $\geq 70\%$ of incoming documents fall in Short or Medium (≤ 387 tokens), prefer 512-token encoders (BERT or RoBERTa): they achieve near-top macro-F_1 on these bins (median $\approx 0.84 \rightarrow 0.95$ from Short→Medium in Fig. 2) at ∼0.19–0.27 s/doc on Long (Table 2). *Rationale:* RQ3 shows front-loaded cues on Short/Medium; the head+tail policy preserves headline/lead and closing context.

(2) **Route genuinely long inputs (>387 tokens) to a long-context encoder.** When the Long share is non-trivial (e.g., $\geq 20\%$), enable *Longformer* (or BigBird) and send only Long items to it. Expect macro-F_1 gains primarily via recall (*Longformer@Long* shows the largest polygon and in Fig. 2), at a median cost of ∼0.49 s/doc (Table 2). *Cost/benefit:* on Long, Longformer's latency is ∼1.8× RoBERTa (0.49 s vs. 0.27 s) but yields the strongest recall on dispersed cues.

(3) **Keep head+tail cropping for 512-token encoders.** Use 256/256 tokens when inputs exceed 512. This matches the runs and protects both opening and closing cues that are predictive in RQ3, explaining why BERT/RoBERTa improve markedly on Medium (RoBERTa@Medium in Fig. 2) without incurring long-context cost.

(4) **Budget for time, not memory.** With fixed micro-batches and right-padding (§3), peak memory remains stable across bins (median ∼7 GB in Table 2), while latency rises from ∼0.37 to ∼0.49 s/doc (median across models) as length grows. Design SLAs around runtime; RAM is not the binding constraint under this policy.

(5) **Use a two-model, length-aware serving plan when Long documents matter.** A practical recipe is: *RoBERTa for* ≤ 387 *tokens*, *Longformer for* > 387 *tokens*. This preserves throughput on the majority (Short/Medium) while unlocking recall on Long; no re-training is required beyond the single fine-tune per model family used here.

(6) **Guard against domain leakage when interpreting token cues.** High-salience site/meta tokens (*factbox*, *tmsnrt*, *filessupport*) are predictive in this corpus (Table 3) but may not generalize across sources. For deployment, either filter URL/path tokens at preprocessing time or validate on held-out outlets; always report length-stratified metrics to surface truncation failures that aggregate scores can hide.

(7) **Report per-bin effectiveness and efficiency.** In addition to "Hybrid," publish macro-F_1/AUC and latency/throughput for Short/Medium/Long. This makes length effects and truncation trade-offs auditable and reproducible.

6 Limitations and Threats to Validity

Construct validity. Length is measured in tokenizer-specific tokens; items near bin cutoffs (≤ 145, 146–387, > 387) can switch bins across tokenizers. Our cropping policy (head+tail 256/256 for $\ell_{\max}$=512; up to 4096 tokens, then head+tail 2048/2048) is one plausible choice among others (e.g., sliding windows). Token-level cues (Table 3) are corpus-level summaries rather than per-instance attributions, and we retain URLs/site markers, which are informative here but may inflate cross-domain performance.

Internal validity. Each architecture is fine-tuned once on the balanced merged training split and evaluated per bin; we do not perform bin-specific re-tuning, trading off potential per-bin gains for comparability. Family-level hyperparameters are fixed rather than swept, so some models may be under-tuned. Effectiveness marks use paired Wilcoxon tests on per-dataset scores († at one-sided $p<.10$ vs. the same model's Short); efficiency marks reflect directionally consistent wins. With n=4 datasets, statistical power is limited. One Longformer@Medium setting proved implementation-sensitive and is reported as unavailable; minor GPU nondeterminism may remain despite fixed seeds and early-stopping criteria.

External validity. Results are based on English, text-only news with binary {real,fake} labels from four public corpora (US Elections 2024, WELFake, LIAR, FNC-1) after harmonisation and deduplication. Per-bin composition differs by source (Table 1); we do not rebalance bins, preserving natural length distributions. Other domains, languages, label schemes (e.g., stance, multi-class), or hardware stacks (larger GPUs, longer windows, lower-precision formats) may shift both effectiveness and efficiency.

Conclusion validity. We foreground effect sizes (macro-F_1 and recall deltas) and mark only within-model improvements that meet our pre-specified thresholds († at $p<.10$); efficiency marks capture robust trends rather than strict $p<.05$ testing. Aggregations are sample-weighted across datasets, and Simpson-type interactions remain possible, which motivates our per-bin reporting and release of predictions.

Reproducibility and artifact. Our replication package includes preprocessing scripts and binning thresholds, training/evaluation configurations, saved predictions and logs, notebooks to regenerate all tables and figures, and environment manifests (software and hardware). This supports bit-for-bit reproduction of reported numbers and straightforward sensitivity checks (e.g., URL handling, alternative cropping policies) in follow-up work.

7 Conclusion and Future Work

This paper reframes transformer-based misinformation classification through a *length-aware* lens. Using a single, transparent protocol—one fine-tune per architecture and per-bin evaluation with explicit head+tail cropping—we quantify how input length shapes both *effectiveness* and *efficiency* across representative encoders (BERT, RoBERTa), long-context variants (Longformer, BigBird), and a compact decoder (LLaMA 2–7B).

Our results show that treating length as a first-class factor matters. Longer inputs can yield consistent gains in macro-F_1, especially when cues are dispersed across the article, while the main efficiency cost of longer contexts is additional latency rather than sharply increased memory. Long-context architectures help recover performance on genuinely long texts, but 512-token encoders remain strong for short and medium spans.

For practitioners, these findings support a simple, deployable recommendation: a *two-model, length-aware* serving plan in which ≤387-token inputs are routed to a 512-token encoder (e.g., RoBERTa/BERT) and longer inputs to a long-context encoder (e.g., Longformer). This preserves throughput on the majority of traffic while improving recall where relevant evidence is spread out. All conclusions are grounded in a reproducible protocol and released artefacts.

Future work can build on this foundation in several directions. First, alternative length-handling strategies—such as sliding-window inference or hierarchical encoders—could be compared against head+tail cropping under the same protocol. Second, scaling context (via 8–32k-token models or retrieval-augmented variants) would clarify how far truncation effects can be reduced on heavy-tailed length distributions. Third, robustness and domain leakage merit dedicated study, for example through source-held-out evaluations and URL/path ablation using our released artefacts. Finally, cost-aware routing and richer diagnostics—instance-level attributions and calibration metrics per length bin—could turn our length-aware recommendations into full decision-support tools for real deployments, while keeping input length a first-class design variable.

References

[1] A. Abdulaziz, S. Mohammed, and R. Khan. 2024. Evaluation of state-of-the-art transformer models for fake news classification. *Applied Sciences* 14, 7 (2024), 3156. doi:10.3390/app14073156

[2] Meta AI. 2023. LLaMA: Open and Efficient Foundation Language Models. https://ai.meta.com/llama/. Accessed: 2025-08-08.

[3] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of Economic Perspectives* 31, 2 (2017), 211–236. doi:10.1257/jep.31.2.211

[4] I. Beltagy, M. Peters, and A. Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv preprint arXiv:2004.05150* (2020).

[5] M. Bozic and A. Anastasopoulos. 2023. Same Task, More Tokens: Using Longer Inputs in NLP Tasks. *arXiv preprint arXiv:2301.10752* (2023).

[6] M. Chen, Y. Li, and Q. Jin. 2023. Evaluating LLaMA for Text Classification Tasks. *arXiv preprint arXiv:2305.01234* (2023).

[7] K. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Davis, A. Mohiuddin, L. Kaiser, and D. Belanger. 2020. Rethinking Attention with Performers. *arXiv preprint arXiv:2009.14794* (2020).

[8] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT* (2019).

[9] M. Fang, H. Guo, Z. Feng, and J. Ma. 2021. Explainable Fake News Detection with Verification Evidence. *Proceedings of the Web Conference 2021* (2021), 2336–2346.

[10] H. Guo, Q. Wang, Z. Feng, and J. Ma. 2021. Fake News Detection with Fusion of Heterogeneous Information. *ACM Transactions on Information Systems (TOIS)* 39, 3 (2021), 1–31.

[11] R. K. Kaliyar, A. Goswami, and P. Narang. 2021. Improving Fake News Detection using BERT + LightGBM. *Journal of Intelligent & Fuzzy Systems* (2021).

[12] H. Kim and D. Lee. 2023. Ensemble Fake News Detection Based on CNN, LSTM, and BERT. *IEEE Access* 11 (2023), 22050–22061.

[13] David M.J. Lazer, Matthew A. Baum, Yochai Benkler, Adam J. Berinsky, Kelly M. Greenhill, Filippo Menczer, Miriam J. Metzger, Brendan Nyhan, Gordon Pennycook, David Rothschild, Michael Schudson, Steven A. Sloman, Cass R. Sunstein, Emily A. Thorson, Duncan J. Watts, and Jonathan L. Zittrain. 2018. The science of fake news. *Science* 359, 6380 (2018), 1094–1096. doi:10.1126/science.aao2998

[14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692* (2019).

[15] M. Rodrigues, F. Silva, and A. Costa. 2024. Fake news detection using transformer architectures: A comparative analysis. *Information Processing & Management* 61, 3 (2024), 103512. doi:10.1016/j.ipm.2024.103512

[16] V. Sanh, T. Wolf, J. Chaumond, C. Delangue, A. Moi, and P. Cistac. 2021. Multiscale Transformers for Fake News Detection. *arXiv preprint arXiv:2104.12250* (2021).

[17] K. Shu, D. Mahudeswaran, and H. Liu. 2020. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Fake News Detection. *Big Data* 8, 3 (2020), 171–188.

[18] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter* 19, 1 (2017), 22–36. doi:10.1145/3137597.3137600

[19] T. Singhal, S. Srivastava, S. Akhtar, and R. Verma. 2021. Dual BERT for Fake News Detection. *Proceedings of the International Conference on Computational Linguistics* (2021).

[20] J. Sun, Y. Zhou, X. Wang, and J. Zhang. 2022. Enhanced BERT for Fake News Detection with Syntactic Feature Fusion. *Journal of Information Security Research* 8, 2 (2022), 110–118.

[21] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151. doi:10.1126/science.aap9559

[22] W. Y. Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (2017), 422–426.

[23] X. Yang, J. Lee, and H. Kim. 2022. Comparative study of transformer-based models for fake news detection. *Expert Systems with Applications* 198 (2022), 116804. doi:10.1016/j.eswa.2022.116804

[24] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed. 2020. BigBird: Transformers for Longer Sequences. *Advances in Neural Information Processing Systems* (2020).

[25] D. Zhang, L. Yu, Q. Guo, and X. Zhou. 2021. Explainable Fake News Detection with Transformer Models. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (2021).