Technical Summary: Synchronous Faithfulness Monitoring for Trustworthy Retrieval-Augmented Generation

Kai-Yu Lu

1 Research Problem and Motivation

Retrieval augmented language models (RALMs) combine large language models (LLMs) with external knowledge sources such as Wikipedia, web search, or tools to solve knowledge intensive tasks including open domain question answering and long form generation. Despite strong task performance, human evaluations show that RALMs frequently generate *unfaithful* content, that is, statements that contradict the retrieved context or are not grounded in any evidence.

Existing approaches address this reliability issue in two main ways. Post hoc attribution and revision methods attempt to check or repair the output after generation, but incur substantial computation and cannot influence the decoding process while it is happening. Synchronous interventions such as dynamic retrieval, likelihood based filtering, contrastive decoding, and Self-RAG style critique tokens act during decoding, but are mainly optimized for task accuracy rather than explicit faithfulness. These signals are not evaluated as sentence level faithfulness detectors and do not provide a way to control or guarantee the faithfulness level of the final response.

The paper therefore targets the following problem. There is a need for a lightweight, accurate, and synchronous mechanism that can (a) monitor the faithfulness of each generated sentence in real time during decoding and (b) use these signals to guide the decoding process toward outputs that are both informative and faithful to the retrieved context, with explicit control over faithfulness.

2 Related Work

2.1 Context Faithful Language Models

Context faithfulness studies how well LLMs respect retrieved evidence rather than relying solely on internal parametric knowledge. Prior work has identified over reliance on parametric knowledge when contexts contradict internal memories and has shown that even with improved knowledge verbalization, models can still prefer misleading or conflicting evidence.

Common strategies to improve context faithfulness include adapting the base LLM to context based generation, improving the quality of retrieved context, modifying decoding procedures, and performing post hoc detection or revision of hallucinations. Unfaithfulness to the context is closely related to context conflicting hallucinations.

Several detection methods use model features such as hidden states or confidence scores to identify hallucinations, while others rely on external lexical alignment models or entailment based checkers. However, these detectors are typically not designed to operate synchronously during decoding and are not integrated into the decoding algorithm itself.

2.2 Advanced Decoding for Retrieval Augmented Models

Recent decoding methods for RALMs aim to refine outputs by interacting with retrieval during generation. Iterative retrieval approaches repeatedly query external knowledge while generating. Contrastive decoding amplifies the influence of retrieved context in token level logits. FLARE uses token level likelihood thresholds to trigger dynamic retrieval. Toolformer style methods teach the model to call tools at appropriate points in the output. Self-RAG uses special critique tokens to score and rerank hypotheses generated under different retrieved documents.

These approaches improve accuracy and sometimes reduce hallucinations, but they do not provide explicit sentence level guarantees on faithfulness. They usually operate at token level or document level, and their signals are not systematically evaluated as faithfulness detectors on long form RAG benchmarks.

3 Dataset Construction

3.1 Benchmark Overview

The paper constructs a benchmark that covers four common long form retrieval augmented generation tasks:

- Question answering (QA).
- Summarization (Summ).
- Data to text generation (Data2txt).
- Biography generation.

Six datasets are used in total:

- QA, Summ, and Data2txt from the RAGTruth benchmark.
- FS (factscore benchmark) for biography generation.
- F-100 and F-100-anti, two new biography datasets created by the authors.

These datasets are combined to evaluate sentence level faithfulness detection and response level faithfulness oriented decoding for two base LLMs: Llama 2 7B Chat and Mistral 7B Instruct.

3.2 RAGTruth Based Tasks

For QA, Summ, and Data2txt, the paper uses the inputs and retrieved contexts from RAGTruth. The sources are:

- QA: MS MARCO, a large scale passage retrieval and question answering dataset.
- Summ: CNN/Daily Mail, a news summarization dataset.
- Data2txt: Yelp Open Dataset, where structured reviews are converted into free form text.

RAGTruth is designed for long form RAG settings, so questions and prompts encourage multi sentence answers that draw from multiple retrieved passages.

Dataset	#Train	#Test	Avg. context sent.	Avg. output sent.	% faithful sent.	% faithful inst.
QA	9 669	1 453	22.8	5.6	68.3	45.6
Summ	6 737	965	11.3	4.7	61.7	39.5
Data2txt	4 028	576	14.0	5.2	55.2	31.4
FS	0	1 003	8.1	5.3	55.0	28.2
F-100	0	100	29.0	7.4	76.7	53.0
F-100-anti	0	100	28.6	7.1	52.9	23.0

Table 1: Basic statistics of the datasets used in the benchmark for Llama 2 7B Chat. Numbers match the main paper and its appendix.

3.3 Biography Generation Tasks

The biography tasks focus on fact checking entity centric generations.

FS (Factscore Benchmark). FS is a benchmark where the model generates biographies based on structured or textual input, and factual claims in the biographies are evaluated by an automatic fact checking pipeline.

F-100 and F-100-anti. To create more challenging settings where unfaithful generations are likely, the paper introduces two new datasets:

- **F-100**: the model is asked to write biographies for 100 famous entities. The contexts are retrieved from Wikipedia using BM25 over the 2021 English Wikipedia dump, and the retrieved evidence is relevant to the target entity.
- **F-100-anti**: the same set of target entities is used, but the contexts are constructed by substituting evidence retrieved for a different entity. In other words, the context is intentionally mismatched with the target entity. This setting stresses the model's ability to resist generating parametric knowledge that conflicts with misleading contexts.

F-100 and F-100-anti query popular entities where parametric knowledge is strong, creating scenarios where models have to suppress confident but context inconsistent facts.

3.4 Train–Test Splits and Statistics

For QA, Summ, and Data2txt, the benchmark follows the train–test splits provided by RAGTruth. FS, F-100, and F-100-anti only have test splits. For SYNCHECK training, the authors use:

- QA, Summ, and Data2txt training sets to train task specific detectors for these three tasks.
- FS train labels to train detectors applied to F-100 and F-100-anti.
- F-100 train labels to train detectors applied to FS.

Table 1 summarizes the basic dataset statistics for Llama 2 7B Chat. The Mistral 7B Instruct statistics follow the same pattern with small numeric differences.

Here, the percentage of faithful sentences refers to the fraction of sentences in the outputs that are labelled as faithful, and the percentage of faithful instances refers to the fraction of outputs where all sentences are faithful.

3.5 Sentence Level Faithfulness Labels

Outputs are decomposed into sentences using the NLTK Punkt sentence tokenizer. The benchmark then assigns a binary faithfulness label to each sentence.

RAGTruth tasks. For QA, Summ, and Data2txt, the RAGTruth benchmark provides human annotated baseless spans and conflict spans. Any sentence that overlaps with these spans is marked as unfaithful. Sentences without such overlaps are labelled as faithful.

Biography tasks. For FS, F-100, and F-100-anti, the authors adopt an automatic pipeline:

1. A pre-trained propositionizer decomposes the output into decontextualized propositions (atomic factual statements).
2. For each proposition, an AutoAIS model checks whether it is supported by the retrieved context. This produces proposition level faithfulness labels.
3. A lexical matching algorithm maps proposition level labels back to sentence level labels by aligning propositions to their source sentences and propagating labels.

This pipeline yields sentence level ground truth labels for faithfulness tracking.

4 Query Protocol and Task Definitions

4.1 Retrieval Augmented Generation Setting

The paper considers retrieval augmented generation of free form long responses.

- Let x denote the input sequence encoding the question or instruction.
- Let c denote the retrieved contexts, a concatenation of multiple text segments from external sources.
- The base LLM M takes the concatenation $[x; c]$ and generates a sequence of segments $(s_1, \ldots, s_m)$, where each segment is a sentence in the experiments.

In this setting, each generated sentence should ideally be faithful to the context c, which means every factual claim in the sentence is supported by some part of the retrieved context.

4.2 Context Faithfulness Tracking Task

The *context faithfulness tracking* task is defined as follows. For each newly generated sentence s_i, a detector must assign a faithfulness label based on the input x, the retrieved context c, and optionally previously generated sentences $s_{1:i-1}$.

Many detectors output real valued scores rather than hard labels. The evaluation metric is the Area Under the Receiver Operating Characteristic curve (AUROC):

- AUROC measures how well the continuous scores rank faithful versus unfaithful sentences.
- A value of 1.0 indicates perfect ranking, 0.5 corresponds to random guessing, and values around 0.6 to 0.7 indicate weak discriminative ability.

The benchmark evaluates AUROC for each method across six datasets and two LLMs.

4.3 Faithfulness Intervention Task

The second task is *faithfulness intervention*, where the goal is to modify decoding so that the final response remains informative but has higher faithfulness. This is evaluated at the response level using two metrics:

- **Faithfulness**: the proportion of faithful propositions among all propositions in the response. A proposition is faithful if an automatic fact checker finds it supported by the retrieved context.
- **Informativeness**: the number of propositions in the response. Higher values indicate more content, while very low values may correspond to overly conservative or abstaining outputs.

Responses that are completely abstained or empty are excluded from faithfulness evaluation and are assigned informativeness 0.

5 Modeling Approach

5.1 Problem Formulation

The core modeling problem is to design a lightweight sentence level faithfulness detector that can run synchronously with autoregressive decoding, and then integrate that detector into a decoding algorithm that improves faithfulness with controllable trade offs.

The proposed solution has two components:

- **SYNCHECK**: a synchronous faithfulness monitor that aggregates multiple decoding time signals to classify each sentence as faithful or unfaithful.
- **FOD (Faithfulness Oriented Decoding)**: a two stage decoding scheme guided by SYNCHECK to backtrack from low quality sentences and explore alternative continuations.

5.2 SYNCHECK: Synchronous Faithfulness Monitoring

SYNCHECK is a feature based detector that observes real time decoding dynamics and outputs a faithfulness score for each sentence. It focuses on three types of failure:

1. The combined parametric and contextual knowledge is insufficient to answer the question.
2. The model fails to use the retrieved context in its predictions.
3. The model uses the context in an unfaithful way, such as misinterpretation or incorrect aggregation of evidence.

To capture these behaviors, SYNCHECK monitors four families of decoding time features for each sentence s_i.

Likelihood features. Low likelihood tokens often indicate knowledge gaps or off distribution outputs. SYNCHECK tracks:

- The minimum token probability in the sentence.
- The length normalized average token probability.

Intuitively, if the model assigns low probability to the tokens it produces, this suggests uncertainty or lack of support from both parametric and contextual knowledge, which is correlated with unfaithful behavior.

Uncertainty features. Predictive uncertainty reflects how confident the model is about its generated tokens. SYNCHECK uses:

- The average token level entropy within the sentence, which increases when the model is unsure between multiple candidate tokens.
- The local intrinsic dimension (LID) of intermediate layer activations for several layers, which is intended to measure how complex or unstable the representation manifold is at that point.

Higher entropy or larger LID values signal that the model finds the current token distribution difficult to predict, which often correlates with hallucinations.

Context influence features. A key failure mode is over dominance of parametric knowledge: the model ignores the retrieved context and answers based on memorized facts, which can conflict with or ignore the evidence in c.

To capture this, SYNCHECK compares two conditional distributions over the next sentence:

- With context: the distribution over tokens when decoding from $[x; c; s_{1:i-1}]$.
- Without context: the distribution over tokens when decoding from $[x; s_{1:i-1}]$.

The token level Kullback–Leibler divergence between these two distributions measures how much the retrieved context changes the model's predictions. If the divergence is small, the context has little influence, suggesting that the model may be ignoring the retrieved evidence. SYNCHECK aggregates this into features such as the mean KL divergence and the proportion of tokens with large KL divergence.

Semantic alignment features. Even if the context strongly influences generation and the model is confident, the output can still misinterpret or misrepresent the context. To catch such semantic misalignment, SYNCHECK runs an entailment based checker between each sentence s_i and the context c. The checker outputs a semantic alignment score that approximates how consistent the sentence is with the context. Lower alignment scores indicate likely unfaithfulness.

Feature aggregator. For each sentence s_i, the features from these four families form a compact feature vector. SYNCHECK uses a lightweight aggregator to map this vector to a scalar faithfulness score. The paper experiments with three hypothesis classes:

- Logistic regression (denoted $\text{SYNCHECK}_{\text{LR}}$).
- XGBoost decision trees (denoted $\text{SYNCHECK}_{\text{XGB}}$).
- A small multilayer perceptron (denoted $\text{SYNCHECK}_{\text{MLP}}$).

The aggregators are trained on small labelled sets derived from the sentence level ground truth labels described earlier. Importantly, the paper finds that these aggregators do not have to be heavily task specific or model specific; detectors trained on one task can often transfer to others, especially for QA.

5.3 FOD: Faithfulness Oriented Decoding

Faithfulness Oriented Decoding (FOD) uses SYNCHECK scores during generation to intervene in the decoding process. The algorithm has two stages.

Stage 1: Greedy search and backtracking.

1. Starting from an empty output, the base LLM generates sentences greedily given $[x; c]$.
2. After each sentence s_i is generated, SYNCHECK computes its faithfulness score f_i.
3. As long as f_i is above a backtrack threshold τ_1, the sentence is appended to the output prefix.
4. Once a sentence has a score below τ_1, generation stops and the algorithm backtracks to the previous sentence s_{i-1}, discarding the low faithfulness sentence and any later ones.

This yields a faithful prefix that already contains useful information, especially for summarization, data to text, and FS biography tasks.

Stage 2: Faithfulness guided beam search. Starting from the faithful prefix produced in Stage 1, FOD performs a beam search guided by SYNCHECK.

1. Initialise a beam set with the current prefix as the only beam.
2. At each step, for every beam:
   - Sample multiple candidate continuations using the base LLM.
   - For each candidate next sentence, compute its SYNCHECK score.
   - Discard any candidate whose score is below a sample pruning threshold τ_2.
   - For each surviving candidate, create a new beam by appending the candidate sentence to the beam's current prefix.
3. After processing all beams, select the top K beams with the highest aggregated faithfulness scores and repeat until at least one beam produces an end of sequence token or no new beams survive.
4. Return the beam with the highest aggregated faithfulness score as the final output.

In the experiments, the authors use $\tau_1 = 0.7$, $\tau_2 = 0.85$, beam size $K = 2$, and per step sample size $S = 6$. These hyperparameters are fixed across all tasks and both base models, which demonstrates that FOD does not require extensive per task tuning.

Baseline decoding strategies. To contextualize FOD, the paper evaluates several baselines:

- **Greedy**: standard greedy decoding without any intervention.
- **Abstention**: when $\text{SYNCHECK}_{\text{MLP}}$ predicts that any sentence in the response is below a threshold (0.7), the system refuses to answer or outputs only the faithful prefix.
- **Reranking**: sample multiple full responses and then choose the one with the highest average sentence level SYNCHECK score.
- **CAD**: a contrastive decoding method that adjusts token level logits to amplify the influence of the context while generating.

FOD differs from these baselines by using sentence level faithfulness scoring to prune unfaithful continuations during beam search and by providing a minimum faithfulness guarantee at sentence level.

6 Empirical Results

6.1 Context Faithfulness Tracking Performance

Table 1 in the paper reports AUROC for all context faithfulness tracking methods on six tasks and two base LLMs. The key observations are:

- Traditional RAG system quality control methods provide limited accuracy at sentence level. For example, CRITICTOK and FLARE often achieve average AUROC around 0.6, indicating weak ability to distinguish faithful from unfaithful sentences.
- SPANEXTRACT and lexical alignment models such as ALIGNSCORE and MINICHECK show mixed generalization. SPANEXTRACT performs reasonably on tasks closer to its training data but degrades on biography generation tasks. ALIGNSCORE performs well on QA, Summ, and biography tasks, where its pretraining distribution is similar, but fails to generalize to the Data2txt task that requires fine grained checking of numeric and location details.
- $\text{SYNCHECK}_{\text{MLP}}$ achieves the strongest average performance across all tasks for both Llama 2 7B Chat and Mistral 7B Instruct. The average AUROC is approximately 0.831 for Llama 2 and 0.867 for Mistral.
- Despite using ALIGNSCORE as one of its semantic alignment features, SYNCHECK surpasses ALIGNSCORE on most tasks, showing that combining model centric features with lexical alignment yields a stronger detector than either type alone.
- Logistic regression based $\text{SYNCHECK}_{\text{LR}}$ already performs competitively, which suggests that the feature set itself is highly informative and can be exploited by simple aggregators.

6.2 Faithfulness Oriented Decoding

Table 2 of the paper reports faithfulness and informativeness for different decoding strategies on all tasks. The following patterns emerge:

- Compared to greedy decoding, FOD with backtracking only (FOD (BT)) already substantially increases faithfulness on all tasks while keeping a moderate level of informativeness.
- Compared to abstention, FOD (BT) improves both faithfulness and informativeness. Abstention often yields highly faithful but very short outputs with few propositions, while FOD retains more content by keeping a faithful prefix.
- FOD (Full), which includes beam search, further increases informativeness with little or no loss in faithfulness. This means that the algorithm successfully finds alternative faithful continuations beyond the initial prefix.
- CAD improves faithfulness for QA, Summ, and Data2txt but is ineffective for biography generation tasks. In contrast, FOD consistently outperforms CAD across all six tasks and both LLMs, except for summarization with Mistral where they perform similarly.
- Compared to sampling and post hoc reranking, FOD significantly improves faithfulness while using the same number of sampled candidates, indicating the benefit of on the fly monitoring and pruning of unfaithful samples.

The paper also examines faithfulness as a function of prefix length (Faithfulness@L) and shows that FOD consistently yields higher faithfulness than both greedy decoding and CAD when outputs are truncated to a fixed number of sentences.

6.3 Ablation and Generalization Analyses

The analysis section provides additional insights.

Feature ablation. Ablation experiments remove one feature family at a time from SYNCHECK$_{\text{MLP}}$. The results show:

- Removing any feature family harms AUROC for both Llama 2 and Mistral.
- Semantic alignment is the most influential feature family, but it cannot achieve the best performance alone.
- Context influence is the second most important; removing it causes an absolute AUROC drop of about 0.02 to 0.03, underscoring its unique role in detecting context ignoring behavior.

Cross task faithfulness tracking. The authors train SYNCHECK$_{\text{MLP}}$ on one task and evaluate on another. The results indicate:

- Detectors trained on one task often transfer reasonably well to others.
- In particular, most tasks transfer well to QA, suggesting that QA faithfulness signals are robust.
- This reduces the need for task specific labelled data when deploying SYNCHECK in new domains.

Cross model faithfulness tracking. The detector trained on one model can be applied to other models, such as different sizes of Llama or Mistral. The average AUROC remains high (around 0.82 to 0.86), indicating that the monitored features generalize across model architectures.

7 Summary

7.1 Novelty and Differences from Prior Methods

The paper introduces several conceptual and practical contributions:

- A synchronous, sentence level faithfulness monitor, SYNCHECK, that combines model centric decoding features and semantic alignment signals. Previous approaches typically relied on a single type of signal, such as likelihood or lexical alignment, and were not systematically evaluated as faithfulness detectors.
- A two stage faithfulness oriented decoding algorithm, FOD, that uses SYNCHECK scores to backtrack from low faithfulness sentences and guide beam search toward more faithful continuations. In contrast, prior interventions like FLARE or CAD operate mainly at token level and do not provide explicit guarantees on the faithfulness of individual sentences.
- A comprehensive sentence level benchmark for faithfulness tracking across six long form RAG tasks and two base LLMs, including new biography datasets F-100 and F-100-anti that stress parametric knowledge dominance.

- Empirical evidence that simple aggregators applied to carefully designed decoding time features can achieve strong and transferable faithfulness detection performance, enabling lightweight deployment.

7.2 Limitations and Future Directions

The study also reveals several limitations:

- SYNCHECK relies on labelled sentence level data for training its aggregator. Although cross task and cross model generalization is strong, some labelled data are still required, especially for new domains with different types of unfaithfulness.
- The semantic alignment features use an existing entailment model and lexical alignment scores. Errors in these auxiliary models can propagate to SYNCHECK and may limit faithfulness detection accuracy in domains far from the entailment model's training distribution.
- The fact checking pipeline for faithfulness and informativeness uses automatic propositionizers and fact checkers, which may introduce noise into ground truth labels. This affects both training and evaluation.
- FOD incurs additional decoding cost due to backtracking and beam search. Although the beam size and sample size are small, this method is more expensive than greedy decoding, and a systematic latency–accuracy trade off analysis is not fully explored.

Future work can investigate unsupervised or self supervised training of the SYNCHECK aggregator, tighter integration with retrieval, and more efficient decoding strategies that approximate FOD with lower computational cost. Another direction is to extend the framework to multi modal RAG settings and to more complex discourse structures beyond sentence boundaries.

7.3 Three Key Takeaways

1. Real time faithfulness monitoring using a combination of likelihood, uncertainty, context influence, and semantic alignment features enables accurate sentence level detection of unfaithful content in long form RAG.
2. Faithfulness Oriented Decoding guided by these signals can significantly improve response level faithfulness while preserving or even enhancing informativeness, outperforming abstention, reranking, and contrastive decoding baselines.
3. Carefully designed decoding time features generalize well across tasks and models, making SYNCHECK and FOD promising building blocks for trustworthy retrieval augmented generation systems.