HALLUCANA: Fixing LLM Hallucination with A Canary Lookahead

Tianyi Li   Erenay Dayanik   Shubhi Tyagi   Andrea Pierleoni
Amazon Alexa AI, Cambridge, UK
{tylteddy, erenay, tshubhi, apierleo}@amazon.co.uk

Abstract

In this paper, we present HALLUCANA, a canary lookahead to detect and correct factuality hallucinations of Large Language Models (LLMs) in long-form generation. HALLUCANA detects and intervenes as soon as traces of hallucination emerge, during and even before generation. To support timely detection, we exploit the internal factuality representation in the LLM hidden space, where we investigate various proxies to the LLMs' factuality self-assessment, and discuss its relation to the models' context familiarity from their pre-training. On biography generation, our method improves generation quality by up to 2.5x, while consuming over 6 times less compute.[1]

1 Introduction

The rise of Large Language Models (LLMs) has seen their increasing presence in people's daily lives. Factuality hallucination, a phenomenon where LLMs assertively generate factually unfaithful content (Ji et al., 2023; Tonmoy et al., 2024), severely damages users' trust, and blocks their application in critical use cases.

Specifically, for long-form knowledge-intensive LLM responses, mitigating factuality hallucinations w.r.t. world knowledge has attracted increasing attention from the community. Existing approaches to factuality hallucination detection, such as SelfCheckGPT (Manakul et al., 2023) or Varshney et al. (2023); Mündler et al. (2024), require assistive generations (samples, drafts, etc.), leading to high computational footprint and latency.

In this paper, we propose HALLUCANA, **a novel knowledge-free canary lookahead**, which identifies potential hallucinations selectively and as soon as traces emerge. HALLUCANA is based on *lookahead*, a recent decoding strategy which manipulates

generations ad-hoc at the token level, such that repeated re-drafting is avoided. First introduced by Lu et al. (2022) to enforce lexical constraints, lookahead has been used to improve faithfulness in abstractive summarization (Wan et al., 2023).

Lookahead evaluates prospective continuations with a scorer, and uses these auxiliary scores to amend the LLM logit scores, so that generation is steered toward desired directions. For abstractive summarization, prior work (Wan et al., 2023) has calculated lookahead scores as entailment between prospective continuations and source documents, where *entails* means *not hallucinated*. However, for long-form LLM generation w.r.t. world knowledge, no reference documents are directly available.

To overcome this issue, we extract a factuality representation from the LLMs' internal hidden space, as our lookahead scorer. Following prior work in open-domain QA and factoid statement verification (Kadavath et al., 2022; Azaria and Mitchell, 2023), we train faithfulness classifiers over LLM hidden states, as our lookahead scorer to predict the factuality of continuations.

To ensure generalizability, we explicitly avoid in-domain training sets, and instead train the classifiers on out-of-domain datasets (Joshi et al., 2017; Kwiatkowski et al., 2019), then use them off-the-shelf as lookahead scorers in long-form LLM generation. Note that we also avoid the complexity and overhead of retrieving external knowledge, so that our approach applies to low-resource domains where reliable knowledge sources are unavailable.

Relevant to using faithfulness classifiers as lookahead scorers, we introduce three innovations to the lookahead. First, since our lookahead inputs are merely hidden states, which do not rely on textual completions, we can predict LLM response factuality pre-hoc, before generation begins.

Second, as faithfulness classifiers are parametric proxies, we observe excessive noise when the classifier is applied at every time step. To address the

---

[1]Our code are available at https://github.com/amazon-science/HALLUCANA-Fixing-LLM-Hallucination-with-A-Canary-Lookahead.

noise and improve efficiency, we identify ***critical time steps*** during decoding, so that lookahead is only selectively applied when the LLM is at a cross-road to generating diverse continuations, which are also when hallucination tends to emerge.

For instance, in *King Charles was born in the Buckingham Palace*, it is unambiguous to generate *Palace* after *Buckingham*. Thus, by identifying critical time steps (e.g. *born*, *Buckingham*) and ignoring mere continuations (e.g. *in*, *Palace*), we reduce noise while retaining hallucination detection ability. We use a logit entropy heuristic to identify critical time steps in decoding, which improves both generation quality and efficiency.

Third, at critical time steps, we observe that branches with high LLM logit scores and low lookahead scores are often still selected during generation, which leads to hallucinations. Without eclipsing logit scores with too much weight on lookahead scores, we additionally introduce a ***veto mechanism***, where branches with very low faithfulness scores are removed, and the choice among remaining branches is left to moderately amended scores.

We conduct evaluation on a long-form generation benchmark generating people's biographies (Min et al., 2023). We show that HALLUCANA, exploiting LLM internal factuality representation, is robust and grounded in the LLMs' parametric memory of their context familiarity. In addition to training with accuracy labels, we also train faithfulness classifiers using LLMs' context familiarity from pre-training as supervision labels. We find that both classifiers predict LLM factual faithfulness in long-form generation comparably well, and that both classifiers produce highly correlated predictions. On the end-to-end evaluation, HALLUCANA improves LLM generation quality by up to 2.5x, outperforming SOTA baselines while consuming over 6 times less compute.

In summary, our contributions include:

- we propose HALLUCANA, a light-weight canary lookahead, exploiting the internal factuality representation of LLMs for hallucination detection;

- we apply the lookahead selectively at critical time steps, and introduce the veto mechanism in addition to scoring;

- we show that this internal factuality representation is grounded in the LLMs' parametric

memory of context familiarity, and is robust across natural language tasks.

2   Related Work

In abstractive summarization, factuality hallucination has been studied as the inconsistency between summaries and source documents (Wan et al., 2023), and for QA as answer accuracy (Kadavath et al., 2022; Li et al., 2023). Maynez et al. (2020) made the distinction between faithfulness w.r.t. source knowledge, and factuality w.r.t. world knowledge. As we are concerned with long-form generation with no specific source (Min et al., 2023; Manakul et al., 2023; Varshney et al., 2023), we simply define hallucination against factuality.

Hallucination detection can be classified into black-box, grey-box, and white-box approaches. Black-box approaches (Lin et al., 2022; Manakul et al., 2023; Varshney et al., 2023; Lin et al., 2023; Mündler et al., 2024; Madaan et al., 2023; Peng et al., 2023) only use LLM generated tokens, which apply to closed-source LLMs but require computationally expensive assistive generations. Grey-box approaches (Kuhn et al., 2023) additionally use logit distributions to help identify hallucinations.

White-box approaches (Kadavath et al., 2022; Azaria and Mitchell, 2023; Li et al., 2023; Hernandez et al., 2023) develop transformations and classifiers on LLM hidden states, which requires the decoding process to be transparent (i.e. white-box); on the other hand, since LLM hidden states, as the source of both logit distributions and the textual outputs, encodes more fine-grained information, the white-box approach provides a chance to detect hallucinations earlier and more efficiently. Our method is also a white-box approach.

When hallucination is detected, several types of mitigation techniques have been used. Kadavath et al. (2022) rejects questions with hallucinatory answers; on tasks with fixed answer sets, Wang et al. (2023) samples reasoning paths multiple times and takes the most consistent answer; Mündler et al. (2024) prompts the LLM to delete hallucinatory content and rewrite relevant sentences; Varshney et al. (2023); Madaan et al. (2023) also re-writes with retrieved knowledge and self-critique, respectively; Li et al. (2023); Hernandez et al. (2023) apply hidden state transformations to generate more factual tokens; Chuang et al. (2024); Wan et al. (2023); Lee et al. (2022) manipulate the LLM decoding process directly to select more factually
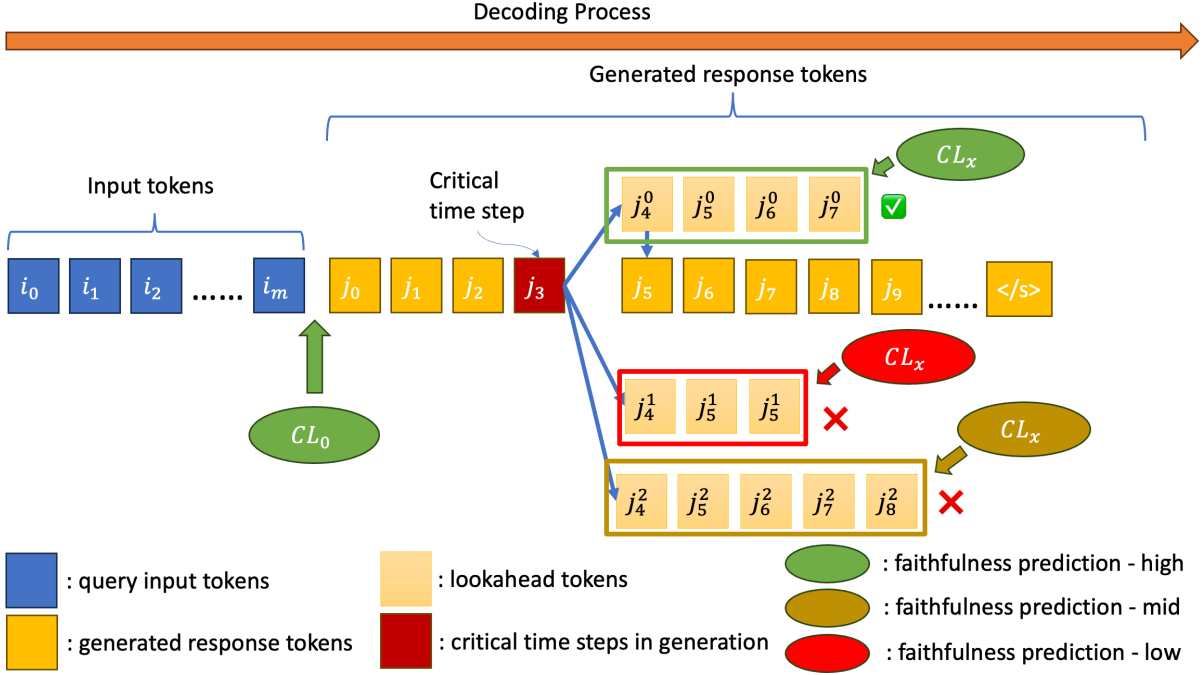
Figure 1: Diagram illustration of HALLUCANA in action.

faithful next-tokens.

Particularly, in abstractive summarization, Wan et al. (2023) uses a decoding strategy called lookahead, first introduced in Lu et al. (2022), to reduce hallucination. We also use a lookahead, but exploit LLMs' internal factuality representation to extract faithfulness lookahead scores instead of external entailment judgements from reference documents. More generally in LLM research, lookahead mechanism has been viewed as a decoding-time alignment strategy to incorporate reward signals, in the form of lookahead scores, into the decoding process (Huang et al., 2024).

The LLMs' internal factuality representation has also been discussed in prior work. Kadavath et al. (2022); Azaria and Mitchell (2023) showed that there exists a representation in the LLM hidden states, which encodes whether a statement or an answer is accurate or not. Yin et al. (2023) found that LLMs also possess the ability to detect unanswerable questions. He et al. (2024) studied patterns in LLM hidden states when generating faithful or hallucinatory answers, using a methodology inspired by physiological lie detectors. On the other hand, Levinstein and Herrmann (2023) argues that such representation can be brittle to simple dataset variations; Orgad et al. (2024) also shows that this truthfulness representation is different across distinct tasks. Kossen et al. (2024) learnt this truthfulness representation by training on a proxy semantic en-

tropy signal, and observed improvements on generalisabilty. In this paper, we compare between truthfulness representations learnt from corpus-based familiarity and on accuracy, and establish that for representations learnt from accuracy signals, the generalisable portion of their performance is also grounded in the LLMs' context familiarity.

For more related work on LLM hallucination, we refer the readers to relevant surveys (Ji et al., 2023; Tonmoy et al., 2024).

On a separate thread, Kandpal et al. (2023); Sun et al. (2023) have discussed the relation between LLM QA accuracy and the frequency/popularity of involved entities. In this paper, we show that the LLMs are self-aware of their familiarity with contexts, with this information encoded in the LLMs' hidden states; we also show that this awareness is a strong contributor to their factuality awareness.

3 HALLUCANA

In this section, we introduce our canary lookahead in LLM decoding, as graphically illustrated in Figure 1. In the below, we first introduce our lookahead architecture in §3.1, then introduce the classifiers in §3.2, where we also discuss the source of factuality classification abilities in LLMs.

3.1 The Canary Lookahead

Our canary lookahead is designed to reduce hallucination in long-form natural language generation,

it consists of two phases: CL_0, a pre-hoc scorer, which is applied before generation begins (after i_m and before j_0 in Fig. 1); CL_x, an ad-hoc scorer, which is applied during generation.

When generating a response following an input $i_{0...m}$, we begin by applying the CL_0 scorer. CL_0 takes the hidden state $hs(i_m)$ generated after encoding the last input token, and predicts the factuality of the not-yet-generated response for input $i_{0...m}$:

$$CL_0(i_{0...m}) = classifier_0(hs(i_m))$$

When $CL_0(\cdot) < \tau_0$, decoding is not initiated, and an abstained response is returned.

Otherwise, when decoding is initiated, we listen for critical time steps to apply the CL_x scorer. Critical time steps are decoding steps when the LLM faces uncertainty regarding its subsequent generation. We identify these critical time steps by examining the entropy of the LLM logit score distributions. For those steps j_{crit}, where entropy exceeds a threshold τ_{crit}, CL_x is applied:

$$lookahead(j_{crit}^k) = hs(decode^{N-1}(j_{crit}^k))$$
$$CL_x(j_{crit}^k) = classifier_x(lookahead(j_{crit}^k))$$

i.e. we create a lookahead branch for each of the top-K likely next token at the critical time step j_{crit}^k, where we continue greedy decoding ($decode(\cdot)$) for N-steps[2] to allow for a statement to be completed. Then, we take the hidden states at the last lookahead token, $lookahead(j_{crit}^k)$, and apply $classifier_x(\cdot)$ to predict the faithfulness of this lookahead statement.

We *veto* the branches with $CL_x(\cdot) < \tau_x$, then take weighted geometric mean between the lookahead score ($CL_x(\cdot)$) and the LLM logit score ($logit(\cdot)$) for the remaining branches to select the next token:

$$scr(\cdot) = \begin{cases} -\infty, & CL_x(\cdot) < \tau_x \\ logit(\cdot) \times CL_x(\cdot)^{\alpha}, & CL_x(\cdot) \geq \tau_x \end{cases}$$
$$k_{selected} = argmax_{k \in K}(scr(j_{crit}^k))$$

where $logit(\cdot)$ is the LLM logit score, α is the weight term. The branch with the highest overall $scr(\cdot)$ is selected. In case all K options are *vetoed*, i.e. $\forall k \in K, scr(t^k) = -\infty$, we simply assign period (.) as the next token to avoid hallucination.

3.2 Training Faithfulness Classifiers

We now introduce the *classifier_0* and *classifier_x* classifiers corresponding to the two scorers. While these classifiers are used for long-form generation hallucination detection during inference, we specifically train them on out-of-domain short-form QA datasets to ensure generalisability. For *classifier_0*, we extract hidden states at the last question tokens as training input; for *classifier_x*, we use hidden states at the last answer tokens as training input, reflecting the positions of application during inference.

We explore two types of supervision labels for training classifiers: QA accuracy (Lbl_{acc}) and context-familiarity from corpus (Lbl_{corpus}).

QA Accuracy Following prior work (Kadavath et al., 2022), we train faithfulness classifiers using string match accuracy of LLM predictions on QA datasets. This is a simple criterion of faithfulness: when the LLM-predicted answer matches the gold answer, in the LLM's internal factuality representation the hidden state should also encode "factual". When the predicted answer matches a gold answer by string match, $Lbl_{acc} = \mathbb{1}$ (factual) is assigned; otherwise, $Lbl_{acc} = \mathbb{0}$ (hallucinatory) is assigned.[3]

Context Familiarity in Corpus Especially for *classifier_0*, where no responses or answers are yet specified to be evaluated, and for *classifier_x* as well, we show that the classifiers' ability to detect the factuality of **their own generations**, is grounded in LLMs' representation for their context familiarity.

Concurred with prior work (Kuhn et al., 2023), we notice that hallucination is associated with LLMs' uncertainty about a statement, which can be traced back to their unfamiliarity with the statement, or more broadly, the statement's context.[4]

Kandpal et al. (2023) has found that the LLMs' QA accuracy is positively correlated with the frequency of the contexts in LLM pre-train corpora, which embodies context familiarity.

---

[2]The initial token for each branch counts as the first decoded token, so a total of $N - 1$ additional decoding steps are applied.

[3]See Appendix B.1 for prompts and match metric; we also experimented with NLI-based factuality labels, but did not see improvements.

[4]Uncertainty is connected with entropy, however, in long-form generation, high entropy can come from e.g. the uncertainty of either the statement itself or which topic to discuss, so it cannot be used for hallucination detection.

Inspired by the above, we hypothesize that: 1) from their pre-training, LLMs develop a context familiarity representation in their hidden space; 2) this representation is exploited as a major contributor to the factuality representation, and thus the classifier performance, especially in pre-hoc predictions ($classifier_0$).

Based on the hypothesis, directly training on context familiarity should yield comparable or even superior performance. To verify this, we train classifiers using corpus frequency labels (Lbl_{corpus}), which approximates context familiarity. Importantly, with Lbl_{corpus}, we eliminate all traces of factual accuracy from the training process.

Following Kandpal et al. (2023), we use entity combinations to disambiguate contexts; we count their frequencies in LLM pre-train corpora as an upper-bound estimate of LLMs' familiarity with the respective contexts. For instance, whenever <*Barack Obama, US president, 2008*> are mentioned together, it likely locates the context of *Barack Obama running for office in 2008 and being elected US president.*

For each passage in the corpora, we extract named entities using ReFinED entity linker (Ayoola et al., 2022), and count context mentions by **consecutive** N-entity spans. Then, for each QA pair in the training sets, we similarly build N-entity spans and look up their frequencies in the pre-train corpora. We take geometric mean over these frequencies, as the context-familiarity label.

Notably, we use these frequencies not directly at inference time, but as labels for training. We afford to not assign labels for some entries in the training sets so that noise is reduced for the remaining labels: we restrict N-entity counts to only adjacent mentions, and use 3-entity spans instead of entity pairs (see Appendix B.2 for a comparison).

4 Experiment Setup

Models In this paper, we focus on two well-known and fully open-source LLMs: Falcon-7b-instruct (Almazrouei et al., 2023) and Flan-UL2 (Chung et al., 2022; Tay, 2023). Falcon-7b-instruct is a strong decoder-only LLM, which makes it especially good at text generation; on the other hand, Flan-UL2 is a strong encoder-decoder LLM widely used for classification tasks.

For the faithfulness classifiers, we use simple 3-layer MLP models, using the last-layer hidden states as inputs. For all our experiments, a frozen falcon-7b-instruct model is used to produce these input hidden states, the LLM parameters are not involved in the training of classifiers. We refer readers to Appendix C for detailed hyper-parameter configs.

Datasets Following prior research (Min et al., 2023; Manakul et al., 2023; Varshney et al., 2023), we assess the effectiveness of our canary lookahead in the context of long-form constrained generation, and conduct an evaluation focused on biography generation. Specifically, we use the factscore dataset (Min et al., 2023): for our development set, we use their "labelled" subset, comprising 183 entities, while their "unlabeled" subset, containing 500 entities, serves as our test set for reporting the final results.

In order to train the faithfulness classifiers, we use two popular generic open-domain QA datasets, Trivia QA (Joshi et al., 2017) and Natural Questions (Kwiatkowski et al., 2019), expanding the token-level answers to propositional statements for stylistic consistency.[5] These QA datasets are out-of-domain for the factscore dataset, involving diverse types of question-answer pairs, with less than 0.3% entity overlap, which we argue is consistent with the natural distribution of text.

When computing corpus frequencies for the context familiarity labels (Lbl_{corpus}), we use the respective pre-training corpora for each LLM: for Falcon, we extract frequencies from the falcon-refinedweb corpus (Penedo et al., 2023); for Flan-UL2, we use the C4 corpus (Raffel et al., 2020).

Hyper-Parameters We have introduced 4 hyper-parameters in our canary lookahead method: the lookahead length N, the number of explored branches K, and the veto thresholds for CL_0 (τ_0) and for CL_x (τ_x). The lookahead length N is anecdotally set to 8, as a trade-off between allowing atomic statements to finish and avoiding excessive computational cost.[6] The number of branches is set to $K = 3$ as a minimum default value, where we already observe substantial gains. On the other hand, the two veto thresholds are empirically selected to maximize development-set F-scores along the F-R curves (to be elaborated in the below). Specifically,

---

[5]For instance, "Q: Where is the capital of France? A: Paris." is expanded to: "Q: Where is the capital of France? A: The capital of France is Paris." See Appendix A.1.

[6]When sentence-end is met before looking N tokens ahead, the lookahead decoding also stops since the statement has already finished.

for CL_x scorers, since *veto* is repeatedly applied through a decoding run, we assign more benign τ_x maximizing F-1.2; for CL_0 and baselines, F-1 is maximized.

Baselines We compare our approach against various baselines: 1) vanilla LLMs with greedy and beam decoding (using beam size of 4 for comparable computational footprint); 2) *SelfCheckGPT* (*w/NLI* variant, as recommended) (Manakul et al., 2023), current SOTA of hallucination detection; 3) MAX($-logp$), used as a strong baseline in Manakul et al. (2023).[7] For *SelfCheckGPT*, sample responses are generated for reference, then response factuality is predicted using its consistency with the generated references. For MAX($-logp$), we calculate the minimum logit probability of generated tokens over a piece of response, and use it as a baseline measure of the LLMs' self-assessed uncertainty.

Metrics To compare HALLUCANA against baselines, we measure the factuality of generated responses. We use an automatic measure, FActScore (FS) (Min et al., 2023), which judges whether each atomic fact in the response is supported by a gold KG.[8] For each query-response pair, FS produces a floating point value between [0, 1], which denotes the ratio of supported atomic facts in the response.

For **evaluating classification performance** (§5.1), since the FS labels are non-binary, we are unable to use the P-R curves. Instead, we introduce the FS-Rejection-curve, similar to the Accuracy-Rejection-curve in Lin et al. (2023) (see Appendix E.2 for graphic explanations). We calculate *AUC-FR*, Area Under the FS-Rejection Curve, as our classification metric; we normalize the values between random classifiers $AUC\text{-}FR_{norm}(rand) = 0$ and perfect classifiers $AUC\text{-}FR_{norm}(pft) = 1$. Intuitively, $AUC\text{-}FR_{norm}$ measures the ratio of performance realized by the classifier, relative to its theoretical upper bound:

$$AUC\text{-}FR_{norm} = \frac{AUC\text{-}FR - AUC\text{-}FR(rand)}{AUC\text{-}FR(pft) - AUC\text{-}FR(rand)}$$

For **hallucination reduction performance** (§5.2), we use FActScore for the not-abstained responses, as the factuality metric. In addition, we measure informativeness, using the number of atomic facts (*#Facts$_d$*) in each response d, and responsiveness, using the ratio of not-abstained responses (*% respond*).[9]

We combine these aspects of generation quality with a *Generation Quality Index* (GQI):

$$gqi_\gamma(d) = (FS_d - \gamma) \times \textit{\#Facts}_d$$

$$GQI_\gamma = \frac{\sum_{d \in D} gqi_\gamma(d)}{|D| \times \sqrt{\textit{response rate}}}$$

GQI reflects our belief in response values: 1) responses with $FS_d < \gamma$ are negative assets, thus the $FS_d - \gamma$ term (we empirically set $\gamma = 0.2$ for reporting results in §5); 2) longer responses for less queries are preferred over short responses for more queries, thus the $\sqrt{\cdot}$. We use GQI as a holistic metric for straightforward comparison, but we also provide individual metrics for finer-grained inspections. We include a detailed analysis of the GQI_γ metric in Appendix E.3.

Apart from generation quality, we also measure the computational footprint, using the average number of tokens processed by the LLM, for an atomic fact to be generated in the final response, namely, *Tokens per Fact* (TpF):

$$TpF = \frac{\sum_{d \in D} (\lambda * |seq_d^{enc}| + |seq_d^{dec}|)}{\sum_{d \in D} \textit{\#Facts}_d}$$

where the $0 < \lambda < 1$ reflects that token encoding cheaper than token decoding. Specifically, we set $\lambda = 0.5$, which is aligned with the pricing ratio with Open AI (https://openai.com/pricing, visited Jan 2, 2024). For simplicity, we ignore the computational footprint our classifiers, which are light-weight and called only once per explored branch, as well as smaller NLI models in *SelfCheckGPT* baselines.

5 Results and Discussions

We report results on the stand-alone classification task in §5.1, and end-to-end on long-form generation in §5.2. In the main tables are results with Falcon-7b-instruct, we leave results with Flan-UL2 to App. D, where conclusions are consistent.

---

[7]Apart from these, Varshney et al. (2023) is also relevant to our work. We exclude this baseline for two reasons: 1, it assumes the availability of external knowledge; 2, no implementation is available for its reproduction.

[8]We use a SOTA Mistral LLM (Jiang et al., 2023) in FActScore calculation, which yields consistent results as the original implementation, see Appendix E.1 for details.

[9]We count a response as *abstained*, when it starts with "sorry", or is shorter than 16 words, or is predicted as *hallucinatory* at CL_0.

Method	Phase	$AUC\text{-}FR_{norm}$
$MAX(-logp)$	end	36.3
SelfCk (Manakul et al., 2023)*	end	52.6*
$classifier_0\text{-}Lbl_{acc}$ (Kadavath et al., 2022)	CL_0	44.2
$\textbf{classifier}_0\textbf{-}\textbf{Lbl}_{\textbf{corpus}}$	$\textbf{CL}_0$	**48.1**
MAX$(-logp)$	CL_x	10.4
SelfCk (Manakul et al., 2023)*	CL_x	48.8*
$\textbf{classifier}_{\textbf{x}}\textbf{-}\textbf{Lbl}_{\textbf{acc}}$ (Kadavath et al., 2022)	$\textbf{CL}_{\textbf{x}}$	**37.5**
$classifier_x\text{-}Lbl_{corpus}$	CL_x	23.1

Table 1: Classification performance for falcon-7b-instruct LLM, measured by % of $AUC-FR_{norm}$ on greedy generations for **factscore dev set**. *SelfCk*(2023)* requires generating extra reference responses, thus the asterisk. CL_0 means prediction is made before decoding, CL_x means during decoding at sentence-ends, end means after generation ends.

Method	$GQI_{0.2}\uparrow$	Avg. FS (%) $\uparrow$	Avg. #Facts $\uparrow$	% respond $\uparrow$	$TpF\downarrow$
greedy	0.68	25.4	15.2	100%	7.8
beam-4	1.01	29.9	11.7	94.2%	53.4
$MAX(-logp)$	1.07	29.7	14.4	69.1%	12.0
SelfCk* (2023)	1.36	32.6	15.3	66.9%	298.5
HALLUCANA					
CL_0	1.17	30.3	15.1	71.7%	7.9
CL_x	1.34	30.9	14.0	75.4%	51.6
$\textbf{CL}_0 + \textbf{CL}_{\textbf{x}}$	**1.69**	**36.0**	**14.4**	**56.3%**	**47.1**
$(CL_x)_{corpus}$	1.07	30.8	10.5	94.2%	48.2
$(CL_0 + CL_x)_{corpus}$	1.46	33.6	12.5	57.3%	46.5

Table 2: Hallucination reduction performance for falcon-7b-instruct on **factscore test set**. For strict compute budgets, CL_0 is good for preventing hallucination; generally, full HALLUCANA ($CL_0 + CL_x$) (**bolded**) is recommended.

5.1 Faithfulness Classifiers

In Table 1, we report our faithfulness classifier's stand-alone classification performance, and compare it with baselines described in §4.

We measure how well classifier predictions align with FActScore labels on the LLM greedy generations on factscore dev set. We evaluate each classifier against what they predict: we evaluate $classifier_0$ classifiers and end baselines against the factuality of full responses, and $classifier_x$ against that of the current sentences ending with the classifier input tokens.

In general, faithfulness classifiers (*clsf-*), trained on QA datasets, generalize well to hallucination detection in long-form generation: they exhibit non-trivial performance, achieving up to half the performance of a perfect classifier, consistently outperforming the logit baseline ($MAX(-logp)$) (48.1 vs. 36.3; 37.5 vs. 10.4). Especially in the CL_0 phase, they achieve results comparable to the far more expensive *SelfCk* baseline, without requiring tokens to be decoded (CL_0 vs. end).

Different training signals are favoured in CL_0 and CL_x phases. CL_0, the pre-hoc phase, depends more strongly on context familiarity; as expected, training directly on context familiarity (Lbl_{corpus}) yields superior results to QA accuracy (Lbl_{acc}) (48.1 vs 44.2). This supports our hypothesis in §3.2, that when we train on QA accuracy labels, we end up learning a similar representation as when training on context familiarity labels.

Moreover, predictions from classifiers trained on the two signals are also strongly correlated, with Spearman's $\rho(Lbl_{acc}, Lbl_{corpus}) = 0.63$, and p-value 1.4e-21. For comparison, between predictions from the *SelfCk* baseline and these two classi-

fiers, this metric are $\rho(SelfCk, Lbl_{acc}) = 0.37$ and $\rho(SelfCk, Lbl_{corpus}) = 0.31$, respectively, both much lower.

On the other hand, for CL_x, the ad-hoc phase, $classifier_x$-Lbl_{acc} prevails. This shows that at the level of individual statements, other aspects of the faithfulness representation than context familiarity, such as distractors, etc., are non-negligible and also encoded in LLM hidden space, but not captured in context familiarity. Nevertheless, without any exposure to factual accuracy, $classifier_x$-Lbl_{corpus} still achieves non-trivial performance.

As the final verdict, we take $classifier_0$-Lbl_{corpus} for CL_0 and $classifier_x$-Lbl_{acc} for CL_x as the default setup for our HALLUCANA lookahead to be evaluated below, and additionally report results solely using classifiers trained on Lbl_{corpus}, as a robustness lower-bound.

5.2 The Canary Lookahead

We now evaluate HALLUCANA end-to-end, on reducing hallucination in long-form LLM generation.

When applying the lookahead, we convert classifier predictions to binary verdicts using representative thresholds on the dev set, maximizing F-scores along the F-R curves as elaborated in §4.

Table 2 presents the results. First, when we use pre-hoc scorers to prevent hallucination before decoding (CL_0), we approach the performance of the SOTA *SelfCk* baseline, while requiring less than 3% of its compute (7.9 vs. 298.5).

Second, when we use ad-hoc scorers to manipulate decoding towards faithful continuations (CL_x), we outperform beam decoding: we get 1.0 point higher FActScore, and generate 20% more atomic facts per response trading off 20% responses.

Crucially, our full lookahead ($CL_0 + CL_x$) yields the best overall generation quality, improving over vanilla greedy decoding by up to 2.5x; we also improve over prior SOTA by 24%, while consuming 6 times less compute (47.1 vs. 298.5).

Interestingly, using classifiers trained with Lbl_{corpus} context-familiarity labels without exposure to accuracy, $(CL_x)_{corpus}$ and $(CL_0 + CL_x)_{corpus}$ also effectively reduces hallucinations and improves generation quality, both outperforming their respective baselines. This shows the key role of context familiarity in the classification performance and the robustness of HALLUCANA.

To summarize, we may conclude:

1. the CL_0 scorer is an effective non-invasive

hallucination detection technique, which is always recommended in application;

2. where compute budget allows, the full lookahead is recommended, as it substantially improves factual faithfulness of LLM generations, while being informative and responsive;

3. the LLM internal representation of context familiarity is a major contributor to the factuality classification performance, such that classifiers trained on context familiarity are able to effectively detect hallucination in long-form generation;

5.3 Ablation Studies

We conduct ablation studies on the factscore dev set to evaluate our design choices. Table 3 presents the results.

Is it important to identify Critical Time Steps? *Yes*. As introduced in §3.1, we use an entropy-based heuristic to identify **critical time steps**, and apply CL_x scorer only at these time steps.

In the $- critical$ row, we report results without identifying critical time steps. The average FS is higher, but responses are substantially shorter (-4.3 atomic facts per response), and the response rate suffers severely (48.6% → 16.4%). This shows, that selectively applying CL_x at critical time steps is essential for reducing noise and false vetos, and does not spoil response factuality.[10] In addition, critical time steps also help reduce the computational footprint (TpF down 133.4 → 48.7).

Is it important to veto continuations with low CL_x scores? *Yes*. When the ad-hoc lookahead score $CL_x(j_{crit}^k) < \tau_x$, we veto the branch j_{crit}^k because it likely leads to hallucination.

In the $-veto$ row, we report results without veto. With the CL_x scores, the LLM generates more information before hitting <eos> (+4.9 atomic facts per response), but the average FActScore of these responses drops sharply (33.7 → 25.3). This is because, without the veto mechanism, CL_x cannot help when all tested options are hallucinatory.

Is it important to amend logit scores in addition to vetoing? *Yes*. When the CL_x score is above

---

[10]Note that the first few facts in a biography tend to be easier to infer and have higher FActScores (e.g. inferring nationality from names, etc.), therefore with the large difference in average length and responsiveness, the moderate difference in **Avg. FS** is not sufficiently indicative.

Ablation	$GQI_{0.2}\uparrow$	Avg. FS (%) $\uparrow$	Avg. #Facts $\uparrow$	% respond $\uparrow$	$TpF\downarrow$
$CL_0 + CL_x$	**1.52**	**33.7**	**14.8**	**48.6%**	**48.7**
$-\ critical$	0.70 (-0.82)	36.4 (+2.7)	10.5 (-4.3)	16.4% (-32.2%)	133.4 (+84.7)
$-\ veto$	0.63 (-0.89)	25.3 (-8.4)	19.7 (+4.9)	66.7% (+18.1%)	50.9 (+2.2)
$-\ score$	0.90 (-0.62)	32.9 (-0.8)	14.1 (-0.7)	49.7% (+1.1%)	48.8 (+0.1)
$-\ continuation$	1.44 (-0.08)	40.9 (+7.2)	16.9 (+2.1)	33.3% (-15.3%)	12.8 (-35.9)
beam-4	0.67 -0.85	26.7 (-7.0)	10.5 (-4.3)	97.8% (+49.2%)	55.5 +6.8

Table 3: Ablation study over HALLUCANA, for Falcon-7b-instruct generated biographies on **factscore dev set**.

threshold τ_x (i.e. not vetoed), we take the weighted geometric mean between CL_x and logit scores as the final score to select the next tokens.

In the $-score$ row, we remove this score amendment, and observe a significant drop in generation quality. Note that **Avg. FS** takes a macro-average over the responses, whereas GQI incorporates a micro-average. GQI index suffers more than **Avg. FS** and **Avg. #Facts**, this means, $CL_x-score$ leads to a skewed distribution of FActScores where long responses have much lower FS than short ones.

Summarizing the two ablations above, for CL_x, the veto mechanism is important for purging hallucinatory continuations, and the scoring mechanism is also important, for keeping the LLM on track of having faithful contents to present.

Can we further improve efficiency by avoiding decoding the continuations for CL_x? *Yes and no.* Also with the CL_x scorer, from each likely branch j_{crit}^k, we decode for up to $N = 8$ steps to complete the current statement before predicting factuality. If we avoid these extra decoding steps, and apply the classifier after the first lookahead token (effectively setting $N = 1$), CL_x would become more lightweight, with negligible computational overhead.

In the $-continuation$ row, we report lookahead performance without the extra decoding steps. We observe relatively promising results with generation quality only slightly lower than our main setup. Especially, the average FActScore and the average number of atomic facts are both higher; however, the con side is that its response rate is excessively low, where only a third of queries receive valid responses. Even without the CL_0 filtering, the ratio of valid responses is still only 41.5%.

Therefore, we do not use this config as our main setup, but note that users with low response rate expectations and/or tight computation constraints

(the TpF metric for $-continuation$ is comparable to that of regular greedy decoding) are still encouraged to apply tok_x without extra decoding.

6 Conclusion

In this paper, we introduce HALLUCANA, a canary lookahead to address factuality hallucination in LLM long-form generation. HALLUCANA consists of two phases: a pre-hoc scorer applied before generation begins, and an ad-hoc scorer applied during generation at critical decoding steps.

HALLUCANA detects hallucinations using lightweight classifiers trained using out-of-domain data to extract the internal factuality representation in the LLM hidden space. Through our experiments, we show that HALLUCANA substantially outperforms prior SOTA methods on generation quality, while at the same time consuming 6x less compute. Furthermore, our exploration reveals the intimate correlation between the LLM's internal representations for factuality and the familiarity of context. The performance of the faithfulness classifiers is deeply rooted in the parametric memory of context familiarity, embedded within the LLMs and acquired during the pre-training phase.

For future work, we plan to improve context disambiguation for context familiarity proxy, explore more sophisticated critical time step identifiers, and extend HALLUCANA to reasoning-intensive tasks.

Limitations

In this paper, our experiments are based on the benign users' assumption, where we do not consider prompt attacks.

Also, we develop HALLUCANA only based on greedy decoding. We leave experiments with best-first-search to future work.

Our evaluation metric focuses on factuality, informativeness and responsiveness, where we do

not quantitatively evaluate the fluency of the responses due to the lack of an automatic metric. We offer a case study in Appendix F where we qualitatively compare the generation fluency between our method and baselines.

Our approach, falling under the category of white-box approaches, does not apply to LLMs with closed-source model parameters. For the benefit of the analysis, we use LLMs whose pre-train corpora are also available, although we note that in the absence of the LLMs' own pre-train corpora, other open-source large pre-train corpora such as falcon-refiendweb or C4, are also good for inducing the context-familiarity proxy.

Our evaluation metric for computational footprint does not differentiate the amount of compute and latency. This is because we assume tight computing resources, where the overall objective is not only to reduce latency for individual queries, but also to reduce the overall workload of serving.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon Series of Open Language Models. ArXiv:2311.16867 [cs].

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. ReFinED: An Efficient Zero-shot-capable Approach to End-to-End Entity Linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Amos Azaria and Tom Mitchell. 2023. The Internal State of an LLM Knows When It's Lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson,

Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models. ArXiv:2210.11416 [cs].

Jinwen He, Yujia Gong, Zijin Lin, Cheng'an Wei, Yue Zhao, and Kai Chen. 2024. LLM factoscope: Uncovering LLMs' factual discernment through measuring inner states. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10218–10230, Bangkok, Thailand. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. ArXiv:2111.09543 [cs].

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2023. Inspecting and Editing Knowledge Representations in Language Models. ArXiv:2304.00740 [cs].

James Y. Huang, Sailik Sengupta, Daniele Bonadiman, Yi an Lai, Arshit Gupta, Nikolaos Pappas, Saab Mansour, Katrin Kirchhoff, and Dan Roth. 2024. Deal: Decoding-time alignment for large language models.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):248:1–248:38.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. ArXiv:2310.06825 [cs].

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared

Kaplan. 2022. Language Models (Mostly) Know What They Know. ArXiv:2207.05221 [cs].

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large Language Models Struggle to Learn Long-Tail Knowledge. In *Proceedings of the 40th International Conference on Machine Learning*, pages 15696–15707. PMLR. ISSN: 2640-3498.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466. Place: Cambridge, MA Publisher: MIT Press.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. In *Advances in Neural Information Processing Systems*.

B. A. Levinstein and Daniel A. Herrmann. 2023. Still No Lie Detector for Language Models: Probing Empirical and Conceptual Roadblocks. ArXiv:2307.00175 [cs].

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with Confidence: Uncertainty Quantification for Black-box Large Language Models. ArXiv:2305.19187 [cs, stat].

Zhenzhen Liu, Chao Wan, Varsha Kishore, Jin Peng Zhou, Minmin Chen, and Kilian Q. Weinberger. 2023. Correction with Backtracking Reduces Hallucination in Summarization. ArXiv:2310.16176 [cs].

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic A*esque Decoding: Constrained Text Generation with Lookahead Heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of Hallucination by Large Language Models on Inference Tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774, Singapore. Association for Computational Linguistics.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, pages 1–8, New York, NY, USA. Association for Computing Machinery.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. In *The Twelfth International Conference on Learning Representations*.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2024. Llms know more than they show: On the intrinsic representation of llm hallucinations.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The Refined-Web Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. ArXiv:2306.01116 [cs].

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback. ArXiv:2302.12813 [cs].

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Kai Sun, Yifan Ethan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. 2023. Head-to-Tail: How Knowledgeable are Large Language Models (LLM)? A.K.A. Will LLMs Replace Knowledge Graphs? ArXiv:2308.10168 [cs].

Yi Tay. 2023. A new open source flan 20b with ul2.

S. M. Towhidul Islam Tonmoy, S. M. Mehedi Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. ArXiv:2401.01313.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. ArXiv:2307.09288 [cs].

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A Stitch in Time Saves Nine: Detecting and Mitigating Hallucinations of LLMs by Validating Low-Confidence Generation. ArXiv:2307.03987 [cs].

David Wan, Mengwen Liu, Kathleen McKeown, Markus Dreyer, and Mohit Bansal. 2023. Faithfulness-Aware Decoding Strategies for Abstractive Summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2864–2880, Dubrovnik, Croatia. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. ArXiv:2203.11171 [cs].

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do Large Language Models Know What They Don't Know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

A Prompts in LLM generations

A.1 Q-A Rephrasing

For stylistic consistency, we expand the answers in the QA data entries into sentences. To do so, we use the Mistral-7B-Instruct-v0.1 model (Jiang et al., 2023), and prompt it to take in a Q-A pair and convert it into a declarative sentence. We use one in-context example in the prompt, and wrap the prompt in a chat template provided with Mistral:

> **User:** *Paraphrase the following Q-A pair into a proposition. Q: Where is the capital of France? A: XXXX.*

Assistant: *The capital of France is XXXX.*

User: *Paraphrase the following Q-A pair into a proposition: Q:* [QUESTION] *A: XXXX.*

Assistant:

When training the faithfulness classifiers (specifically, the $classifier_x$ classifiers), hidden states at the last expanded answer tokens are used as training input. The placeholder "XXXX" is installed here, so that it can be easily replaced with different answer tokens: when training using Lbl_{acc} labels, LLM predicted answers are plugged in to model the factuality of this predicted answer; when using Lbl_{corpus} labels, gold answers from the training dataset are used to purge exposure to accuracy, and model only context familiarity.

A.2 Long-form Generation

When generating biographies for the factscore dataset, we use the following conversational prompt:

> **User:** *Can you write me a bio to introduce* [PERSON]*?*
>
> **Assistant:** *Sure, here's their bio:*

Note that for the CL_0 scorer in the lookahead, we apply it at the last prompt token, i.e. the ":" after "bio" in the prompt response.

B Labels for Faithfulness Classifiers

B.1 Lbl_{acc} Details

When training $classifier_x$ faithfulness classifiers with Lbl_{acc} labels, we extract hidden states using LLMs' own predicted answers. These answers are predicted using greedy decoding, with a set of four few-shot examples, wrapped in a simple Q-A template. Specifically:

> *Q:* [EXAMPLE 1] *A:* [EXAMPLE 1]
>
> *Q:* [EXAMPLE 2] *A:* [EXAMPLE 2]
>
> *Q:* [EXAMPLE 3] *A:* [EXAMPLE 3]
>
> *Q:* [EXAMPLE 4] *A:* [EXAMPLE 4]
>
> *Q:* [EXAMPLE 5] *A:* [EXAMPLE 5]
>
> *Q:* [QUESTION] *A:*

Accuracy labels are then acquired by string match between the predicted answer and the gold answer. A string match is met when the predicted answer exactly matches the gold answer, or when the predicted answer is a prefix or a postfix of the gold answer, and vice versa.

Notably, we also tried using NLI labels produced by DeBERTa-V3 (He et al., 2023) to replace string match as the measure of factual accuracy. However, using NLI labels yielded consistently inferior results, therefore we leave them out of this paper.

B.2 Lbl_{corpus} Details

When training $classifier_0$ and $classifier_x$ classifiers with context corpus-familiarity labels (Lbl_{corpus}), we use the frequency of contexts in pre-training corpora as the proxy.

We extract mentions of named entities and mentions of time, year, ordinals, etc., with the ReFinED entity linker (Ayoola et al., 2022), and acquire entity spans from them.

Kandpal et al. (2023) used entity combinations that co-occur in the same passage, as the criterion for context mentions in pre-train corpora. We observe that this criterion is noisy when entity pairs are considered co-occurring whenever they both occur in the same document: they could be far apart and have very weak semantic connections. Therefore, we impose the restriction where only adjacently-appearing entities are considered co-occurring.

Differently from Kandpal et al. (2023); Sun et al. (2023), who analyse the relation between entities frequencies/popularities and the LLMs' parametric knowledge, our goal here is to extract an internal context-familiarity representation from the LLMs' own hidden space. Therefore, we do not need to assign a corpus-frequency label to every entry, as long as the labelled entries to be included in the training set are sufficiently representative.

As such, in addition to the adjacency restriction mentioned above, we also define "context" in a more restrictive, finer-grained way. We count two mentions in the pre-train corpora as mentions of the same context, only when the same set of 3 entities appear consecutively in both passages. This is also more stringent than the entity pairs, which leads to less coverage, but also reduces noise and yields more accurate contexts. For instance, the entity pair *<France, Argentina>* is very broad and ambiguous, whereas the 3-tuple *<France, Argentina,*

Config	AUROC
Trivia QA (Joshi et al., 2017)	
sentence 2-gram	60.6
consecutive 2-gram	61.8
consecutive 3-gram	62.4
Natural Questions (Joshi et al., 2017)	
sentence 2-gram	64.2
consecutive 2-gram	64.4
consecutive 3-gram	64.7

Table 4: Correlation between different Lbl_{corpus} configs and Lbl_{acc}, measured using area under ROC.

world cup> is much more precise, where all the mentions with these entities appearing consecutively together are truly relevant to each other.

Without blindly trusting the string-match accuracy of LLM predictions as the gold metric of success, we do argue that an excessively low correlation between the context familiarity labels and string-match accuracy labels indicate excessive noise. Thus, in Table 4, we validate our choice of Lbl_{corpus}, by comparing among different instantiations of Lbl_{corpus} with respect to this correlation. From the results, we observe that **consecutive 3-gram** is the least noisy.

In light of its advantage in noise reduction, we use frequencies of **consecutive 3-gram** entity mentions as the context familiarity proxy.

C Hyper-parameters in Classifier Training

For each MLP classifier, we set the input dimensionality of each layer to be log-linearly decreasing from the LLM hidden size to 64 (at the last MLP layer).

For classifier inputs, we use two popular open-domain QA datasets, TriviaQA (Joshi et al., 2017) and Natural Questions (NQ) (Kwiatkowski et al., 2019), where we concatenate the two datasets at each epoch. Since the test sets for both QA datasets have hidden labels, we sub-split the train sets into train2 and dev2 subsets to conduct training, and repurpose the original dev set as a test-set substitute to monitor performance in the training domains.

For NQ dataset, we remove those questions where the gold answers are null. This is because, in the annotation process of the NQ dataset, null gold answers mean the annotators could not find the gold

answers from the reference document, where it is unclear whether those answers exist w.r.t. world knowledge, or what those answers would be.

For each QA entry, we take the last layer hidden states at the last question/answer tokens that are *not special tokens (such as </s>)..* We have also experimented using earlier layers and excluding the punctuation tokens, but no improvements have been observed.

We train the faithfulness classifiers with AdamW optimizer (Loshchilov and Hutter, 2019), using learning rate 1e-3. We train the classifiers for 300K entries, saving every 25K entries. Using a batch size of 128, this translates to a total of 2344 steps and 196 steps between checkpoint saves.

Notably, Lbl_{acc} labels are binary, thus we use cross-entropy loss for training; Lbl_{corpus} labels are scalar values between 0 and 1, therefore we use mean-squared error loss for training. In either case, we track dev2 set losses as the checkpoint selection criterion; when dev2 loss stops decreasing for 2 consecutive saves, we impose early stopping.

D Flan-UL2 Results

We present hallucination mitigation results with Flan-UL2 in Table 5. We can observe that, as an encoder-decoder model, Flan-UL2 generates responses that are shorter than that from Falcon-7b-instruct. This is related to its fine-tuning on FLAN, a dataset not focused on text generation but classification/reasoning tasks with short responses.

Nevertheless, the results are consistent with Table 2, with HALLUCANA outperforming baselines.

Note that corresponding to the generally shorter responses produced by this encoder-decoder LLM, we also relax the length threshold for abstention from 16 words to 8 words.[11]

E Metrics Details

E.1 FActScore Implementation Details

We follow Min et al. (2023) to calculate FActScore for generated biographies, w.r.t. Wikipedia as the gold knowledge source. Compared to their original implementation, we made two adjustments. First, we use a more advanced open-source LLM, Mistal (Jiang et al., 2023), as the backbone LLM, instead of the Inst-LLaMA (Touvron et al., 2023; Wang et al., 2022) used in the original implementation.

---

[11]By words, we mean space-separated natural language words, a larger unit than LLM tokens.

Method	$GQI_{0.2}$ ↑	Avg. FS (%) ↑	Avg. #Facts ↑	% respond ↑	TpF ↓
greedy	0.22	21.6	7.0	99.8%	10.2
beam-6	0.07	24.3	6.7	98.6%	55.6
MAX(−logp)	0.29	22.4	6.9	93.8%	11.0
SelfCk (2023)	0.51	27.2	7.5	63.5%	747.9
HALLUCANA					
CL_0	0.61	29.0	6.8	57.3%	9.9
CL_x	0.25	24.4	7.6	82.6%	58.6
$CL_0 + CL_x$	**0.59**	**32.1**	**7.2**	**46.1%**	**54.0**

Table 5: Hallucination mitigation performance for Flan-UL2 generated biographies for **factscore test** dataset. For strict computation constraints, CL_0 is recommended; generally, $(CL_0 + CL_x)$ is recommended.

We make this change because the Mistral LLM is independent of the backbone LLMs used in our experiments, exhibits SOTA performance, and is an open-source model with a permissive apache-2.0 licence. Second, instead of penalizing short generations using the γ term as in their GitHub release[12], we take into account the generation informativeness in a more fine-grained way, where the GQI metric is proportional to the generation lengths.

In order to verify the validity of the Mistral-based FActScore metric, we compare it with the two variants of FActScores released by (Min et al., 2023) using ChatGPT and Inst-LLaMA. Specifically, we use the LLM generations from the set of LLMs used in their evaluation (Table 3 of their paper), and present the comparison in Table 6. As shown, the ranking among the set of LLMs is consistent between the "retrieval+Mistral+npm" setup that we use, and their two original setups, proving its validity.

E.2 Area under FActScore-Rejection Curve

For comparing the discriminative power of different classification approaches on the vanilla greedy LLM generations on the factscore dataset, we use each classifier to rank the generated passages by faithfulness, and compare each ranking against the random ranking and the perfect ranking.

Using the predictions from each classifier, we iteratively reject the passages with the lowest predicted-faithfulness, and calculate an average FActScore supposing that the classifier is thresholded to reject only this and more lowly-predicted passages. We put all the thresholds on a FActScore-RejectionRate plain, where the thresholds are connected as the FActScore-Rejection Curve.

In Figure 2, we show an example set of FActScore rejection curves, reported on falcon-7b-instruct greedy generations. The bold **orange** curve represents the random ranking, where precision expectedly does not change no matter how many samples are rejected; the bold **blue** curve represents the perfect ranking, where the lowest FActScore passage is always rejected at each turn.

Actual classifiers have FActScore-Rejection curves between the random and the perfect rankings, so we measure the performance of a classifier by how much above-random discriminative power it has, relative to perfection. This is $AUC\text{-}FR_{norm}$, the ratio of above-random area under the FR-curve of the classifier w.r.t. the perfect upper bound.[13]

When selecting a threshold for each classifier to use in the lookahead (§5.2), we choose the representative thresholds maximizing F_β-scores on dev sets. For $classifier_0$ classifiers, we set $\beta = 1.0$, the default value; for $classifier_x$ classifiers, since classification is repeatedly applied throughout a generation run, and vetoing all options at any critical time step leads to generation being halted, we use $\beta = 1.2$ to avoid over-rejection.

E.3 Generation Quality Index (GQI)

We provide GQI as an intuitive holistic metric of LLM generation quality, using one single index. This metric is provided for the reader's convenience, and reflects how we value a generated response, taking into account factuality, informativeness and responsiveness:

---

[12] https://github.com/shmsw25/FActScore

[13] On a different classification task of natural language inference, McKenna et al. (2023) has used an AUC_{norm} metric on the Precision-Recall curve, which is in a similar spirit to the metric we use.

LLM	retrieval+Mistral+npm	retrieval+ChatGPT	retrieval+llama+npm
InstructGPT	39.6	47.6	41.1
ChatGPT	56.1	65.1	58.7
PPLAI†	58.9	72.3	61.6

Table 6: FActScores (%) for the set of 3 LLMs, measured using different configurations; "retrieval+Mistral+npm" is the configuration used in our main experiments. Using different configurations, the rankings among the set of LLMs are consistent. PPLAI† is retieval-augmented.
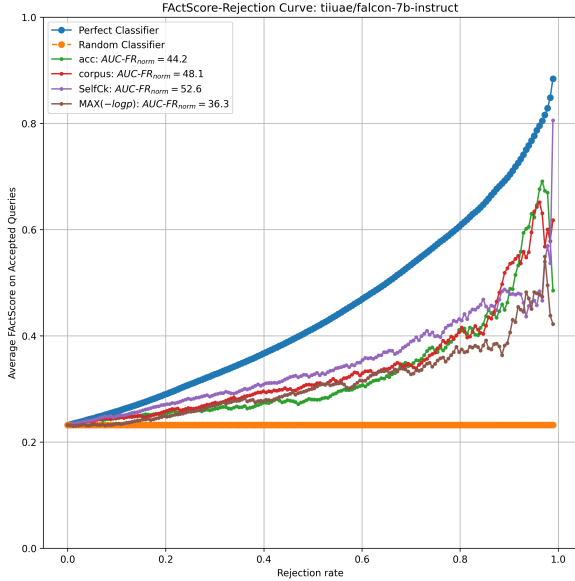

Figure 2: The FActScore-Rejection Curves of various classification approaches, when applied on greedy-decoded generations of Falcon-7b-instruct, over factscore dev set.

$$gqi_\gamma(d) = (FS_d - \gamma) \times \#Facts_d$$

$$GQI_\gamma = \frac{\sum_{d \in D} gqi_\gamma(d)}{|D| \times \sqrt{response\ rate}}$$

The design of GQI follows the tenets below:

First, responses with $FS_d < \gamma$ are negative assets, as they do not provide useful information and only add to the confusion. Relative to the baseline performance of the LLMs we test, we empirically set $\gamma = 0.2$.

Second, for the same total number of atomic facts, more informative responses for fewer entries are preferred, over more but less informative responses; this is because, while nationalities etc., which come at the beginning of a biography, are easier to infer from the names etc., later generations are more knowledge-intensive. In the denominator term, taking $(response\ rate)^0 \equiv 1$ means

responsiveness is taken equally as informativeness; on the other hand, taking $(response\ rate)^1 \equiv response\ rate$ means responsiveness is completely ignored. We take a balance between the two extremes and take the square root $\sqrt{\cdot}$.

We acknowledge that the GQI metric reflects our beliefs in the value of LLM long-form generations, and may not universally apply. We encourage readers to also inspect the individual measures of each aspect to gain a complete view and make judgements w.r.t. their own applications.

F Qualitative Analysis on Generation Fluency

In Table 7, we compare the responses generated using vanilla greedy decoding, against those generated using HALLUCANA. We present the first 3 entries in factscore test set where both responses are not abstained. Since the performance of HALLUCANA on hallucination reduction has been quantitatively verified in the above, in this section we are primarily interested in understanding how well it has preserved the fluency of the LLM generations.

We can observe that responses generated using HalluCana remain fluent and grammatical, as much so as the greedy decoding. The only caveats come at the 2 purple coloured text spans: these are when the veto mechanism rejects all the likely branches, and we force the next token to be period (.) to avoid hallucination.

Since these spans do not significantly impact our quantitative evaluations, we do not specifically take a stance on how to post-process the output to remove these spans. We do offer a few possible options, including: 1) deleting words in these spans with forced periods, until the sentence is syntactically complete, e.g. with a lightweight parser; 2) marking "[UNK]" for the slots; 3) querying knowledge sources or larger LLMs at these points to fill in these blanks.

Additionally, we note that a back-tracking approach in LLM decoding has been proposed (Liu et al., 2023). It is also possible to avoid the force-stopped sentences by integrating the back-tracking approach in our algorithm. We leave these explorations to future work.

G   Computational Infrastructure and Footprint

For the experiments in this paper, we use a g5-12xlarge cloud server from AWS[14]. Training faithfulness classifiers consumes 15 minutes using only 1 GPU. Generating long-form responses consumes 30 minutes on factscore dev-set and 70 minutes on factscore test set; 2 A10G GPUs are required for generating responses using Falcon-7B, on the other hand, all 4 A10G GPUs on the server are required for generating responses using Flan-UL2. Evaluating generated responses with Mistral-based FActScore metric consumes 2 hours for factscore dev-set, and 5 hours for factscore test-set, requiring 2 A10G GPUs.

The most computationally demanding part of the pipeline is obtaining linked entities for LLM pre-train corpora using ReFinED entity linker, which consumes 3000 GPU hours for the C4 corpus, and 6000 GPU hours for Falcon-ReFinedWeb. We refer readers to Kandpal et al. (2023) for a rough substitute, where C4 corpus is parsed using an older entity linker (Mendes et al., 2011).

---

[14]https://aws.amazon.com/ec2/instance-types/g5/

Index	Greedy	HALLUCANA
3	Iggy Azazela is a multi-platinum, Australian rapper, singer, songwriter, and actress. She rose to fame in 2010 with the release of her debut single, 'Iggy Azalea,' which reached number one on the Australian Singles Chart. Since then, she has released several successful albums, including 'Reign,' 'The New Classic,' and 'In My Defense.' Azalea has also been featured on songs by other artists, including Eminem and Ariana Grande. She has been nominated for multiple Grammy Awards and has won multiple awards for her music. In addition to her music career, Azalea has also appeared in movies and TV shows, including 'Furious 7' and 'The Fast and Furious' franchise.	Iggy Azazela is a rapper, singer, and songwriter from Australia. She gained popularity in 2010 with the release of her mixtapes Iggy Azalea and Reclassified. Azalea rose to fame with the release of her debut studio album, The New Classic, in 2014. She has since been nominated for numerous awards, including multiple ARIA Awards and a BET Award.
6	Radhika is an Indian actress who has appeared in numerous Hindi, Telugu, and Tamil films. She has also acted in a few English-language films. She is known for her roles in the web series 'The Raikar Case' and the film 'The Perfect Girl 2.' Radhika has won several awards for her performances, including the Filmfare Award for Best Actress in a Supporting Role.	Radhika is an Indian actress who has predominantly worked in Telugu and Tamil films. She has also appeared in a few Hindi films. She has been a part of critically acclaimed films like Raja Reddy, RRR, and Jersey. She has won several awards for her performances, including the Filmfare Award for Best Actress in a Supporting Role for. She is known for her versatile roles and is considered one of the most sought-after actresses in the South Indian film industry.
7	David Galloway is a writer and journalist based in London. He has written for a variety of publications, including The Guardian, The Independent, and The New York Times. He is the author of the book 'The New York Times Best of 2008' and has written for the BBC, The Economist, and The Financial Times. He is currently working on a book about the history of the New York Stock Exchange.	David Galloway is a freelance writer and editor based in the UK. He specializes in writing about travel, food, and culture. He has been writing professionally for over 10 years and has.

Table 7: Comparing biographies generated by Falcon-7b-instruct using greedy search and HALLUCANA. Examples are the first 3 entries in factscore test set where both responses are not abstained, indices are from the original test set.