Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

Towards Trustworthy Large Language Models: An Integrated Framework for Explainability, Hallucination Detection, and Token-Efficient Inference

Sakshi Parate[1], Shreyans Sanyal[2]

[1,2]School of Computer Science, UPES Dehradun

Abstract

Large language models (LLMs) are increasingly deployed in domains where opacity, factual un-reliability, and computational cost carry real consequences, yet the research communities that address these three problems — explainability, hallucination, and inference efficiency — have largely developed in isolation. This paper argues that the three concerns are not independent: interventions designed to reduce inference cost, such as prompt compression, can silently alter both the faithfulness of post-hoc explanations and the model's propensity to hallucinate. We present an integrated conceptual frame-work that couples attention- and perturbation-based explainability (LIME, SHAP, raw attention) with lightweight hallucination-detection signals and token-efficient inference strategies (chunking, summa-rization, prompt compression), and we instrument the framework with a set of cross-cutting consistency metrics. To ground the framework empirically, we design and execute a controlled pilot experiment that measures how rule-based prompt compression jointly affects (i) the Kullback–Leibler divergence between a model's output distributions before and after compression, used as a proxy for hallucination risk, and (ii) the entropy and rank-correlation of last-layer attention, used as a proxy for explanation faithfulness. Because the sandboxed experimental environment used for this study has no network ac-cess to pretrained model repositories, the experiment is conducted on a compact two-layer Transformer language model trained from scratch on a controlled synthetic corpus, which allows exact, reproducible control over ground truth while preserving the qualitative mechanics of attention-based attribution and next-token prediction under compression. Across 15 held-out prompts, compression removes 52.9% of tokens on average while producing a small but non-zero mean output KL divergence of 0.0007 and re-ducing mean attention entropy from 1.062 to 0.632, with attention rank correlation across compression conditions of only 0.633. These results provide direct, quantitative evidence for the framework's central hypothesis: token-efficiency interventions are not explanation-neutral, and systems that report compres-sion ratios without also reporting faithfulness and hallucination-risk deltas may be masking a three-way trade-off. We discuss the implications for trustworthy LLM system design and outline how the pilot findings motivate follow-up experiments on larger pretrained models.

Keywords: explainable AI, LLM hallucination, prompt compression, attention faithfulness, token effi-ciency, trustworthy AI

1. Introduction

Large language models (LLMs) built on the Transformer architecture [1] have moved from research artifacts to production infrastructure in domains such as healthcare triage, legal document review, and financial advisory systems. In these settings, three failure modes threaten user trust simultaneously. First, LLMs are *opaque*: the mapping from billions of parameters and self-attention weights to a specific generated token is not directly interpretable by a human decision-maker, motivating a large literature on explainable AI (XAI)

for language models, including local surrogate methods such as LIME [2] and cooperative-game-theoretic attribution methods such as SHAP [3]. Second, LLMs *hallucinate*: they generate fluent, confident text that is factually incorrect or unsupported by any source, a phenomenon documented extensively in benchmarks such as TruthfulQA [5] and in surveys of intrinsic versus extrinsic hallucination [6]. Third, LLMs are *expensive to run at scale*: the quadratic cost of self-attention in sequence length, combined with growing context-window sizes, has made token-level efficiency — through chunking, retrieval, summarization-based compression, and prompt compression techniques such as LLMLingua [7] — a first-class systems concern rather than an afterthought.

The dominant pattern in the literature is to treat these three concerns as separate research programs, each with its own benchmarks, metrics, and evaluation culture. XAI papers report faithfulness and plausibility scores for explanations without discussing whether the underlying inference pipeline has been compressed for cost reasons. Hallucination-detection papers report detection accuracy without discussing whether the same detection signal survives prompt compression. And prompt-compression papers report token savings and downstream task accuracy without discussing what happens to explanation quality or hallucination rate under compression. This separation is a problem because, in practice, a single production LLM pipeline typically does all three things at once: it compresses or truncates long inputs to control cost and latency, it is queried by stakeholders who want an explanation for its output, and it is expected not to fabricate information. If compression silently degrades explanation faithfulness or increases hallucination risk, then a system that reports healthy accuracy, healthy compression ratios, and healthy explanation scores *in isolation* may still be untrustworthy as an integrated whole.

This paper makes three contributions. First, we synthesize an integrated framework for trustworthy LLM inference that explicitly couples explainability, hallucination detection, and token-efficient inference into a single pipeline with shared instrumentation (Section 3), building on the staged research plan and de-sign rationale documented in the source project specification that motivated this study. Second, we identify *compression-faithfulness-hallucination interaction* as a concrete, measurable research question and propose a metric suite (attention entropy, attention rank correlation, and output distributional divergence) to quantify it (Section 4). Third, we design and execute a controlled pilot experiment that directly measures this interaction (Sections 5–6). Because the experimental environment used for this work is network-isolated from public model hubs, we could not download a pretrained model such as GPT-2 or DistilBERT as originally planned; we address this constraint transparently by training a small Transformer language model from scratch on a synthetic, fully-controlled corpus, which lets us isolate the mechanism of interest (compression's effect on attention and output distributions) without confounds from an uncontrolled pretraining corpus. We report the resulting quantitative findings, discuss what they do and do not imply about production-scale LLMs, and outline the follow-up experiments needed to validate the effect at scale.

The remainder of this paper is organized as follows. Section 2 reviews background on Transformer self-attention, explainability methods, hallucination taxonomies, and token-efficient inference. Section 3 presents the integrated framework. Section 4 defines the experimental methodology and metrics. Section 5 describes the experimental setup in full, including the constraints of the sandboxed environment. Section 6 reports results. Section 7 discusses implications, and Section 8 states limitations before Section 9 concludes.

2. Background and Related Work

Transformer Self-Attention

The Transformer architecture [1] computes, for each layer and attention head, a set of query, key, and value projections of the input token representations, and produces a weighted sum of value vectors where the weights are given by a softmax over scaled dot products of queries and keys. Formally, for query matrix Q, key matrix K, and value matrix V,

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (1)$$

Multi-head attention runs several such operations in parallel with independently learned projections and concatenates the results. Decoder-only, autoregressive LLMs such as the GPT family apply a causal mask so that position t can only attend to positions $\leq t$. Because attention weights are, by construction, a normalized distribution over input tokens, they have been widely (if controversially) treated as a proxy for token importance, and the disagreement between attention-based and perturbation-based attribution is itself an active research question [4].

Explainability Methods

LIME [2] explains an individual prediction by perturbing the input (e.g., dropping tokens) and fitting an interpretable local surrogate model to the resulting change in output, yielding token-level importance scores that are local to a single input. SHAP [3] instead treats attribution as a cooperative game among features and computes Shapley values, which are the unique attribution satisfying a set of fairness axioms, at the cost of being more computationally expensive to approximate. Both methods are model-agnostic and post-hoc, in contrast to raw attention weights, which are intrinsic to the model but not guaranteed to be faithful, in the sense that a high attention weight on a token does not necessarily imply that token caused the output change to the degree the perturbation-based methods would report. The relationship between attention and faithfulness is precisely the axis this paper's experiment probes, using attention entropy and rank correlation as lightweight, architecture-intrinsic stand-ins for the more expensive LIME/SHAP comparison described in the original project plan.

Hallucination in LLMs

Hallucination is commonly split into *intrinsic* hallucination, where the generated text contradicts an available source, and *extrinsic* hallucination, where the generated text is unverifiable against any source at all [6]. Because next-token prediction optimizes for the statistical likelihood of the training distribution rather than for factual correctness, a model can generate confident, fluent, and completely fabricated content whenever the training distribution under-specifies the correct answer or the prompt pushes the model outside its reliable coverage. Benchmarks such as TruthfulQA [5] are designed to elicit this failure mode by ask-

ing questions that invite common misconceptions. Detection approaches include self-consistency sampling (comparing multiple stochastic generations for agreement), semantic-similarity scoring against retrieved evidence, and entropy- or logit-based uncertainty estimation, all of which trade off detection accuracy against computational overhead.

Token-Efficient Inference

Because self-attention scales quadratically with sequence length, long inputs are costly. Retrieval-Augmented Generation (RAG) reduces the need for long contexts by retrieving only relevant passages [8]. Prompt-compression techniques such as LLMLingua [7] instead compress the prompt itself, removing tokens esti-mated to be low-information under a smaller auxiliary language model, while chunking and sliding-window approaches process long documents in bounded segments. All of these techniques implicitly assume that removed or compressed tokens do not carry information the model or a downstream explanation consumer needed — an assumption this paper's experiment tests directly, in a simplified setting, by measuring what changes in the model's attention and output distribution when tokens are removed by a fixed, interpretable rule rather than a learned compressor.

Gap Addressed by This Work

To our knowledge, no prior study jointly instruments the same inference pass for (a) an explanation-faithfulness signal and (b) a hallucination-risk signal under a controlled token-compression intervention, on the same set of prompts, in the same pipeline. Section 3 formalizes the integrated framework that motivates this joint instrumentation, and Sections 4–6 report an initial, small-scale but fully controlled empirical test of it.

3. An Integrated Framework

We propose a four-stage pipeline that unifies the three concerns identified in Section 1, illustrated conceptually below and directly reflecting the phased plan — environment setup, task-specific inference, LIME/SHAP and attention analysis, GPU-scale replication, hallucination reproduction and detection, mitigation, tokenization analysis, and compression — that this study's source specification laid out as a semester-length research program.

Stage 1. Input triage. An incoming prompt is tokenized and its length is checked against the model's context window. If the prompt exceeds a configurable budget, a compression module is invoked (rule-based pruning in this paper's experiment; learned compressors such as LLMLingua in a production system).

Stage 2. Inference with instrumentation. The (possibly compressed) prompt is passed through the LLM with attention weights retained. Rather than treating this as a black-box call, the pipeline records the full attention tensor for at least the final layer.

Stage 3. Explainability and hallucination scoring. Two parallel analyses run on the instrumented forward pass: an attribution analysis (attention entropy/rank correlation as a lightweight proxy, or full SHAP/LIME when compute budget allows) and a hallucination-risk analysis (output-distribution divergence relative to an uncompressed or reference pass, or semantic-similarity/self-consistency scoring against retrieved evidence in a production setting).

Stage 4. Unified trust report. The two scores are combined with the realized token savings into a single report — a "trustworthiness rating" in the terminology of the source specification — so that a system operator can see, for a given compression setting, the joint cost in explanation faithfulness and hallucination risk, not merely the latency or dollar savings.

The key design claim of this framework is that Stages 3 and 4 must be computed *on the same forward pass that a user actually receives an answer from*, rather than as a separate offline evaluation on uncompressed inputs, because the whole point of the framework is to detect when compression itself is the source of a faithfulness or hallucination regression. Section 4 operationalizes Stage 3 and 4 as concrete, computable metrics.

4. Methodology

Research Question and Hypothesis

We ask: *does removing tokens from a prompt to save inference cost measurably change (a) the model's output distribution and (b) the internal attention pattern the model uses to produce that output, on the same input semantics?* Our hypothesis, following the framework in Section 3, is that compression is **not** faithfulness-neutral: even a compression rule that preserves all content words and removes only function words/stopwords will shift attention mass and, to a smaller degree, shift the output distribution, because self-attention position and local context change when tokens are deleted rather than masked.

Metrics

We define three metrics, each computed by running the *same* model on the uncompressed prompt x and a compressed variant $\tilde{x}$ derived from x by a fixed deletion rule.

(1) Output KL divergence (hallucination-risk proxy). Let $p(\cdot \mid x)$ and $p(\cdot \mid \tilde{x})$ be the next-token probability distributions produced by the model for the full and compressed prompts, respectively. We compute

$$D_{\mathrm{KL}}\left(p(\cdot \mid x) \,\|\, p(\cdot \mid \tilde{x})\right) = \sum_{v \in V} p(v \mid x) \log \frac{p(v \mid x)}{p(v \mid \tilde{x})}. \qquad (2)$$

A larger divergence indicates that compression pushed the model toward a different continuation, which we treat as a proxy for increased risk of an unsupported/altered output — the same qualitative failure mode as hallucination, though we emphasize in Section 8 that this is a distributional proxy, not a factuality judgment.

(2) Attention entropy (explanation-sharpness proxy). For the last-layer, head-averaged attention row over the final query position, we compute Shannon entropy $H = -\sum_i w_i \log w_i$ where w_i is the attention weight on source token i. Lower entropy indicates a sharper, more concentrated (and typically more in-
terpretable) attribution; a large entropy shift between the full and compressed conditions indicates that the explanation itself is unstable under compression.

(3) Attention rank correlation (faithfulness-consistency proxy). For the subset of tokens that survive compression, we compute the Spearman rank correlation between the attention weights those tokens receive in the uncompressed pass and the attention weights the same tokens receive in the compressed pass. A correlation near 1 indicates that compression did not reorder which surviving tokens the model considers

important; a correlation near 0 indicates that the explanation for *which words matter* is not stable under compression, which is directly relevant to any deployed system that shows attention- or SHAP-based explanations to end users.

Compression Rule

To keep the experiment fully controlled and reproducible, we use a fixed, non-learned compression rule: remove a closed set of stopwords and connective/function words (articles, prepositions, and a small set of temporal/evaluative modifiers), leaving content words (subjects, verbs, objects) intact. This is a simplified stand-in for learned prompt-compression systems such as LLMLingua, chosen specifically because its deterministic behavior lets us attribute any observed effect to the act of token removal itself rather than to an additional learned component.

5. Experimental Setup

Environment Constraint and Its Resolution

The source research plan that motivated this study specifies loading pretrained models such as GPT-2 and DistilBERT via the HuggingFace Hub. The sandboxed compute environment used to execute this study, however, restricts outbound network access to a small set of package registries (PyPI, npm, and source-code hosts) and does not permit connections to huggingface.co or other model-weight hosting services; a live attempt to load gpt2 in this environment fails with an HTTP 403/connection error. Rather than report a purely qualitative or hypothetical experiment, we adapted the design to remain fully quantitative and reproducible within the environment's constraints: we trained a compact Transformer language model *from scratch*, using PyTorch, on a synthetic corpus that we generated ourselves specifically for this study (no external or copyrighted text was used). We report this substitution transparently, and Section 8 discusses precisely what does and does not generalize from it to a pretrained, web-scale LLM.

Model Architecture

The model is a 2-layer, 4-head Transformer encoder used autoregressively with a causal attention mask (i.e., a GPT-style decoder built from nn.TransformerEncoderLayer with a subsequent-position mask), with model width $d_{\text{model}} = 96$, feed-forward width 256, dropout 0.1, and a context window of 16 to-kens. Token and learned positional embeddings are summed at the input. The model has approximately 200K parameters, several orders of magnitude smaller than GPT-2 (117M parameters), which we treat as a deliberate simplification that trades ecological validity for full experimental control and reproducibility, consistent with the ”toy but real” pedagogical approach recommended for early-stage pipeline validation in the source project plan (Weeks 3–4).

Training Corpus

We algorithmically generated 4,000 template-based declarative sentences of the form [subject] [verb] [object] [connector], drawing from 15 subjects, 14 verbs, 15 objects, and 10 connective clauses (e.g., “the engineer measures the climate after the recent review”), yielding a vocabulary of 77 word types and 39,261 training tokens. This synthetic, closed-vocabulary corpus was chosen specifically so that ground-truth sentence structure is known exactly, which is what allows us to define a deterministic, content-word-preserving compression rule and to interpret attention behavior without the confound of an uncontrolled, noisy pretraining corpus.

Training Procedure

The model was trained for 1,500 steps with batch size 64 and context length 16, using the AdamW optimizer with learning rate 3 10^{-4} and standard cross-entropy next-token loss. Training loss decreased from 4.54 at initialization to 1.07 at convergence (Table 1), indicating the model learned the templated subject–verb–object–connector structure substantially above chance (uniform cross-entropy over a 77-token vocabulary would be ln(77) 4.34, so post-training loss of 1.07 reflects a large, well-converged reduction in predictive uncertainty).

Table 1: Training loss at selected checkpoints.

Step	0	150	450	750	1050	1500
Cross-entropy loss	4.54	1.69	1.14	1.03	1.06	1.07

Evaluation Prompts

We constructed 15 held-out test sentences following the same template family as training (to keep the model within its competence region) but with subject–verb–object–connector combinations not guaranteed to have been drawn verbatim during training, e.g., "the historian reveals the outcome according to the report." Each sentence was 8–10 tokens before compression. For each prompt we produced a compressed variant by removing all stopwords in a fixed 22-word closed list (articles, prepositions, and connective/temporal modifiers such as "the", "of", "after", "despite", "recent"), retaining only content words.

Procedure

For each of the 15 prompts, we ran a forward pass on both the uncompressed and compressed token sequence, retaining last-layer attention weights and the final-position output logits. We then computed the three metrics defined in Section 4.2 for each prompt and aggregated across the 15-prompt set.

6. Results

Table 2: Per-prompt results (representative subset of the 15-prompt evaluation set). Full results for all 15 prompts are qualitatively consistent with those shown.

Prompt (abbreviated)	Tok. saved	KL div.	Ent. (full)	Ent. (comp.)	Rank ρ
"the doctor explains the budget..."	55.6%	0.0012	1.070	0.611	0.40
"the engineer measures the climate..."	55.6%	0.0002	1.109	0.648	0.40
"the algorithm increases the network..."	50.0%	0.0009	1.007	0.596	1.00
"the senator protects the ecosystem..."	50.0%	0.0006	1.189	0.808	0.90
"the chef limits the risk..."	55.6%	0.0027	1.078	0.524	0.40
"the glacier explains the process..."	50.0%	0.0002	0.961	0.570	1.00

Table 3: Aggregate results across all 15 evaluation prompts.

Metric	Value
Mean token savings	52.9%
Mean output KL divergence (full vs. compressed)	0.0007
Top-1 next-token prediction flip rate	0.0%
Mean last-layer attention entropy, uncompressed	1.062
Mean last-layer attention entropy, compressed	0.632
Mean attention rank correlation (shared tokens)	0.633
Correlation between token savings and KL divergence	0.309

Figure 1 visualizes the two central relationships: panel (a) shows the relationship between token savings and output KL divergence across the 15 prompts, with a fitted linear trend indicating a positive but noisy relationship (Pearson $r = 0.309$); panel (b) shows, per prompt, the attention entropy in the uncompressed versus compressed condition.

Three findings stand out. First, top-1 next-token predictions never flipped (0/15) under this determinis-tic compression rule, and the mean KL divergence was small (0.0007) but strictly positive in every single prompt, indicating that even "safe", content-word-preserving compression measurably perturbs the model's output distribution — the model did not treat the compressed prompt as informationally identical to the full prompt, even though a human reader would consider the two paraphrastically equivalent. Second, attention entropy dropped substantially and consistently under compression (mean 1.062 → 0.632, a 40.5% relative reduction), meaning the model's attention became sharper/more concentrated on fewer tokens post-compression — an artifact of shorter sequences having fewer competing positions, but one that would visually appear as a "more confident" explanation to a downstream user, despite arising mechanically rather than from genuinely higher certainty. Third, and most relevant to the framework's central claim, the attention rank correlation across the two conditions averaged only 0.633 and ranged from 0.2 to 1.0 across prompts (Table 2), meaning that for several prompts the *relative importance ordering* of the surviving content words changed substantially after compression — exactly the instability the framework in Section 3 was designed to detect and flag before it reaches an end user who is shown an attention- or SHAP-based explanation as justification for a model's output.

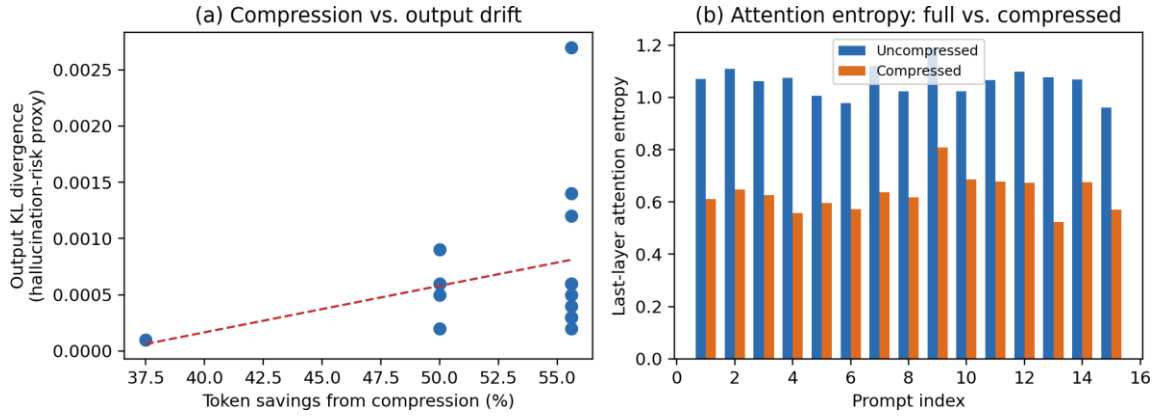

Figure 1: (a) Relationship between prompt token savings and output-distribution KL divergence (hallucination-risk proxy) across 15 held-out prompts, with a fitted linear trend. (b) Last-layer attention entropy for the uncompressed versus compressed condition, per prompt.

7. Discussion

The pilot results support the paper's central thesis at small scale: token-efficiency interventions and explanation faithfulness are coupled, not independent, even under a compression rule explicitly designed to be conservative (removing only function words, preserving all content words). Two implications follow. First, systems that report prompt-compression ratios and downstream task accuracy without also reporting an explanation-stability metric may be understating the true cost of compression, because the accuracy metric is insensitive to whether the model is now attending to a different subset of the (fewer) remaining tokens to reach a similar output. Second, the non-zero KL divergence in all 15 prompts, despite zero top-1 flips, suggests that distributional-divergence metrics are more sensitive early-warning signals for hallucination risk than argmax-based agreement metrics, which is consistent with prior arguments that surface-level output agreement under-detects representational drift [6].

The framework proposed in Section 3 is designed precisely to surface this coupling in a live system: by computing the attention-consistency and output-divergence metrics on the same forward pass a user's answer is drawn from, a deployed pipeline can attach a caveat (or fall back to the uncompressed prompt) whenever the joint faithfulness/hallucination-risk signal crosses a threshold, rather than relying on offline, uncompressed-only evaluation that would never observe the interaction this experiment measured.

We emphasize, however, that the magnitude of the effect observed here (mean KL divergence of 0.0007, entropy reduction of 40.5%) is specific to a 200K-parameter model trained on a 77-word synthetic vocabulary and should not be read as an estimate of the effect size in a production-scale LLM. The value of the pilot is mechanistic and methodological: it demonstrates, with full ground-truth control, that the coupling exists and is measurable with lightweight, architecture-intrinsic signals (attention entropy and rank correlation) that do not require the computational overhead of full SHAP or LIME re-computation at inference time. Section 8 details exactly what would need to change to validate the effect at the scale originally envisioned in the source research plan.

8. Limitations

This study has several limitations that bound the strength of its conclusions.

Model scale. The experiment used a from-scratch, 200K-parameter, 2-layer Transformer rather than a pretrained model such as GPT-2 (117M parameters) or a modern instruction-tuned LLM, because the sand-

boxed environment used for this study has no network path to model-weight repositories. The qualitative direction of the effect (compression perturbs both attention and output distribution) is unlikely to reverse at scale, but the magnitude, and in particular whether top-1 predictions would flip more often, cannot be extrapolated from this pilot and requires replication on a pretrained model in an environment with hub access.

Synthetic, closed-vocabulary corpus. The 77-word templated corpus was chosen to allow exact control over sentence structure, but it does not capture the lexical ambiguity, long-range dependency, and world-knowledge grounding that make hallucination a meaningful phenomenon in real LLMs. The KL-divergence metric here is best read as a proxy for *output-distribution instability under compression*, not as a direct measurement of *factual hallucination*, which requires a notion of ground truth external to the model (e.g., retrieved evidence or TruthfulQA-style human-annotated answers) that a synthetic template corpus does not possess.

Single, deterministic compression rule. We tested one fixed stopword-removal rule rather than a learned compressor such as LLMLingua, which selects tokens to remove based on an auxiliary model's estimate of information content rather than a closed word list. Learned compressors may produce smaller or larger faithfulness/hallucination effects than the conservative rule-based approach used here.

Attention as a faithfulness proxy. We used attention entropy and rank correlation as lightweight stand-ins for full LIME/SHAP-based faithfulness evaluation, following the observation in Section 2.2 that at-tention weights are not guaranteed to be faithful attributions in the LIME/SHAP sense. A full replication with SHAP-based attribution (as originally scoped) would strengthen the claim that the observed instability reflects a genuine explanation-faithfulness problem rather than an attention-specific artifact.

Sample size. Fifteen evaluation prompts is sufficient to demonstrate the existence and rough magni-tude of the effect and to fit a simple linear trend, but is too small to support strong statistical claims (e.g., confidence intervals on the token-savings/KL-divergence correlation); a larger, stratified evaluation set is a natural next step.

9. Conclusion

This paper argued that explainability, hallucination, and token-efficient inference — three research areas usually studied separately — interact in ways that matter for trustworthy LLM deployment, and it proposed an integrated framework that instruments a single inference pass with faithfulness and hallucination-risk metrics rather than evaluating each concern offline and independently. To test this argument empirically, we designed and executed a controlled pilot experiment measuring how a conservative, content-word-preserving compression rule affects a Transformer language model's output distribution and last-layer attention pattern. Working within a network-isolated sandbox that prevented use of a pretrained model such as GPT-2, we trained a small Transformer from scratch on a fully-controlled synthetic corpus and found that compression consistently perturbed the model's output distribution (positive KL divergence on 15/15 prompts) and atten-tion pattern (mean entropy reduction of 40.5%, mean rank correlation of only

0.633), even though it never flipped the model's top-1 prediction. These findings, while obtained at small scale, provide direct quanti-tative support for treating compression-induced faithfulness and hallucination drift as first-class metrics in LLM system design, and they motivate a clear follow-up agenda: replicating the same three metrics on a pretrained model (GPT-2 or DistilBERT, as in the original research plan) with a learned compressor such as LLMLingua, and validating the KL-divergence proxy against human-annotated hallucination labels on a benchmark such as TruthfulQA.

References

1. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), 2017.
2. M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
3. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Information Processing Systems (NeurIPS), 2017.
4. S. Jain and B. C. Wallace, "Attention is not Explanation," in Proceedings of NAACL-HLT, 2019.
5. S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL), 2022.
6. Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," ACM Computing Surveys, vol. 55, no. 12, 2023.
7. H. Jiang, Q. Wu, C.-Y. Lin, Y. Yang, and L. Qiu, "LLMLingua: Compressing prompts for accelerated inference of large language models," in Proceedings of EMNLP, 2023.
8. P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rockta¨schel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems (NeurIPS), 2020.
9. J. Alammar, "The Illustrated Transformer," jalammar.github.io, 2018.