Is There No Such Thing as a Bad Question? H4R: HalluciBot for Ratiocination, Rewriting, Ranking, and Routing

William Watson*, **Nicole Cho***, **Nishan Srishankar**
J.P. Morgan AI Research
nicole.cho@jpmorgan.com

Abstract

Hallucination continues to be one of the most critical challenges in the institutional adoption journey of Large Language Models (LLMs). While prior studies have primarily focused on the post-generation analysis and refinement of outputs, this paper centers on the effectiveness of queries in eliciting accurate responses from LLMs. We present **HalluciBot**, a model that estimates the query's propensity to hallucinate ***before generation***, without invoking any LLMs during inference. HalluciBot can serve as a proxy reward model for query rewriting, offering a general framework to estimate query quality based on accuracy and consensus. In essence, HalluciBot investigates how poorly constructed queries can lead to erroneous outputs - moreover, by employing query rewriting guided by HalluciBot's empirical estimates, we demonstrate that 95.7% output accuracy can be achieved for multiple choice questions. The training procedure for HalluciBot consists of perturbing 369,837 queries n times, employing $n+1$ independent LLM agents, sampling an output from each query, conducting a Multi-Agent Monte Carlo simulation on the sampled outputs, and training an encoder classifier. The idea of perturbation is the outcome of our ablation studies that measures the increase in output diversity (+12.5 agreement spread) by perturbing a query in lexically different but semantically similar ways. Therefore, HalluciBot paves the way to ratiocinate (76.0% test F1 score, 46.6% in saved computation on hallucinatory queries), rewrite (+30.2% positive class transition from hallucinatory to non-hallucinatory), rank (+50.6% positive class transition from hallucinatory to non-hallucinatory), and route queries to effective pipelines.

Extended version — https://arxiv.org/abs/2404.12535

1 Introduction

Despite the promising potential for a myriad of use cases, Large Language Models (LLMs) offer limited insights into their chain of thought (Liang et al. 2022; Wei et al. 2023; Kojima et al. 2023) and have the propensity to hallucinate in various circumstances (Jiang et al. 2021). Common factors that drive hallucinations encompass high model complexity, flawed data sources, or inherent sampling randomness.

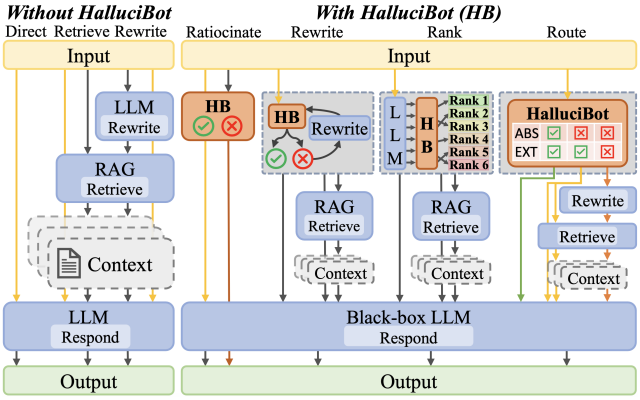

Figure 1: Comparison of traditional inference methods and HalluciBot's use-cases. In the former, the user inputs a query either through a direct inference or a retrieval-augmented generation (RAG) pipeline. If the output is hallucinatory, the user must decide whether to end the session or revise the query for successive generation rounds. In contrast, HalluciBot can be used to assess the query's quality ***before generation***. Therefore, users can gain insight into the hallucination risk ("Ratiocinate"), automate the query rewriting stage through informed feedback ("Rewrite") or Best-of-N sampling across multiple candidates ("Rank"), and route the query across different operating modes ("Route"), since HalluciBot is scenario-aware (**Extractive** / **Abstractive**), potentially bypassing computationally expensive stages, such as RAG or Rewrite.

Specifically, the intrinsic trade-off between greedy deterministic decoding and the creativity spawned through nucleus sampling induces a heightened propensity to hallucinate (Huang et al. 2023) - LLMs frequently advance output quality through different sampling methods (Holtzman et al. 2020, 2018; Radford et al. 2018). The challenge of understanding hallucinations is compounded by limitations such as the frequent inaccessibility into the LLMs' training datasets (Liang et al. 2022). HuggingFace's release of its "Hallucinations Leaderboard" on January 29th, 2024 (Gao et al. 2023) highlights the importance of resolving hallucination-related issues via the concerted effort of evaluating different LLMs. In this context, the majority of current studies have focused on the post-generation phase of output analysis as expanded

*These authors contributed equally.

in Peng et al. (2023) such as - (1) self-refinement via feedback loops on the model's output (Madaan et al. 2023; Watson et al. 2025), (2) analysis of logit output values to detect hallucination (Varshney et al. 2023), or (3) for a minority of studies focused on the pre-generation phase, the ingestion of recent knowledge to improve performance (Tonmoy et al. 2024). We propose a novel model, **HalluciBot**, that predicts the probability of hallucination, ***before any generation***, for a given query. In essence, this paper refocuses the study of hallucination to an empirical evaluation of the input query - how much does the query's quality influence the model's propensity to hallucinate? Therefore, HalluciBot estimates,

- a binary classification of the query's propensity to hallucinate ("Yes" or "No"), as well as,
- a non reinforcement-learning method to guide query rewriting, enabling the construction of this encoder to be agnostic to closed-source or open-source LLMs (Ma et al. 2023).

We train HalluciBot as a binary classifier to predict whether a query will lead to erroneous outputs. To generate ground truth labels, we use a Multi-Agent Monte Carlo simulation that perturbs the query and checks for inaccuracies. If any perturbed version causes an error, the original query is labeled as hallucinatory. In this paper, HalluciBot leverages `gpt-3.5-turbo`, trained via (1) perturbing 369,837 queries n times to retain the original semantic meaning yet diverge lexically, (2) employing $n+1$ independent agents to sample an output from each perturbation including the original query, at a temperature of 1.0 for diversity, (3) conducting a Monte Carlo simulation on 2,219,022 sampled outputs, and (4) deriving an empirical estimate into the expected rate of hallucination $p_h(q_0)$ for the original query. We prove that introducing perturbations before sampling $n+1$ outputs for query q_0 garners a 13.2 point spread in the lower and upper bound accuracy, with a 12.5 point decrease in Fleiss's κ for agreement, even as the modal accuracy remains largely unchanged (1.3 point difference). In other words, perturbations introduce more variability in the outputs, while preserving the central tendency. As HalluciBot generates the probability of hallucination in the pre-generation phase, the estimates can be used in a myriad of downstream modalities (Figure 1) such as: **"Ratiocinate"** to purely estimate the query's quality; **"Rewrite"** to leverage the probabilities and improve the query's quality via iterative feedback; **"Rank"** to rank perturbations, using probabilities as a proxy reward model in *Best-of-N* sampling; **"Route"** to route the best next steps, depending on scenarios such as **Extractive** or **Abstractive**. By cross-tabulating the predicted hallucination rates across scenarios, HalluciBot can act as a router, through which certain queries can be guided to a black-box LLM, while others will require a more complex pipeline including context retrieval, web search, or agents (Watson et al. 2023; Zeng et al. 2024; Cho et al. 2024).

Contributions. As a result, our study has culminated in the following pillars of contribution. **(1)** HalluciBot is the first encoder-based model to derive, ***before generation***, an anticipated rate of hallucination *for any type of query*, achieving a validation accuracy of 73.6% (80.2% F1) and a testing accuracy of 69.5% (76.0% F1). **(2)** our approach to construct HalluciBot absorbs the computational complexity of Monte Carlo sampling, exploration, and training prior to the user session. Thus, institutions that employ HalluciBot can systematically save on the considerable amount of computational waste engendered by "highly probable" hallucinatory queries (46.6% in saved computation during testing). **(3)** the hallucination probability can be leveraged as a proxy reward model in a myriad of different infrastructures; HalluciBot paves the way to rewrite (+31.9% positive class transition from hallucinatory to non-hallucinatory), rank (+51.4% positive class transition from hallucinatory to non-hallucinatory), and route (+60.0% diverted) queries. **(4)** HalluciBot generalizes to systems with RAG or few-shot question answering systems with an LLM generator by differentiating the scenario in its prompt. Also, it can generalize to closed systems only accessible via API calls (OpenAI 2022; Google 2023). **(5)** HalluciBot's training methodology can be leveraged for any model or training corpus; it can be leveraged as a general means by which the research community can develop an encoder to assess query quality.

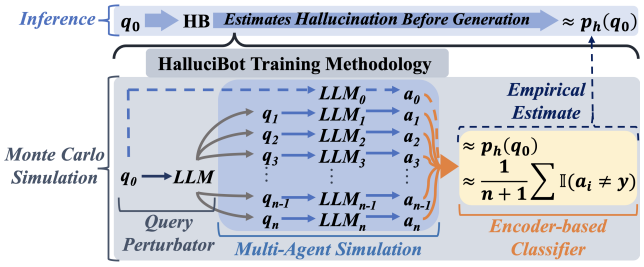

Figure 2: Training Overview. A single query, q_0, is **perturbed** in n different ways. Next, The original and perturbed queries q_i are independently answered by the Generator agents. This Multi-Agent Monte Carlo simulation provides an estimate into the **rate of hallucination** $p_h(q_0)$ for the original query q_0. Via these simulated results, HalluciBot is trained to predict the probability that any q_0 *could* hallucinate, and predict the expected consensus of sampled outputs ***before generation***.

2 Related Work

With regards to hallucination mitigation studies, an overwhelming majority focuses on the post-generation stage of analyzing outputs. A minority concentrates on the pre-generation phase and even amongst those, the focus lies in incorporating recent knowledge into LLMs. In detail, many expand on the universally utilized method of context-based retrieval systems (Lewis et al. 2020). Other methods include relying on the model's general knowledge (Khashabi et al. 2020). Certain work has focused on mitigating hallucinations by augmenting the way LLMs generate their answers. One of the more popular techniques is to have the model enumerate its chain-of-thought (Wei et al. 2023) while building context. Another method to augment generation with context is by semantic retrieval (Lewis et al. 2020; Liu et al. 2021), handling hallucinations as they arise (Varshney et al. 2023). PromptChainer (Wu et al. 2022) profiled techniques to craft LLM chains, in which the output of one LLM's generation process, when fed into the next LLM, can allow for more com-

Scenario	Datasets
Extractive	*SQuADv2*
Multiple Choice	*TruthfulQA*, *SciQ*, MMLU, PIQA, BoolQ, OpenBookQA, MathQA, ARC - E/C
Abstractive	*SQuADv2*, *TruthfulQA*, *SciQ*, WikiQA, HotpotQA, TriviaQA

Table 1: Dataset scenario split with *Reused Assets*.

plex tasks. Language Model Cascades (Dohan et al. 2022) demonstrated that LLMs can yield probabilistic programs to tackle multi-step reasoning problems. Self-consistency (Wang et al. 2023) leveraged a new decoding strategy to sample multiple generative pathways - then select the most consistent answer. Most recent work has focused on sampling-based calibration within a single model (Cole et al. 2023) or self-verification (Kadavath et al. 2022) - the latter focuses on generating a set of outputs and feeding those back into the LLM. Furthermore, Snyder, Moisescu, and Zafar (2023) explores how artifacts can differentiate hallucinated outputs. One common feature amongst these approaches is that the focus is on the output rather than the query. Alzahrani et al. (2024) explored how LLMs are highly sensitive to minute perturbations, such as changing the order of answer choices. Also, while Zheng and Saparov (2023) study lexical perturbations, no study on hallucinations employs a Multi-Agent approach coupled with query perturbations - which are hallmark features of HalluciBot.

3 Methodology Overview

What is Hallucination? In general terms, hallucination refers to a false perception of patterns or objects resulting from one's senses. With regards to LLMs, a myriad of studies bifurcate into (1) *factuality hallucinations* that refer to outputs which directly contradict or fabricate the ground truth while (2) *faithfulness hallucinations* define outputs that misunderstand the context or intent of the query (Huang et al. 2023; Ji et al. 2023). In this study, we introduce *truthful hallucination* as the motivation on why we are perturbing the original query. *Truthful hallucination* is defined as an LLM's inability to answer semantically similar but lexically different perturbations of a query. The motivation for *truthful hallucination* stems from the analysis that neural networks display an intrinsic propensity to memorize training data (Carlini et al. 2021) - in this case, memorizing the query and output. Given the risk of over-training LLMs, their opaque training data, and propensity to memorize - generating multiple outputs from the same query or analyzing a single output from a single query do not help measure *truthful hallucination*.

What is the Motivation for HalluciBot? HalluciBot focuses on distilling *LLM behavior* into a speedy encoder that can predict hallucination ***before generation***. Foremost, this is in contrast to prior work that uses multiple generations during a user's session to provide self-consistency (Manakul, Liusie, and Gales 2023). Next, our proposal differs from entropy based, log-prob based, or model based estimation techniques (Huang et al. 2023) that rely on the LLM's uncertainty to predict hallucinations - these methods focus on the model's bias while we focus on empirical estimates. Moreover, our approach consists of a Multi-Agent simulation which stands in stark contrast to the majority of current experiments that have focused on leveraging a single LLM agent to generate outputs from a single query (Cole et al. 2023; Kadavath et al. 2022; Snyder, Moisescu, and Zafar 2023). The training procedure for HalluciBot consists of perturbing each query $n = 5$ times, employing $n + 1 = 6$ independent LLM agents, sampling an output from each query, conducting a Monte Carlo simulation on 2,219,022 sampled outputs, and training an encoder classifier.

3.1 Multi-Agent Monte Carlo Simulation

What is the Purpose of a Monte Carlo Simulation? As evidenced by multiple studies and Table 3, hallucination is the outcome of multiple confounding variables - thus, it is highly unlikely that a tractable closed-form solution will be able to model hallucinations. Thus, we employ a **Monte Carlo** simulation as a means to derive empirical estimations of hallucination rates in LLMs, since this method is frequently leveraged to map probability in the presence of random variable inference (Swaminathan 2021). Thus, we estimate the probability density that a query induces hallucination.

What is a Query Perturbator? Via perturbations, we induce diversity to disentangle the generation process from any potential training bias (Alzahrani et al. 2024; Carlini et al. 2021). The **Query Perturbator** is a `gpt-3.5-turbo` LLM agent that generates $n = 5$ perturbations to the original query q_0 while retaining the same semantic meaning. In effect, the generation process can be summarized as returning a set of $\mathcal{Q} = \{q_0, q_1, \ldots, q_n\}$ query perturbations of size $n + 1$. The **Query Perturbator's** singular purpose is to: `Rewrite the query in {n} radically different ways`. One prompt call is sufficient to discourage duplicates. Temperature is set to 1.0 to prioritize creativity and lexical diversity. Our analysis in Table 3 shows that introducing perturbations, rather than sampling $n + 1$ outputs for query q_0, results in a 13 point spread between the lower and upper bound accuracy, a 12.5 point decrease in Fleiss's κ for agreement, while the modal accuracy remains largely unchanged. This suggests that perturbations inject variability into our Monte Carlo simulation, which is critical for observing diverse outputs and hallucinations. This corroborates the observation by Alzahrani et al. (2024) that LLMs are highly sensitive to even minor details.

What is an Output Generator? For the perturbed set $\mathcal{Q}$ for a sample q_0, the **Output Generator** consists of $|\mathcal{Q}| = n + 1$ **six independent** `gpt-3.5-turbo` LLM agents to generate outputs $a_i \in \mathcal{A}$ for each variation $q_i \in \mathcal{Q}$. The LLM agent will receive (1) for **Extractive** queries, a prompt with the query q_i, alongside context c_i, (2) for **Multiple-Choice** queries, candidate choices $k_i \in \mathcal{K}$, and (3) for **Abstractive** queries, no additional context. Temperature for all experiments is set to 1.0 to stress-test and encourage diversity.

How Do We Measure Accuracy? *Accuracy* serves as the measure of correctness, comparing the generated output a_i to the ground truth y, using partial, case-insensitive matching with the `TheFuzz` library. For **Multiple Choice** queries, the choice label is also considered. If there is no match between

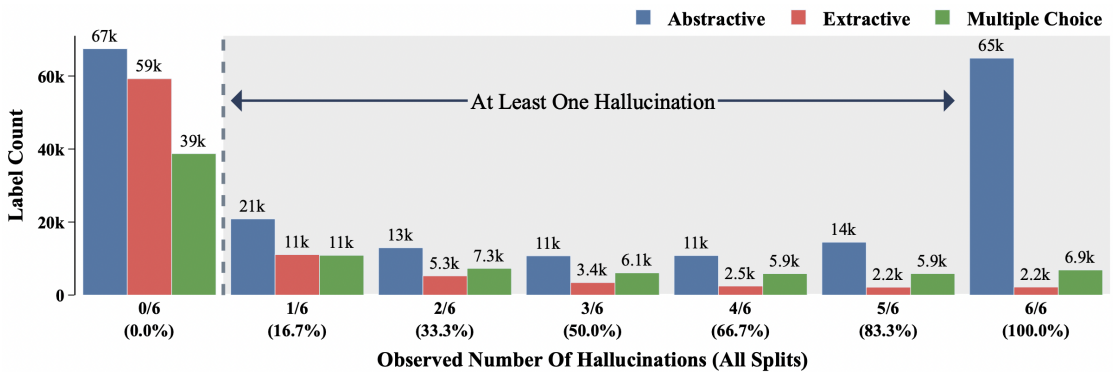

Figure 3: Distribution of the observed number of hallucinations per scenario. For **Extractive**, additional context mitigates hallucinations. For **Multiple Choice**, distractors can cause confusion amongst agents uniformly. For **Abstractive**, the absence of additional information results in extreme disparities in correctness, with most simulations showing no or all hallucinations.

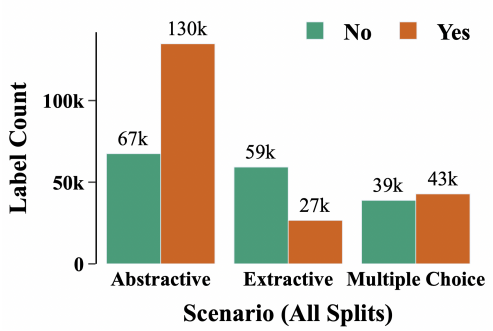

Figure 4: Binary distribution of labels indicating whether at least one hallucination occurred during simulation.

Binary	**Train**	**Val**	**Test**
No ($y = 0$)	139,142	17,153	9,306
Yes ($y = 1$)	163,350	27,338	13,548
Observed Rate	**Train**	**Val**	**Test**
0.0% ($y = 0/6$)	139,123	17,146	9,202
16.7% ($y = 1/6$)	35,114	4,974	2,757
33.3% ($y = 2/6$)	20,213	3,371	1,967
50.0% ($y = 3/6$)	15,749	2,757	1,768
66.7% ($y = 4/6$)	14,477	2,735	1,970
83.3% ($y = 5/6$)	17,123	3,242	2,171
100.0% ($y = 6/6$)	60,693	10,266	3,019
Scenario	**Train**	**Val**	**Test**
Extractive	80,049	5,843	-
Multiple Choice	45,997	14,127	21,573
Abstractive	176,446	24,521	1,281
Total	302,492	44,491	22,854

Table 2: Training splits for HalluciBot.

the output a_i and the ground truth y, we assign $\mathbb{I}[a_i \neq y] \mapsto 1$; otherwise, $\mathbb{I}[a_i = y] \mapsto 0$. The results are compared to the baseline (original query q_0, output a_0), the mode (most common a_i), the lower bound (all correct), and the upper bound (at least one a_i correct).

How Do We Measure Agreement? *Accuracy* alone is insufficient for evaluating the *agreement* among multiple agents. To assess agreement, we report statistical measures for our Monte Carlo experiments including Item Difficulty ($\mu_{\mathbf{D}}$), Fleiss's Generalized κ, Mean Certainty / Entropy ($\mathbf{H}_\eta$), and Gibbs' $\mathbf{M_2}$ Index. These metrics help evaluate agreement levels amongst independent agents. For instance, high agreement on an incorrect answer indicates a misconception, while low agreement could suggest confusion or a poorly formulated query. To address this limitation in HalluciBot, we introduce a dual cross-entropy loss based on hallucination rates and consensus to improve the model's ability to distinguish good queries from bad queries.

3.2 Converting Monte Carlo Estimates To Labels

Empirical Estimate. The *probability of hallucination* for a query q_0, denoted as $p_h(q_0)$, can be empirically estimated based on the output $a_i \in \mathcal{A}$ of our Multi-Agent Monte Carlo simulation. We define the indicator function $\mathbb{I}$ to measure the incorrectness of an output a_i with respect to the ground truth y for query q_0.

$$p_h(q_0) \approx \frac{1}{n+1} \sum \mathbb{I}[a_i \neq y]$$

Binary Hallucination & Consensus Labels. To assess the propensity to hallucinate, we simplify the problem by considering two response values: *whether q_0 produces any hallucination or not*. Thus, we define the binary values for the probability of any hallucination as $p_b(q_0)$. Furthermore, we craft a secondary **consensus** label $p_c(q_0)$ that is a proxy for the agreement of the query. It maps the set of unique answers to 1 if there is any disagreement, otherwise we assign 0 for perfect agreement (1 unique answer). Therefore, we can train a 2 head output `Consensus` model to predict the hallucination probability $p_b(q_0)$, and if the query will cause confusion or consensus $p_c(q_0)$.

$$p_b(q_0) = \begin{cases} 1 & \text{if } p_h(q_0) > 0 \\ 0 & \text{if } p_h(q_0) = 0 \end{cases}$$

$$p_c(q_0) = \begin{cases} 1 & \text{if } \left|\{a_i \,|\, a_i \in \mathcal{A}\}\right| > 1 \\ 0 & \text{if } \left|\{a_i \,|\, a_i \in \mathcal{A}\}\right| = 1 \end{cases}$$

Scenario			Accuracy				Agreement			
	Experiment	#	Base ↑	Mode ↑	Lower ↑	Upper ↑	μ_D ↑	H_η ↑	M_2 ↑	κ ↑
SINGLE	Extractive	85,734	89.8	**90.3**	83.6	94.6	89.8	91.4	92.0	90.4
	Multiple Choice	80,813	74.0	**75.8**	58.1	88.0	73.8	90.3	83.7	75.5
	Abstractive	200,693	56.2	**56.7**	44.2	67.4	56.1	93.2	89.8	80.2
	Total	367,240	68.0	**68.7**	56.4	78.3	67.9	94.4	90.2	81.5
MULTI	Extractive	85,892	**92.1**	91.0	69.0	97.4	87.2	85.5	84.3	75.3
	Multiple Choice	81,697	76.3	**76.8**	47.4	91.6	71.8	75.2	71.3	61.9
	Abstractive	202,248	**55.9**	53.9	32.9	67.3	51.2	81.5	80.0	69.1
	Total	369,837	**68.6**	67.4	44.3	79.4	63.9	81.0	79.1	69.0

Table 3: Comparing **Single Query, Multiple Outputs (SINGLE)** vs. **Single Query, Multiple Perturbations, Single Output (MULTI)** Monte Carlo Experiments (§5). The reported metrics (§3.1) are calculated across all examples, regardless of the original dataset split. For the majority of scenarios, the SINGLE strategy outperforms the the MULTI approach in eliciting correct answers. Therefore, the SINGLE approach demonstrates higher agreement and tighter accuracy bounds, while the MULTI approach introduces more diverse responses and hallucinations with negligible impact on modal accuracy, allowing our simulation to generate more useful labels regarding query quality compared with a SINGLE approach.

	Accuracy ↑			F1 Score ↑			Precision ↑			Recall ↑		
Model	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
`RoBERTa-base`	74.7	64.1	66.1	73.3	66.5	69.6	85.1	78.0	74.4	64.4	57.9	65.3
`+ Scenario`	79.8	73.0	69.0	79.3	76.8	71.7	**88.8**	**81.5**	**78.4**	71.5	72.6	66.0
`+ Consensus`	79.3	73.0	68.7	79.1	77.0	71.5	87.2	81.0	77.7	71.4	73.3	66.2
`+ Calibration`	80.3	**73.6**	**69.5**	81.4	78.8	73.6	83.6	78.4	75.6	79.2	79.2	71.7
`+` $\tau = 0.341$	80.4	**73.6**	**69.5**	81.6	**80.2**	**76.0**	74.7	72.9	70.3	90.0	89.0	**82.6**
`RoBERTa-large`												
`+ Calibration`	84.7	73.5	69.2	**85.5**	78.5	73.0	88.1	78.9	76.1	83.1	78.2	70.1
`+` $\tau = 0.326$	**84.8**	**73.6**	69.4	83.5	80.0	75.6	75.0	71.8	70.5	**94.2**	**90.4**	81.6

Table 4: HalluciBot Binary Evaluation Statistics. We report the Accuracy, F1, Precision, and Recall for all data splits. Probability threshold τ is computed along the closed interval $[0, 1]$ in increments of 0.001 to maximize the validation F1 score for the final model. The best ablation per base model is underlined, while the overall best performing model is in **bold**.

3.3 How To Train a Classifier?

Once the Monte Carlo simulation is complete for our training corpus composed of 369,837 queries spanning 13 different datasets (Table 1), we start training our classifier. These queries encompass **Extractive**, **Multiple Choice**, and **Abstractive** scenarios. Each scenario, with or without additional context, affects the hallucination rate of `gpt-3.5-turbo`. These simulated estimates are directly proportional to the approximated rates of hallucination p_h.

- With a synthetic labeled set of queries q_0 and their rate of hallucinations $p_h(q_0)$, we train an encoder-style `RoBERTa` (Liu et al. 2019) classifier to estimate the hallucination probability density from our Monte Carlo simulation.
- We ablate two versions: a binary model to estimate the propensity a query can hallucinate, and a consensus-aware model to also predict the expected agreement of outputs if sampled $n + 1$ times.

Our experiments constrain the number of perturbations to $n = 5$, and when including the original query and output, we can model the hallucination rate for $n + 1 = 6$ modes. This translates to increments of $16.\overline{6}\%$ in hallucination rates.

How To Encode a Query's Scenario? We conduct an ablation study to explore if incorporating the query's scenario mitigates hallucinations. To create the prompt, we prepend the original query q_0 with either `[EXTRACTIVE]`, `[MULTIPLE CHOICE]`, or `[ABSTRACTIVE]`, using the format «`{tag} {q0}`». Our hypothesis is based on recent research that highlights the use of RAG (Lewis et al. 2020) to alleviate hallucinations. The additional context provides valuable signals related to the hallucination rate of the original query. Furthermore, we apply this technique to distinguish our experimental results from reused datasets in different scenarios, such as SciQ (Johannes Welbl 2017) and SQuADv2 (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018).

H4R: Downstream Modes For HalluciBot. Without HalluciBot's feedback, typical query rewriting models have to act as both an implicit critic and a generator. As a proxy reward model, HalluciBot's probabilistic feedback on a query's quality, given the dual prediction heads for hallucination and consensus, can guide a query rewriting process using an independent `gpt-3.5-turbo` LLM, *before output generation*. In essence, HalluciBot provides the following downstream modes for handling potentially hallucinatory queries:

1. **Rewrite Mode:** A single-shot iterative rewrite of queries classified as hallucinatory.
2. **Rank Mode:** Generating N intermediate perturbations, sorted by HalluciBot's class probabilities for fine-grained scoring. In our implementation, the number of outputs were controlled by the number of chat completion choices in `gpt-3.5-turbo`'s API call.
3. **Route Mode:** For **Abstractive** or **Extractive** queries classified as hallucinatory, testing if switching the scenario

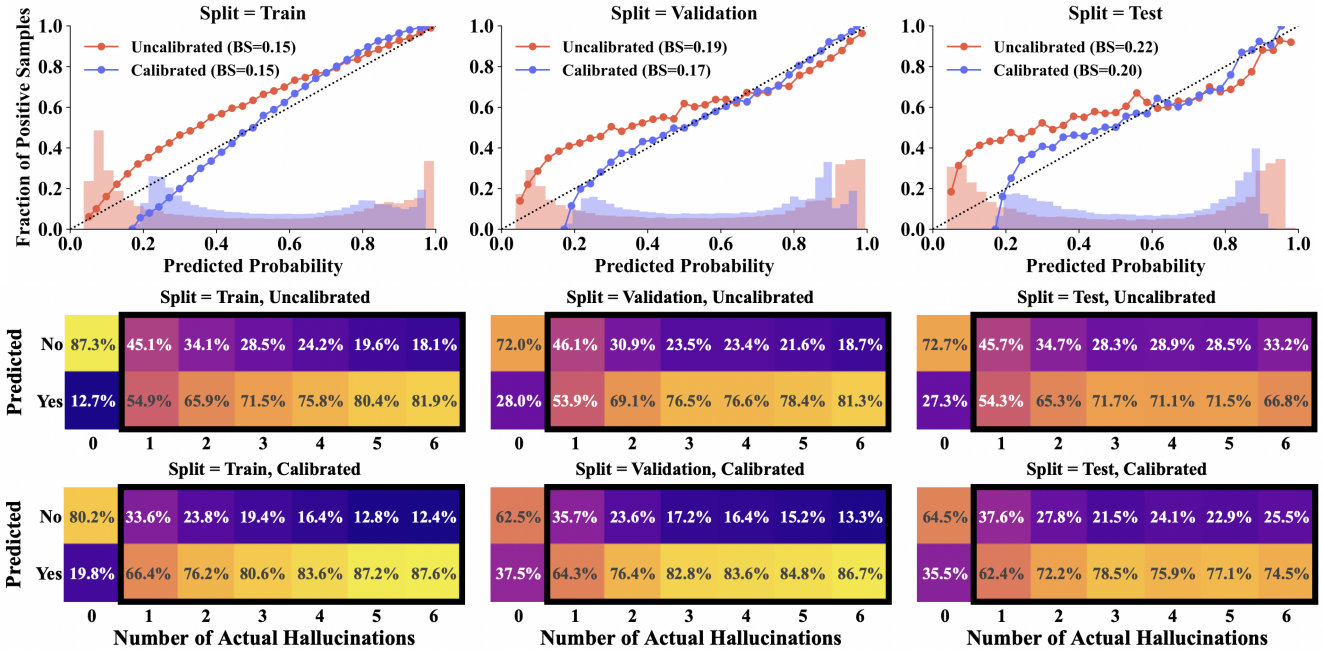

Figure 5: Top: HalluciBot calibration curves with Brier Scores (BS), alongside the histogram of predicted probabilities. Bottom: Predicted hallucination labels juxtaposed against observed hallucination rates during our Monte Carlo simulation, with calibrated matrix below. We highlight 1-6 as corresponding to the binary label "Yes - Hallucinatory" ($y = 1$) during training. Notably, there is significant confusion in queries that are borderline (1, 2) rather than majority hallucinatory prone (3-6).

(e.g. between RAG or direct inference) generates more robust classifications and generations.

4 Experimental Setup

Dataset Coverage & Scenario Split. Our experiments include 13 datasets (Table 1) divided into 3 scenarios: **Extractive**, **Multiple Choice**, and **Abstractive**. To evaluate the impact of context, we use SQuADv2 (Rajpurkar et al. 2016; Rajpurkar, Jia, and Liang 2018) to simulate RAG (Lewis et al. 2020). To assess the effect of multiple choice queries, we repurposed TruthfulQA (Lin, Hilton, and Evans 2022) and SciQ (Johannes Welbl 2017) for two experiments: one where the output agents select from the choices or context, and another where LLM agents generate outputs without context. We maintain the original train, validation, and test splits across scenarios to prevent information leakage to HalluciBot. All LLM agents share the same set of parameters.

HalluciBot Training Parameters & Environment. All experiments were conducted on an AWS EC2 instance with a single GPU. HalluciBot is fine-tuned from both pretrained `BERT` (Devlin et al. 2018) and `RoBERTa` (Liu et al. 2019) models. To address label imbalance, we employed a weighted class loss where each class weight is assigned to its inverted frequency in the training set. The train, validation, and test splits follow the original divisions of the datasets. Specifically, there are 302,492 training, 44,491 validation, and 22,854 testing samples. The distribution of labels across these splits is summarized in Table 2. We apply Platt calibration (Platt 1999) based on the validation logits to help ensure that the raw probabilities align better with the true class labels (Figure 5).

5 Analysis & Discussion

Ablation: Perturbations Induce Output Diversity. We examine the impact of perturbations on the robustness of `gpt-3.5-turbo` in question-answering tasks by comparing two strategies: **Single Query, Multiple Outputs (SINGLE)** and **Single Query, Multiple Perturbations, Single Output (MULTI)**. In the SINGLE strategy, we sample $n+1$ outputs from the original query q_0. In the MULTI strategy, n perturbations of the original query q_0 are used, and each perturbation q_i is answered once. Table 3 shows that while baseline accuracy remains consistent, the lower bound accuracy drops by 12.1 points in the MULTI setting. Additionally, agreement metrics, as indicated by Fleiss's κ, decrease by 12.5 points, indicating reduced consistency. In summary, (1) the SINGLE strategy results in higher agreement and lower-bound accuracy while (2) the MULTI strategy increases response diversity and hallucination rates but offers a slight improvement in upper-bound accuracy for **Extractive** and **Multiple Choice** scenarios. This suggests that perturbations can enhance query quality by introducing necessary diversity, despite minor variations in modal accuracy.

Ratiocinate: Can HalluciBot Detect Hallucinatory Queries? Differentiating the scenario in HalluciBot's prompt yielded a strong $+10.3\%$ increase in validation F1 score. The calibrated, threshold-tuned `RoBERTa-base` HalluciBot in Table 4 achieves a test accuracy of 69.5% with a macro F1-score of 76.0%. Further breaking down the results in Figure 5, calibrating our models with Platt scaling improves the discriminating power for borderline queries, where the observed number of hallucinations was minimal ($y \in \{1, 2\}$). Finally,

Metrics (%)	Test	Metrics (%)	Test
(A) Naive Rewrite		(D) HB Ratiocinate	
+ Class Transitions	6.5	+ Class Transitions	14.8
− Class Transitions	3.2	Rewrite Accuracy	
Unneeded Rewrites	46.6	Top-5 Accuracy	**92.9**
-	-	Similarity Score	**41.7**
(B) HB Informed Rewrite		(E) w/ Consensus	
+ Class Transitions	30.2	+ Class Transitions	31.9
Rewrite Accuracy		Rewrite Accuracy	
Top-5 Accuracy	**94.3**	Top-5 Accuracy	**90.2**
Similarity Score	**46.9**	Similarity Score	**57.5**
(C) HB Best-of-N Rewrite		(F) w/ Consensus	
+ Class Transitions	50.6	+ Class Transitions	51.4
Rewrite Accuracy		Rewrite Accuracy	
Top-5 Accuracy	**95.2**	Top-5 Accuracy	**95.7**
Similarity Score	**47.4**	Similarity Score	**55.9**

Table 5: Query generation metrics under each HalluciBot (HB) strategy. **Multiple Choice** queries use a soft accuracy criterion where the score is +1 if any of the n generations match the ground truth. **Abstractive** queries report the average cosine similarity score between the ground truth and the n generation outputs. Embedding vectors are computed using `all-MiniLM-L6-v2` (Wang et al. 2020).

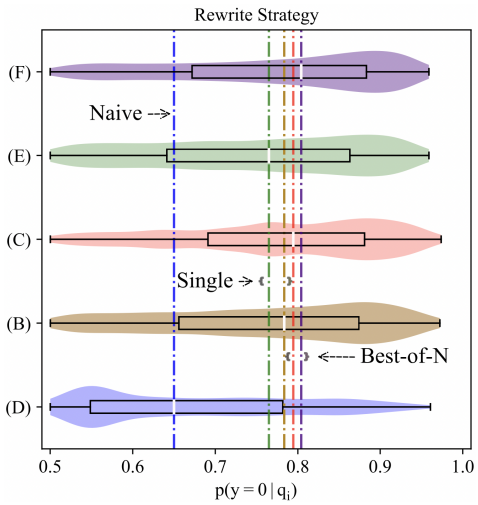

Figure 6: Class probability of queries that were rewritten and reclassified to be non-hallucinatory.

HalluciBot demonstrates strong recall scores (89.0% validation, 82.6% testing) to effectively flag risky queries that are likely to generate *at least one hallucination* during inference. The importance of HalluciBot as a ratiocinating process can be seen in [Table 5 (A)] under a naive rewriting strategy. Without HalluciBot, a naive rewrite strategy has the potential to convert queries originally estimated to be non-hallucinatory to hallucinatory (negative class transition), because there is no mechanism to differentiate queries. With HalluciBot restricting the test set to only potentially hallucinatory queries (11.2K samples), a naive rewrite [Table 5 (D)] can only enact positive class transitions (+14.8%), converting queries originally estimated to hallucinate to non-hallucinatory. Furthermore, HalluciBot acting as an arbitrator can prevent computationally expensive rewrite calls for 46.6% of the test set (10.2K samples deemed to be non-hallucinatory).

Rewrite: As a Feedback Mechanism. HalluciBot's feedback allows us to generate a more informed query [Table 5 (B)] resulting in better class transition probabilities than an uninformed rewrite strategy [Table 5 (D)]. This translates to a 14.8% positive class transition and a 1.4% increase in **Multiple Choice** accuracy as well as 5.2% improvement in generation similarity for **Abstractive** queries. Utilizing consensus information during the query rewriting process [Table 5 (E)] generates a slightly larger positive class transition (31.9% vs. 30.2%) than without [Table 5 (B)].

Rank: Best-of-N Rewrite. A Best-of-N rewrite strategy [Table 5 (C, F)] demonstrates a 19.5% and 20.4% gain in positive class transitions in both experiments over a single rewrite [Table 5 (B, E)]. Therefore, HalluciBot's estimated probabilities can be used as a proxy reward model when ranking n sample perturbations. The violin plots in Figure 6 shows that HalluciBot is able to select better queries in the Best-of-N rewrite setting than with a single rewrite, with a higher median predicted non-hallucinatory probability (79.5% (C) vs. 78.4% (B); 80.4% (F) vs. 76.5% (E)). This means that HalluciBot evaluated the rewritten queries to be non-hallucinatory with greater probability. All rewrites are single-shot without subsequent tuning or iterations.

Route: Abstractive to Extractive. Among 948 hallucinatory **Abstractive** queries, switching the scenario to **Extractive** led to a +60.0% positive class transition, compared to just 9.7% with rewriting alone. HalluciBot's ability to distinguish scenarios aids in determining whether direct inference or RAG is more effective for a particular query.

6 Conclusion

We propose a heretofore relatively unexplored realm of hallucination mitigation - predicting a query's hallucination probability. HalluciBot estimates this probability using a diverse training corpus, ensuring robustness across scenarios and domains. Institutions can leverage HalluciBot to measure LLM performance and user accountability through our H4R framework (**Ratiocinate**, **Rewrite**, **Rank**, **Route**). Thus, HalluciBot's contributions add to the ever-growing effort of enabling a robust language generation ecosystem for society.

References

Alzahrani, N.; Alyahya, H. A.; Alnumay, Y.; Alrashed, S.; Alsubaie, S.; Almushaykeh, Y.; Mirza, F.; Alotaibi, N.; Altwairesh, N.; Alowisheq, A.; Bari, M. S.; and Khan, H. 2024. When Benchmarks are Targets: Revealing the Sensitivity of Large Language Model Leaderboards. arXiv:2402.01781.

Amini, A.; Gabriel, S.; Lin, S.; Koncel-Kedziorski, R.; Choi, Y.; and Hajishirzi, H. 2019. MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2357–2367. Minneapolis, Minnesota: Association for Computational Linguistics.

Bisk, Y.; Zellers, R.; Bras, R. L.; Gao, J.; and Choi, Y. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, U.; Oprea, A.; and Raffel, C. 2021. Extracting Training Data from Large Language Models. arXiv:2012.07805.

Cho, N.; Srishankar, N.; Cecchi, L.; and Watson, W. 2024. FISHNET: Financial Intelligence from Sub-querying, Harmonizing, Neural-Conditioning, Expert Swarms, and Task Planning. In *Proceedings of the 5th ACM International Conference on AI in Finance*, ICAIF '24, 591–599. ACM.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2924–2936. Minneapolis, Minnesota: Association for Computational Linguistics.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv:1803.05457v1*.

Cole, J.; Zhang, M.; Gillick, D.; Eisenschlos, J.; Dhingra, B.; and Eisenstein, J. 2023. Selectively Answering Ambiguous Questions. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 530–543. Singapore: Association for Computational Linguistics.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805.

Dohan, D.; Xu, W.; Lewkowycz, A.; Austin, J.; Bieber, D.; Lopes, R. G.; Wu, Y.; Michalewski, H.; Saurous, R. A.; Sohl-dickstein, J.; Murphy, K.; and Sutton, C. 2022. Language Model Cascades. arXiv:2207.10342.

Gao, L.; Tow, J.; Abbasi, B.; Biderman, S.; Black, S.; DiPofi, A.; Foster, C.; Golding, L.; Hsu, J.; Le Noac'h, A.; Li, H.; McDonell, K.; Muennighoff, N.; Ociepa, C.; Phang, J.; Reynolds, L.; Schoelkopf, H.; Skowron, A.; Sutawika, L.; Tang, E.; Thite, A.; Wang, B.; Wang, K.; and Zou, A. 2023. A framework for few-shot language model evaluation.

Google. 2023. Introducing Gemini: our largest and most capable AI model. https://blog.google/technology/ai/google-gemini-ai/. Accessed: 2024-06-15.

Hendrycks, D.; Burns, C.; Basart, S.; Critch, A.; Li, J.; Song, D.; and Steinhardt, J. 2021a. Aligning AI With Shared Human Values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021b. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2020. The Curious Case of Neural Text Degeneration. arXiv:1904.09751.

Holtzman, A.; Buys, J.; Forbes, M.; Bosselut, A.; Golub, D.; and Choi, Y. 2018. Learning to Write with Cooperative Discriminators. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1638–1649. Melbourne, Australia: Association for Computational Linguistics.

Huang, L.; Yu, W.; Ma, W.; Zhong, W.; Feng, Z.; Wang, H.; Chen, Q.; Peng, W.; Feng, X.; Qin, B.; and Liu, T. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. arXiv:2311.05232.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12): 1–38.

Jiang, Z.; Araki, J.; Ding, H.; and Neubig, G. 2021. How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 9: 962–977.

Johannes Welbl, M. G., Nelson F. Liu. 2017. Crowdsourcing Multiple Choice Science Questions.

Joshi, M.; Choi, E.; Weld, D.; and Zettlemoyer, L. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, arXiv:1705.03551.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; Johnston, S.; El-Showk, S.; Jones, A.; Elhage, N.; Hume, T.; Chen, A.; Bai, Y.; Bowman, S.; Fort, S.; Ganguli, D.; Hernandez, D.; Jacobson, J.; Kernion, J.; Kravec, S.; Lovitt, L.; Ndousse, K.; Olsson, C.; Ringer, S.; Amodei, D.; Brown, T.; Clark, J.; Joseph, N.; Mann, B.; McCandlish, S.; Olah, C.; and Kaplan, J. 2022. Language Models (Mostly) Know What They Know. arXiv:2207.05221.

Khashabi, D.; Min, S.; Khot, T.; Sabharwal, A.; Tafjord, O.; Clark, P.; and Hajishirzi, H. 2020. UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1896–1907. Online: Association for Computational Linguistics.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2023. Large Language Models are Zero-Shot Reasoners. arXiv:2205.11916.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 9459–9474. Curran Associates, Inc.

Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; Newman, B.; Yuan, B.; Yan, B.; Zhang, C.; Cosgrove, C.; Manning, C. D.; Ré, C.; Acosta-Navas, D.; Hudson, D. A.; Zelikman, E.; Durmus, E.; Ladhak, F.; Rong, F.; Ren, H.; Yao, H.; Wang, J.; Santhanam, K.; Orr, L.; Zheng, L.; Yuksekgonul, M.; Suzgun, M.; Kim, N.; Guha, N.; Chatterji, N.; Khattab, O.; Henderson, P.; Huang, Q.; Chi, R.; Xie, S. M.; Santurkar, S.; Ganguli, S.; Hashimoto, T.; Icard, T.; Zhang, T.; Chaudhary, V.; Wang, W.; Li, X.; Mai, Y.; Zhang, Y.; and Koreeda, Y. 2022. Holistic Evaluation of Language Models. arXiv:2211.09110.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 3214–3252. Dublin, Ireland: Association for Computational Linguistics.

Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; and Chen, W. 2021. What Makes Good In-Context Examples for GPT-3? arXiv:2101.06804.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Ma, X.; Gong, Y.; He, P.; Zhao, H.; and Duan, N. 2023. Query Rewriting for Retrieval-Augmented Large Language Models. arXiv:2305.14283.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Gupta, S.; Majumder, B. P.; Hermann, K.; Welleck, S.; Yazdanbakhsh, A.; and Clark, P. 2023. Self-Refine: Iterative Refinement with Self-Feedback. arXiv:2303.17651.

Manakul, P.; Liusie, A.; and Gales, M. J. F. 2023. SelfCheckGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models. arXiv:2303.08896.

Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Conference on Empirical Methods in Natural Language Processing*.

OpenAI. 2022. Introducing ChatGPT. https://openai.com/blog/chatgpt. Accessed: 2024-06-15.

Peng, B.; Galley, M.; He, P.; Cheng, H.; Xie, Y.; Hu, Y.; Huang, Q.; Liden, L.; Yu, Z.; Chen, W.; and Gao, J. 2023. Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback. arXiv:2302.12813.

Petroni, F.; Piktus, A.; Fan, A.; Lewis, P.; Yazdani, M.; De Cao, N.; Thorne, J.; Jernite, Y.; Karpukhin, V.; Maillard, J.; Plachouras, V.; Rocktäschel, T.; and Riedel, S. 2021. KILT: a Benchmark for Knowledge Intensive Language Tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2523–2544. Online: Association for Computational Linguistics.

Platt, J. 1999. Probabilistic Outputs for Support vector Machines and Comparisons to Regularized Likelihood Methods.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf. Accessed: 2024-06-15.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. arXiv:1806.03822.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. arXiv:1606.05250.

Snyder, B.; Moisescu, M.; and Zafar, M. B. 2023. On Early Detection of Hallucinations in Factual Question Answering. arXiv:2312.14183.

Swaminathan, P. 2021. Monte Carlo simulations as a route to compute probabilities. arXiv:2108.00851.

Tonmoy, S. M. T. I.; Zaman, S. M. M.; Jain, V.; Rani, A.; Rawte, V.; Chadha, A.; and Das, A. 2024. A Comprehensive Survey of Hallucination Mitigation Techniques in Large Language Models. arXiv:2401.01313.

Varshney, N.; Yao, W.; Zhang, H.; Chen, J.; and Yu, D. 2023. A Stitch in Time Saves Nine: Detecting and Mitigating Hallucinations of LLMs by Validating Low-Confidence Generation. arXiv:2307.03987.

Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. *arXiv preprint 1905.00537*.

Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. arXiv:2002.10957.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. arXiv:2203.11171.

Watson, W.; Cho, N.; Balch, T.; and Veloso, M. 2023. HiddenTables and PyQTax: A Cooperative Game and Dataset For TableQA to Ensure Scale and Data Privacy Across a Myriad of Taxonomies. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 7144–7159. Singapore: Association for Computational Linguistics.

Watson, W.; Cho, N.; Srishankar, N.; Zeng, Z.; Cecchi, L.; Scott, D.; Siddagangappa, S.; Kaur, R.; Balch, T.; and Veloso, M. 2025. LAW: Legal Agentic Workflows for Custody and Fund Services Contracts. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, 583–594. Abu Dhabi, UAE: Association for Computational Linguistics.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Wu, T.; Jiang, E.; Donsbach, A.; Gray, J.; Molina, A.; Terry, M.; and Cai, C. J. 2022. PromptChainer: Chaining Large Language Model Prompts through Visual Programming. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22. New York, NY, USA: Association for Computing Machinery. ISBN 9781450391566.

Yang, Y.; Yih, W.-t.; and Meek, C. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2013–2018. Lisbon, Portugal: Association for Computational Linguistics.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2369–2380. Brussels, Belgium: Association for Computational Linguistics.

Zeng, Z.; Watson, W.; Cho, N.; Rahimi, S.; Reynolds, S.; Balch, T.; and Veloso, M. 2024. FlowMind: Automatic Workflow Generation with LLMs. arXiv:2404.13050.

Zheng, H.; and Saparov, A. 2023. Noisy Exemplars Make Large Language Models More Robust: A Domain-Agnostic Behavioral Analysis. arXiv:2311.00258.