CTRLA: Adaptive Retrieval-Augmented Generation via Inherent Control

Huanshuo Liu[1♣], **Hao Zhang**[1♣♠], **Zhijiang Guo**[1♠], **Jing Wang**[2]
Kuicai Dong[1], **Xiangyang Li**[1], **Yi Quan Lee**[1], **Cong Zhang**[1], **Yong Liu**[1]
[1]Noah's Ark Lab, Huawei Technologies Co., Ltd
[2] Independent Researcher
hzhang26@outlook.com, cartusguo@gmail.com

Abstract

Retrieval-augmented generation (RAG) has emerged as a promising solution for mitigating hallucinations of Large Language Models (LLMs) with retrieved external knowledge. Adaptive RAG enhances this approach by enabling dynamic retrieval during generation, activating retrieval only when the query exceeds LLM's internal knowledge. Existing methods primarily focus on detecting LLM's confidence via statistical uncertainty. Instead, we present the first attempts to solve adaptive RAG from a representation perspective and develop an inherent control-based framework, termed CTRLA. Specifically, we extract the features that represent the honesty and confidence directions of LLM and adopt them to control LLM behavior and guide retrieval timing decisions. We also design a simple yet effective query formulation strategy to support adaptive retrieval. Experiments show that CTRLA is superior to existing adaptive RAG methods on diverse tasks. Honesty steering can effectively make LLMs more honest, and confidence monitoring is a promising indicator of retrieval trigger.[1]

1 Introduction

Retrieval-augmented generation (RAG; Guu et al. 2020; Izacard et al. 2023) has proven effective in mitigating hallucination by integrating external knowledge into LLMs. Early efforts often employ single-round, indiscriminate retrieval, resulting in over-reliance on external knowledge and incomplete retrieval (Wang et al., 2023; Su et al., 2025). To solve the issues, adaptive RAG (ARAG; Jiang et al. 2023b; Wang et al. 2024a) has emerged, which enables dynamic retrieval during generation, activating retrieval only when the query exceeds LLM's internal knowledge (Ni et al., 2024).

The key challenges in ARAG involve determining *what* and *when* to retrieve (Su et al., 2024; Yao et al., 2024; Zhang et al., 2024a). The design of *what* aspect depends on the construction of *when* aspect, making ARAG's primary focus the issues related to *when* aspect. For the *when* aspect, recent ARAGs leverage the ability that LLMs are aware of their uncertainty (Kuhn et al., 2023; Chen et al., 2024; Xiong et al., 2024), utilizing this characteristic to determine retrieval timing by assessing *confidence* level of their knowledge (Su et al., 2024; Yao et al., 2024). They primarily focus on detecting uncertainty in the LLM's outputs to signal retrieval, relying on factors such as output probabilities (Jiang et al., 2023b), entropy of output (Su et al., 2024) or internal states (Yao et al., 2024), or verbal feedback (Wang et al., 2025; Yan et al., 2024). From a statistical standpoint, uncertainty and confidence are conceptually equivalent, both reflect the degree of certainty in a model's predictions (Yang et al., 2023; Band et al., 2024; Tao et al., 2024). Thus, uncertainty can act as a proxy for confidence when determining retrieval timing.

We revisit the assumptions underlying these uncertainty-based methods. First, they presume that LLM's output aligns with its internal knowledge (Lin et al., 2022; Zou et al., 2023; Xiao et al., 2025), *i.e.,* LLMs can accurately reflect internal knowledge in outputs or they are *honest*. However, LLMs often navigate a trade-off between honesty and helpfulness, balancing between discerning its limitations and generating user-satisfied plausible content (Liu et al., 2024a). When the output diverges from internal knowledge, indicating low honesty, they only detect intended output rather than internal knowledge. Second, they equate uncertainty with LLM's *confidence*,[2] which may not always apply to LLM behavior. For instance, an

[♣] The first two authors contributed equally.

[♠] Corresponding authors.

[1] https://github.com/HSLiu-Initial/CtrlA

[2] Confidence is the feeling of belief or trust that a person or thing is reliable (Bandura, 1997).

Findings of the Association for Computational Linguistics: ACL 2025, pages 12592–12618
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

LLM may frequently respond with “I don’t know” or “insufficient information,” suggesting low uncertainty, yet retrieval should still occur. Moreover, semantically equivalent answers can be expressed in various ways in free-form generation, which may lead to high uncertainty (Farquhar et al., 2024). However, retrieval is unnecessary in this scenario.

Based on this analysis, we emphasize that both *honesty* and *confidence* of LLMs are crucial for accurate retrieval timing. However, current ARAGs struggle to address them due to the limitations of statistical uncertainty. We propose to solve ARAG from a representation perspective (Olah, 2023; Bricken et al., 2023; Zou et al., 2023; Templeton et al., 2024), developing an efficient and unified framework that seamlessly tackles the requirements of honesty and confidence. Our core idea involves extracting features corresponding to *honesty* and *confidence* directions from LLMs and using them to control LLM behavior and guide retrieval timing decisions simultaneously.

We devise an Inherent **Contr**o**l**-based **A**daptive RAG (**CTRLA**). To steer LLM toward honesty and monitor its confidence, we extract features aligned with the directions of honesty and confidence within LLM’s representation space. By adjusting the honesty direction-a process we refer to as ***honesty steering***-we can shift LLM’s representation space to promote more honest outputs. Simultaneously, confidence is quantified by measuring the projection of current representation onto the confidence feature, a method we call ***confidence monitoring***. Honesty steering helps LLM recognize its limitations and suppress the generation of fabricated, plausible information. Confidence monitoring, in turn, enhances the precision of retrieval timing. Experiments verify the effectiveness of CTRLA, revealing that adjusting the directions of LLM’s internal states enhances its honesty, while confidence monitoring reliably signals when to trigger retrieval, optimizing the balance between retrieval and internal knowledge use.

2 Related Work

Early RAG efforts (Lewis et al., 2020; Karpukhin et al., 2020; Zhu et al., 2021; Komeili et al., 2022; Khattab et al., 2023; Qi et al., 2024) relied on single-round, indiscriminate retrieval, increasing computational costs and degrading performance (Wang et al., 2023; Su et al., 2025). To address it, ARAG emerged, enabling dynamic retrieval during generation when the query exceeds LLM’s internal knowledge (Jiang et al., 2023b; Wang et al., 2024a; Ni et al., 2024). Previous implementations utilized static rules, such as prior sentences (Trivedi et al., 2023), sliding windows (Borgeaud et al., 2022; Ram et al., 2023), and in-context learning (Zhao et al., 2023; Zhang et al., 2024b; Li et al., 2024). Recent ARAGs leverage LLMs’ self-awareness of uncertainty to optimize retrieval timing by assessing confidence levels through internal states (Yao et al., 2024), likelihoods (Jiang et al., 2023b; Su et al., 2024; Jin et al., 2024a), or verbal feedback (Wang et al., 2025; Ding et al., 2024; Yan et al., 2024). This enhances retrieval timing and balances external and internal knowledge use. However, uncertainty-based ARAGs face challenges with LLM honesty and confidence, crucial for accurate retrieval timing. CTRLA solves these issues from a representation perspective, enhancing control over honesty and confidence to improve the effectiveness.

3 Inherent Control based Adaptive RAG

3.1 Preliminary

Given a query $\boldsymbol{q}$, RAG aims to assist LLMs in generating more precise answers $\boldsymbol{y} = [s_1, \dots, s_m] = [w_1, \dots, w_n]$ containing m sentences or n tokens by retrieving relevant documents $\mathcal{D}_q = \mathcal{R}(\boldsymbol{q})$ from document corpus $\mathcal{D} = \{\boldsymbol{d}_i\}_{i=1}^{|\mathcal{D}|}$ or web via retriever $\mathcal{R}$. The retrieved documents $\mathcal{D}_q$ are usually concatenated with input $\boldsymbol{x}$, *i.e.,* query $\boldsymbol{q}$ with task instruction $\mathcal{I}$, to aid answer generation as $\boldsymbol{y} = \texttt{LLM}([\mathcal{D}_q; \boldsymbol{x}])$, where $[\cdot;\cdot]$ denotes concatenation. In contrast, adaptive RAG performs active retrieval necessity decision via a trigger mechanism $\mathcal{T}(\boldsymbol{x}, \boldsymbol{y}_{<t})$, where $\boldsymbol{y}_{<t}$ is the prior generations as of step $t (t \geq 1)$. If $\mathcal{T}$ is triggered, the query formulation function $\boldsymbol{q}_t = f_q(\boldsymbol{x}, \boldsymbol{y}_{<t})$ will produce a query $\boldsymbol{q}_t$ to search. If $\mathcal{T}$ is triggered at $t = 1$, *i.e.,* $\boldsymbol{y}_{<1} = \emptyset$, $\boldsymbol{q}$ will be original query. Given the retrieved documents $\mathcal{D}_{q_t}$, the model continues generating next output segment (usually, a sentence) $\boldsymbol{y}_t = \texttt{LLM}([\mathcal{D}_{q_t}; \boldsymbol{x}; \boldsymbol{y}_{<t}])$ till the answer comes to its end or next retrieval trigger occurs.

3.2 CTRLA Framework

3.2.1 Representation Feature Extraction

Our approach builds on the linear representation and superposition hypotheses (Olah, 2023; Bricken et al., 2023; Templeton et al., 2024). We aim to extract features that represent *honesty* and *confidence*

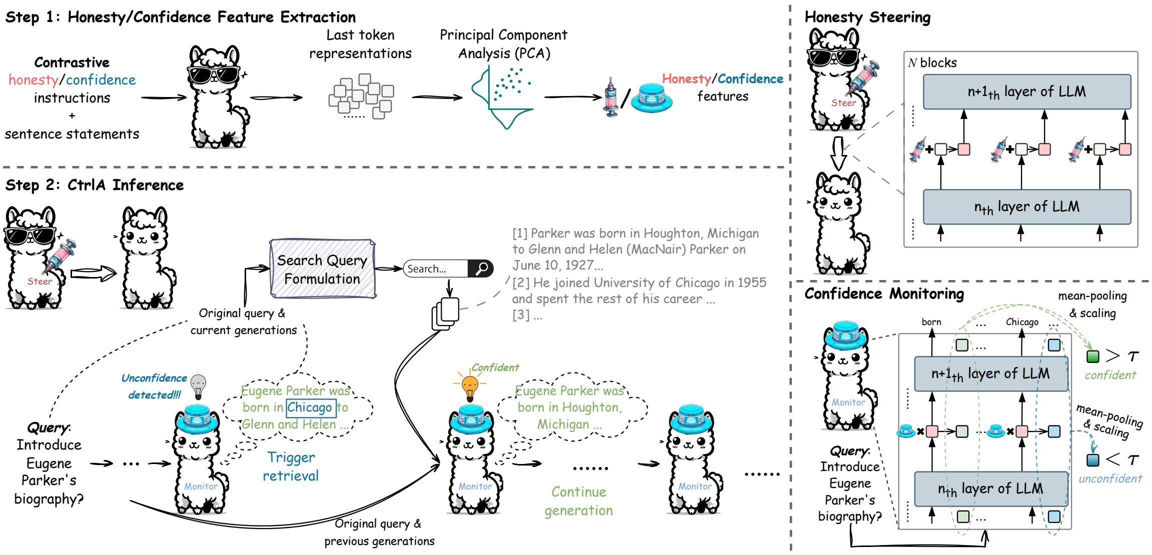

Step 1: Honesty/Confidence Feature Extraction
Contrastive honesty/confidence instructions + sentence statements
Last token representations
Principal Component Analysis (PCA)
Honesty/Confidence features
Step 2: CtrlA Inference
Steer
Search Query Formulation
Search...
[1] Parker was born in Houghton, Michigan to Glenn and Helen (MacNair) Parker on June 10, 1927...
[2] He joined University of Chicago in 1955 and spent the rest of his career ...
[3] ...
Original query & current generations
Unconfidence detected!!!
Eugene Parker was born in Chicago to Glenn and Helen ...
Trigger retrieval
Confident
Eugene Parker was born in Houghton, Michigan ...
Continue generation
Query: Introduce Eugene Parker's biography?
Monitor
Original query & previous generations
Honesty Steering
Steer
N blocks
n+1th layer of LLM
nth layer of LLM
Confidence Monitoring
mean-pooling & scaling
born
Chicago
> τ
confident
mean-pooling & scaling
< τ
unconfident
Monitor
Query: Introduce Eugene Parker's biography?

Figure 1: CTRLA framework. Step 1 extracts the features corresponding to *honesty* and *confidence* directions; Step 2 utilizes extracted features to steer and monitor LLM behaviors at inference. The *honesty* feature **steers** the representation of LLM to make it more honest, while *confidence* feature is used to **monitor** the confidence level of LLM outputs, where the token whose score is lower than the threshold is marked as unconfident. The retrieval is triggered if specific tokens are unconfident.

directions from LLM's representation space and use them to steer or monitor its behavior. Specifically, we manually craft contrastive instructions, as shown in Prompt 3.1, to extract features that represent the directions of honesty and confidence. Let $\mathcal{I}^{+}_{h/c}$ denote the positive instruction of honest or confident, $\mathcal{I}^{-}_{h/c}$ be the negative instruction of dishonest or unconfident, and $\mathcal{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{|\mathcal{S}|}\}$ represent the dataset with $|\mathcal{S}|$ statements used to extract target features.

For honesty feature extraction, each statement s_i is concatenated with both positive and negative instructions, forming $\mathcal{I}^{+}_{h} \oplus s_i$ and $\mathcal{I}^{-}_{h} \oplus s_i$, respectively, resulting in $|\mathcal{S}|$ statement pairs. For the statement pair of s_i, they are sequentially fed into LLM in a teacher-forcing manner to collect token representations. Given that each LLM layer encodes a unique semantic space (Chuang et al., 2024b; Sun et al., 2025), we extract token representations from all LLM layers. Assuming LLM has L layers and $\boldsymbol{s}_i$ contains n tokens, we obtain representations $\{\{\boldsymbol{r}^{l,+}_{i,k}\}^{n}_{k=1}\}^{L}_{l=1}$ and $\{\{\boldsymbol{r}^{l,-}_{i,k}\}^{n}_{k=1}\}^{L}_{l=1}$ for positive and negative instructions, where $\boldsymbol{r}^{l}_{i,k}$ denotes the k-th token representation of s_i at layer l. The contrastive vector for the k-th token at l-th layer is computed as $\boldsymbol{v}^{l}_{i,k} = \boldsymbol{r}^{l,+}_{i,k} - \boldsymbol{r}^{l,-}_{i,k}$. Thus, after processing all tokens of s_i, we derive the set of contrastive vectors $\{\{\boldsymbol{v}^{l}_{i,k}\}^{n}_{k=1}\}^{L}_{l=1}$. Since we employ *teacher-forcing* to encode each token's representation—and "honest" and "dishonest" are the only differing descriptions between $\mathcal{I}^{+}_{h}$ and $\mathcal{I}^{-}_{h}$—the vector $\boldsymbol{v}^{l}_{i,k}$ captures the honesty direction for the k-th token at layer l (Zou et al., 2023). After processing all statements in $\mathcal{S}$, we apply PCA to the collected contrastive vectors $\{\{\{\boldsymbol{v}^{l}_{i,k}\}^{n}_{k=1}\}^{L}_{l=1}\}^{|\mathcal{S}|}_{i=1}$ at each layer l, extracting the first principal component as the general honesty direction. This results in a set of honesty direction vectors $\boldsymbol{v}_h = \{\boldsymbol{v}^{l}_{h}\}^{L}_{l=1}$. Note that confidence feature extraction also utilizes the same method to derive $\boldsymbol{v}_c = \{\boldsymbol{v}^{l}_{c}\}^{L}_{l=1}$.

Prompt 3.1: Instruction for Honesty and Confidence Feature Extraction

[INST] `Pretend you are a <honest/dishonest> | <confident/unconfident> person making statements about the world.` **[/INST]** `<a statement` s_i`>`

We use the True-False dataset (Azaria and Mitchell, 2023) as $\mathcal{S}$ for honesty feature extraction, which tests whether LLMs' internal states reflect truthfulness. For confidence, we synthesize confident and unconfident statements using GPT-4 (ref. Appendix B.1) due to the scarcity of datasets.

3.2.2 Honesty Steering

According to the superposition hypothesis, adjusting LLM by moving each token's representation closer to the direction representing the honesty feature during decoding is a direct way to enhance its honesty (Olah, 2023; Zou et al., 2023; Templeton et al., 2024). To achieve this, we employ a simple linear combination. After extracting the honesty feature, it can be directly used to steer the behavior of the LLM. Assuming the LLM contains L layers, each layer has its corresponding feature. Let $\boldsymbol{v}_h = \{\boldsymbol{v}_h^l\}_{l=1}^L$ denote the honesty feature and $\boldsymbol{R}_k = \{\boldsymbol{r}_k^l\}_{l=1}^L$ represent the token representations for the k-th token at each layer. We then apply a linear combination function for honesty steering:

$$\hat{\boldsymbol{R}}_k = \boldsymbol{R}_k + \lambda \cdot \boldsymbol{v}_h = \{\boldsymbol{r}_k^l + \lambda \cdot \boldsymbol{v}_h^l \mid \forall\, l \in [1, \dots, L]\}, \tag{1}$$

where the coefficient λ controls the strength of honesty steering. Because $\boldsymbol{v}_h$ represents the direction that promotes honesty, the "+" operator is used in Eq 1. Conversely, to reduce honesty, the "−" operator can be employed. As illustrated in Figure 1, honesty steering is applied layer-by-layer and token-by-token during generation. This method is both simple and effective, with minimal impact on inference costs. We denote honesty steering as $\hat{\boldsymbol{y}}_t = \mathcal{P}_h(\boldsymbol{y}_t)$ in the following descriptions.

3.2.3 Confidence Monitoring

According to the linear representation hypothesis, an intuitive way to monitor the LLM's confidence during generation is to evaluate how well token representations align with the confidence feature direction in the representation space (Bricken et al., 2023; Zou et al., 2023; Templeton et al., 2024). Given the extracted confidence feature, we utilize it to monitor LLM's confidence during generation. Let $\boldsymbol{R}_k = \{\boldsymbol{r}_k^l\}_{l=1}^L$ represent the k-th token's representation at each layer, and $\boldsymbol{v}_c = \{\boldsymbol{v}_c^l\}_{l=1}^L$ denote the confidence feature. Specifically, we compute the confidence score for k-th token using the dot product, followed by mean-pooling across layers and a scaling operation for normalization and outlier removal. This produces the confidence score for the k-th token as follows:

$$\begin{aligned} \tilde{m}_k &= \texttt{meanpool}([m_{k,1}, \dots, m_{k,L}]) \\ &= \texttt{meanpool}\big([\boldsymbol{r}_k^{l,\top} \cdot \boldsymbol{v}_c^l]_{l=1}^L\big), \\ \bar{m}_k &= \texttt{scale}([\tilde{m}_0, \dots, \tilde{m}_k])[-1] - \tau, \end{aligned} \tag{2}$$

where τ is the threshold to adjust the sensitivity of confidence monitoring, $\tilde{m}_{<k}$ represents the mean-pooled score of preceding tokens, and the index -1 refers to the score of the last token, *i.e., k*-th token. If $\bar{m}_k > 0$, it suggests that the k-th token's representational direction leans towards the confidence, indicating that LLM is confident in generating this token. Conversely, if $\bar{m}_k < 0$, LLM is unconfident in generating the k-th token. Here we denote confidence monitoring as $\mathcal{P}_c$.

The goal of confidence monitoring is to serve as a reliable detector for accurately determining appropriate retrieval timing (Wu et al., 2024a; Chuang et al., 2024a). For the t-th output segment $\hat{\boldsymbol{y}}_t = [w_{t_s}, \dots, w_{t_e}]$ of the LLM, with confidence scores $[\bar{m}_{t_s}, \dots, \bar{m}_{t_e}]$ for each token, the retrieval necessity is measured by the confidence scores of specific tokens within $\hat{\boldsymbol{y}}_t$. We only consider the confidence scores of *new information* in $\hat{\boldsymbol{y}}_t'$, *i.e.,* content that has not appeared in the previous generation and excludes trivial tokens, like stopwords. The retrieval trigger $\mathcal{T}$ activates if any confidence score in $\hat{\boldsymbol{y}}_t'$ satisfies $\bar{m}_k < 0$, where $t_s \le k \le t_e$. If $\hat{\boldsymbol{y}}_t'$ contains such tokens, retrieval is triggered, *i.e.,* $\mathcal{T}(\mathcal{P}_c(\hat{\boldsymbol{y}}_t')) == \texttt{True}$.

Due to the honesty steering, LLM will generate refusal outputs more frequently, since honesty steering can effectively regulate LLM behavior to make it more honest, leading to more frequent generation of non-responsive or refusal outputs. These refusal responses are well-aligned with the LLM's internal beliefs, *i.e.,* LLM is confident in its knowledge limitations, making them challenging to detect by confidence monitoring. To address this issue, we further develop a refusal handling module, which employs a pattern matching function, as a supplement to confidence monitoring, to identify refusal content. The detailed algorithm is presented in Appendix A.3.1.

3.2.4 Search Query Formulation

Once retrieval is triggered, we need to employ a search query to retrieve relevant documents that aid in LLM generation. The construction of effective search queries plays a pivotal role in enhancing retrieval efficiency (Jiang et al., 2023b). We develop two search query formulation strategies.

Context-Augmented Querying. Initially, for a query $\boldsymbol{q}$, we prompt the LLM to sequentially generate responses. Once the retrieval is triggered, context-augmented querying (CAQ) will concatenate the query $\boldsymbol{q}$ with the processed output segment $\hat{\boldsymbol{y}}_t$ for retrieval, since using the original query as a supplement can avoid intent drift and improve the effectiveness of retrieval (Jagerman et al., 2023). Besides, the output segment $\hat{\boldsymbol{y}}_t = [w_{t_s}, \dots, w_{t_e}]$

$$\texttt{mask}(\hat{\boldsymbol{y}}_t) = \left\{ \bar{w} \middle| \bar{w} = \begin{cases} \emptyset, & \text{if } w \notin \boldsymbol{q} \cup \boldsymbol{y}_{<t} \text{ and } \bar{m}_w < 0 \\ w, & \text{otherwise} \end{cases}, \forall w \in \hat{\boldsymbol{y}}_t \right\}. \tag{3}$$

may contain noise such as unconfident tokens and incorrect contents, we process the sentence by masking out the tokens, which satisfy (i) not appeared in $\boldsymbol{q}$ and previous generations $\boldsymbol{y}_{<t}$, *i.e.,* new information and (ii) unconfident tokens, as shown in Equation 3. Thus, the CAQ generates the refined search query as $f_{\text{CAQ}}(\boldsymbol{x}, \hat{\boldsymbol{y}}_t) = [\boldsymbol{q}; \texttt{mask}(\hat{\boldsymbol{y}}_t)]$.

Targeted Validation Querying. CAQ directly masks out the noise of the output segment and concatenates it with the original query to form a search query. Yet, off-the-shelf retrievers may prefer a well-formatted query (Karpukhin et al., 2020). Thus, we also develop a targeted validation querying strategy (TVQ), f_{TVQ}. It instructs LLM to produce a search query using the original query and the current output segment as references (see Prompt A.1). The goal of TVQ is to generate a query to validate the accuracy of the current output segment by searching for supporting documents. For simplicity, we use f_q to represent both f_{CAQ} and f_{TVQ}.

3.3 Inference Process

For an input $\boldsymbol{x}$ and preceding generation $\boldsymbol{Y}_{<t}$, the model generates the output segment along with honesty steer $\mathcal{P}_h$ and derives $\hat{\boldsymbol{y}}_t$. Simultaneously, the confidence monitor $\mathcal{P}_c$ is activated to compute the confidence score of each token during generation. We collect the confidence scores of new information $\hat{\boldsymbol{y}}'_t$ to determine retrieval necessity via retrieval trigger $\mathcal{T}$. If retrieval is not required, the model continues predicting the next output segment. Otherwise, we adopt query formulation, f_q, to produce a search query $\boldsymbol{q}_t$ and retrieve documents $\mathcal{D}_q$ via retriever $\mathcal{R}$. The retrieved documents $\mathcal{D}_q$, input $\boldsymbol{x}$, and preceding generation $\boldsymbol{Y}_{<t}$ are concatenated to regenerate the current output segment. This algorithm will iteratively execute until it either produces a complete response or reaches the maximum generation length. Details of the algorithm with refusal handling are presented in Appendix A.3.2.

Method	TriviaQA	PopQA
wo-RAG$^{\diamond}_{7\text{B}}$	53.8	25.7
SR-RAG$^{\diamond}_{7\text{B}}$	62.7	51.9
FL-RAG$^{\diamond}_{7\text{B}}$	60.8	28.1
FS-RAG$^{\diamond}_{7\text{B}}$	54.3	26.9
QD-RAG$^{\diamond}_{7\text{B}}$	52.3	29.4
FLARE$^{\diamond}_{7\text{B}}$	<u>72.4</u>	48.3
Self-RAG$^{\ddagger}_{7\text{B}}$	66.4	54.9
Self-RAG$^{\ddagger}_{13\text{B}}$	69.3	55.8
RQ-RAG$^{\ddagger}_{7\text{B}}$	-	<u>57.1</u>
QC-RAG$^{\ddagger}_{11\text{B}}$	58.2	-
CTRLA$_{7\text{B}}$	**76.4**	**61.8**

Table 1: Results of short-form QA. $^{\diamond}$ is our reproduced results. ‡ denotes results in the corresponding work.

4 Experiment Setup

Datasets and Evaluation. For *short-form* QA, we select PopQA (Mallen et al., 2023) and TriviaQA (Joshi et al., 2017). For *long-form* QA, we use ASQA (Stelmakh et al., 2022) and biography generation (Bio; Min et al. 2023). For *multi-hop* QA, we follow Su et al. (2024) to choose 2WikiMultihopQA (2WMQA; Ho et al. 2020) and HotpotQA (HQA; Yang et al. 2018). For short-form QA, we report the accuracy. For ASQA, we report str-em, Rouge-L (R-L; Lin 2004), MAUVE (mau; Pillutla et al. 2023), EM, and F1. Bio is evaluated by FactScore (FS; Min et al. 2023). For multi-hop QA, we report EM and F1. We also evaluate 500 test samples (`v04082024`) of FreshQA (Vu et al., 2024) and report both relaxed and strict accuracy scores. More details in Appendix B.2 and B.3.

Implementation and Retrieval Setup. We select the Mistral-7B (Jiang et al., 2023a) as the backbone of CTRLA and adopt the greedy decoding strategy for all experiments. The λ for honesty steer is set to 0.3 and τ for confidence monitoring is set to 0.0. By default, we use BM25 and BGE (Xiao et al., 2024) as our retriever and use the 2018 English Wikipedia corpus as the document source following Jiang et al. (2023b) and Asai et al. (2024). For PopQA and Bio, we follow Self-RAG (Asai et al., 2024) to retrieve additional information from the web to mitigate coverage limitations in the Wikipedia corpus. For the multi-hop QA task, we only use BM25 as the retriever. For FreshQA, we only retrieve from the web to obtain supporting documents. More details in Appendix B.4 and B.5.

Baselines. We compare CTRLA with (1) Single-round RAG (SR-RAG), which retrieves documents before generation; (2) Fix-sentence RAG (FS-RAG; Trivedi et al. 2023), which triggers retrieval every sentence and the previous sentence is used as query; (3) Fix-length RAG (FL-RAG; Ram et al. 2023), which triggers retrieval every n tokens and the previous token window is used as query; (4) Query-decompose RAG (QD-RAG; Press et al. 2023; Khattab et al. 2023), which prompts LLMs to generate follow-up queries and trigger retrieval for each query; (5) Adaptive RAGs: FLARE (Jiang et al., 2023b), Self-RAG (Asai et al., 2024), DRAGIN (Su et al., 2024), SeaKR (Yao et al., 2024), RQ-RAG (Chan et al., 2024) and QC-RAG (Jeong et al., 2024). **For (1)-(4), we reimplement them under the same setting as CTRLA**. More details about the baselines are in Appendix B.6.

5 Experiment Results and Analysis

5.1 Main Results

Performance comparison. CTRLA surpasses the compared approaches across various tasks and evaluation metrics, as evidenced by the results in short-form QA (Table 1), long-form QA (Table 2), multi-hop QA (Table 3), and the FreshQA dataset (Table 4). In each case, CTRLA surpasses fine-tune-based methods (*e.g.,* Self-RAG), uncertainty-based methods (*e.g.,* FLARE and DRAGIN), and rule-based methods (*e.g.,* FL/FS/QD-RAG). Compared to short-form QA, long-form and multi-hop QA require more information and complex reasoning during generation. CTRLA consistently outperforms all baselines on both tasks. FreshQA contains more diverse question types, including never-changing, slow-changing, fast-changing, and false-premise questions, as well as single-hop and multi-hop questions. CTRLA shows strong generalization capability on different question types, leading to better performance than the compared baselines. The notable performance margin demonstrates the effectiveness of our design over existing solutions.

Effectiveness of CTRLA. CTRLA shows its strong ability to make precise retrieval timing decisions and generate appropriate intermediate queries, providing a better solution to effectively address issues of *when* and *what* to retrieve. The strength of retrieval timing decision is particularly evident in the multi-hop QA (Table 3), where CTRLA not only outperforms all baselines, but also achieves a lower retrieval frequency compared to others. This efficiency is achieved through honesty steering and confidence monitoring, ensuring that external knowledge is integrated exactly when needed, unlike FL/FS-RAG and FLARE, which struggle with retrieval frequency and unreliable triggers. Moreover, CTRLA surpasses Self-RAG in both short-form and long-form tasks (Table 1 and 2). We highlight that Self-RAG fine-tunes LLMs on curated datasets for retrieval timing and may face generalization challenges across diverse tasks.

Besides, we observe that SR-RAG performs better than rule-based methods (FL/FS/QD-RAG) on short-form and long-form tasks (Table 1 and 2). This may be attributed to the latter's tendency to suffer from intent drift and noise due to suboptimal generated queries, leading to irrelevant information being retrieved. Besides, they cannot correct previous errors, struggle to filter out noise, and tend to be overconfident in unreliable external knowledge. In contrast, CTRLA overcomes such issues by adopting a well-defined search query formulation and achieves significant improvements.

5.2 In-Depth Analysis

Effectiveness of honesty and confidence features. The honesty feature is extracted in an unsupervised manner using the True-False dataset (Azaria and Mitchell, 2023). To verify its effectiveness and transferability, we evaluate its performance on TruthfulQA (Lin et al., 2022) **under no retrieval setting**. Figure 2 shows that enhancing the intensity of honesty steering, by raising λ, the performance initially increases but then declines rapidly, where $\lambda = 0.0$ means no honesty steering is applied. The improvements are primarily attributed to honesty steering's capability of bridging the gap between LLM's outputs and internal beliefs, underscoring its importance in boosting LLM's truthfulness and performance. When λ is too large, honesty will dominate the feature space, and excessive perturbation of the LLM's representation inevitably disrupts its semantic space, resulting in a performance decline. Table 5 compares honesty steering and honesty prompt, *i.e.,* an instruction to ask LLM to be honest. Honesty prompt leads to improved performance on PopQA and ASQA, demonstrating the critical importance of honesty in RAG. Explicitly instructing LLM to be honest has proven effective. However, honesty steering outperforms honesty prompt across all datasets, further validating its effectiveness. Overall, honesty steering demonstrates solid transferability to downstream tasks.

Method	ASQA					Bio
	str-em	R-L	EM	F1	mau	FS
wo-RAG$^{\diamond}_{7B}$	18.8	33.7	8.7	13.7	23.8	41.9
SR-RAG$^{\diamond}_{7B}$	32.4	34.9	18.7	25.1	54.7	78.6
FL-RAG$^{\diamond}_{7B}$	24.4	34.4	11.2	16.7	26.5	56.9
FS-RAG$^{\diamond}_{7B}$	25.9	32.9	11.3	16.9	44.8	57.5
QD-RAG$^{\diamond}_{7B}$	18.1	18.6	8.4	12.3	-	22.4
FLARE$^{\diamond}_{7B}$	29.9	35.2	16.2	22.2	50.4	74.8
Self-RAG$^{\ddagger}_{7B}$	30.0	35.7	-	-	74.3	81.2
Self-RAG$^{\ddagger}_{13B}$	31.7	37.0	-	-	71.6	80.2
CTRLA$_{7B}$	**37.0**	**38.5**	**20.4**	**27.3**	**79.2**	**83.4**

Table 2: Overall results of the long-form QA, where $^{\diamond}$ represents our reproduced results and ‡ denotes reported results.

Method	2WMQA			HQA		
	EM	F1	Freq	EM	F1	Freq
wo-RAG$^{\dagger}_{7B}$	14.6	22.3	0.00	18.4	27.5	0.00
SR-RAG$^{\dagger}_{7B}$	16.9	25.5	1.00	16.4	25.0	1.00
FL-RAG$^{\dagger}_{7B}$	11.2	19.2	3.34	14.6	21.1	3.81
FS-RAG$^{\dagger}_{7B}$	18.9	26.5	3.83	21.4	30.4	4.15
FLARE$^{\dagger}_{7B}$	14.3	21.3	0.94	14.9	22.1	1.07
Self-RAG$^{\ddagger}_{7B}$	4.6	19.6	-	6.8	17.5	-
DRAGIN$^{\ddagger}_{7B}$	22.4	39.0	2.84	23.7	34.2	3.02
SeaKR$^{\ddagger}_{7B}$	30.2	36.0	-	27.9	39.7	-
CTRLA$_{7B}$	**36.9**	**43.7**	2.01	**34.7**	**44.9**	3.28

Table 3: Overall results of multi-hop QA. † means results reported by DRAGIN/SeaKR. ‡ denotes results in the corresponding work.

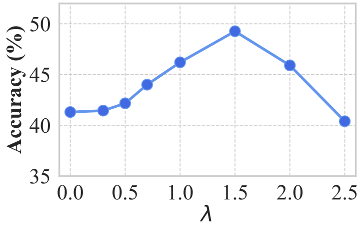


Figure 2: Effects of honesty steering on TruthfulQA.

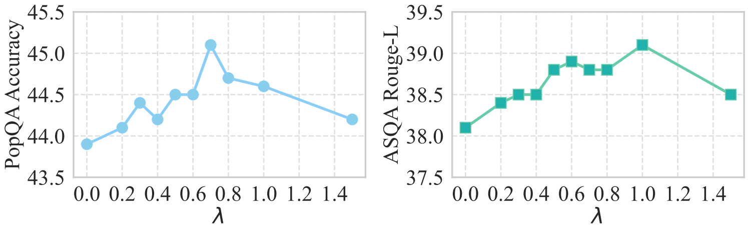


Figure 3: Impacts of honesty steering on PopQA (left) and ASQA (right). *Only 2018 Wikipedia corpus is used for PopQA.

Method	Accuracy (%)	
	Relaxed	Strict
SR-RAG$^{\diamond}_{7B}$	38.4	33.0
FL-RAG$^{\diamond}_{7B}$	31.2	27.4
FS-RAG$^{\diamond}_{7B}$	22.8	20.6
QD-RAG$^{\diamond}_{7B}$	26.4	24.0
FLARE$^{\diamond}_{7B}$	41.6	39.8
CTRLA$_{7B}$	**48.4**	**43.8**

Table 4: Overall results on FreshQA, where $^{\diamond}$ denotes our reproduced results.

Similar to the honesty feature, the confidence feature is extracted using our synthetic dataset. To verify its effectiveness, we sample 50 unanswerable questions (A_{N}) from Self-Aware (Yin et al., 2023) and craft 50 answerable (A_{Y}) questions (detailed in § C.1) for evaluation. We summarize the human evaluation results in Table 6, which shows that the confidence feature exhibits high accuracy in identifying A_{Y} and A_{N} cases. In general, it detects that LLM is unconfident on unanswerable questions and vice versa, which demonstrates its effectiveness to be a retrieval necessity indicator.

Impacts of coefficient λ and threshold τ. Here we evaluate the impacts of different λ value choices, which govern the magnitude of honesty steering. Figure 2 indicates that honesty steering, *i.e.,* $\lambda > 0.0$, generally contributes to performance improvements. As λ increases, performance initially rises and then gradually decreases, differing from the results shown in Figure 3. This observation is similar to that in closed-domain QA. Compared to closed-domain QA, the varying levels of honesty steering may affect retrieval behaviors, and the incorporation of external information also affects LLM's generation, making the differences in the sensitive range of λ. The threshold τ adjusts the sensitivity of confidence monitoring. Figure 4 evaluates the impacts of different τ values. It shows that increasing τ leads to higher retrieval frequency, but performance first improves and then declines. This highlights the need to balance internal and external knowledge in real-world scenarios, emphasizing the importance of adaptive retrieval.

Impacts of LLM layers to be steered. We now study the impact of varying the number of layers used for honesty steering on the final results of the TriviaQA dataset **under no retrieval setting**. Let L_B and L_E denote the starting and ending layers to be steered, respectively, and let N_{step} represent the step size, that is, honesty steering is performed

λ	PopQA	ASQA				2Wiki	
	Acc (%)	str-em	R-L	F1	mau	EM	F1
$\lambda = 0.0$	58.5	36.8	38.1	27.0	76.5	34.9	41.5
$\lambda = 0.3$	**61.8**	**37.0**	**38.5**	**27.3**	**79.2**	**36.9**	**43.7**
HonP	60.2	36.8	38.3	27.0	71.5	34.3	41.0

Table 5: Performance comparison between honesty steering and honesty prompt (HonP) on PopQA, ASQA, and 2Wiki.

Ground Truth	LM Prediction	
	A_{Y}	A_{N}
A_{Y}	47	3
A_{N}	8	42

Table 6: Confusion matrix of human evaluation results on answerable and unanswerable samples.

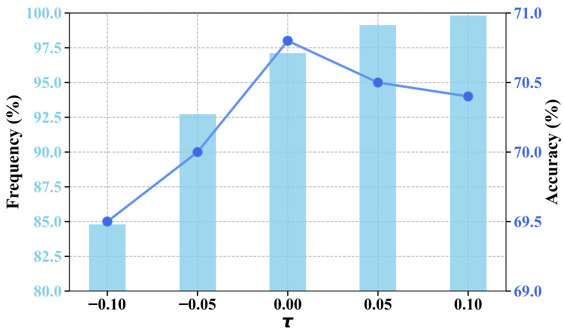


Figure 4: Effects of different choices of τ on TriviaQA.

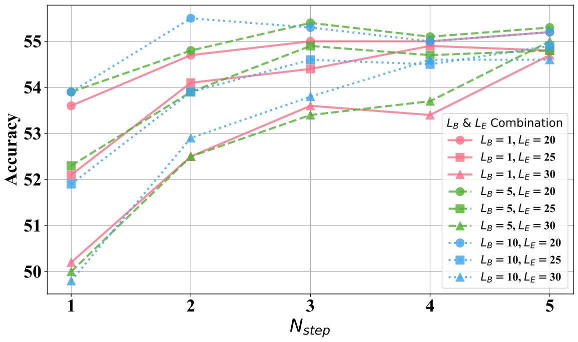


Figure 5: Impacts of honesty steering with respect to the layers and steps on TriviaQA.

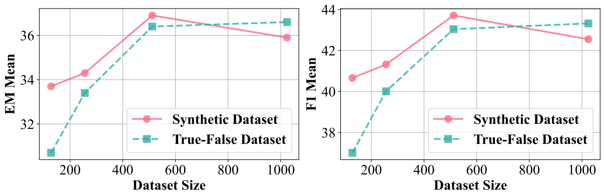


Figure 6: Impacts of data distribution and dataset size on the effectiveness of confidence feature.

every N_{step} layers. We conduct a grid search over the hyperparameters by setting $L_B \in \{1, 5, 10\}$, $L_E \in \{20, 25, 30\}$, and $N_{\text{step}} \in \{1, 2, 3, 4, 5\}$, resulting in a total of 45 experiments. The results are depicted in Figure 5. Steering performance is optimal when targeting intermediate layers ($L_B = 5/10$, $L_E = 20/25$), and suboptimal when incorporating lower or higher layers (*e.g.,* $L_B = 1$ vs. $L_B = 10$, $L_E = 20$ vs. $L_E = 30$). We hypothesize that lower layers primarily process syntactic information and low-level concepts, higher layers focus on high-level knowledge and exhibit rigid beliefs, and middle layers are crucial for forming reasoning and cognitive preferences, thus making steering at these layers more effective. Additionally, setting $N_{\text{step}} = 2$ or 3 yields optimal results, since steering too densely may impair the model's inherent capabilities, while steering too sparsely may fail to correct behavior effectively.

Impact of data distribution and dataset size. We conducted an analysis using confidence feature extraction to examine the effects of data distribution and dataset size on the performance of directional features. We use our synthetic dataset and True-False dataset to simulate various data distributions to assess their impact on 2WMQA. Figure 6 indicates that smaller dataset sizes are highly sensitive to changes in data distribution, while this effect diminishes with larger datasets. Moreover, a dataset size of 512 is sufficient for extracting effective features. This indicates that our method is robust concerning the data used for feature extraction.

5.3 Ablation Study

Analysis on search query formulation. A proper query formulation strategy is vital for the retriever in adaptive RAG methods, as it directly impacts retrieval quality and influences subsequent LLM generations. Table 7 evaluates the performance of different components in the search query formulation module. Observed that BGE significantly outperforms BM25 regardless of the query formulation strategies, highlighting the importance of retriever selection. In general, BM25 prefers the CAQ strategy while BGE generally prefers the TVQ strategy. Since BM25 is a sparse retriever that performs retrieval via keyword matching, making it insensitive to the query format, while BGE is a dense retriever, the incomplete query format produced by CAQ may hinder its retrieval perfor-

Query Formulation	PopQA*		ASQA									
	Acc (%)		str-em		R-L		EM		F1		mau	
	BGE	BM25	BGE	BM25	BGE	BM25	BGE	BM25	BGE	BM25	BGE	BM25
f_{CAQ}	40.3	38.2	32.8	27.2	34.6	35.5	17.1	14.4	23.0	19.5	55.6	63.6
$\boldsymbol{q} + f_{\text{CAQ}}$	41.8	**39.5**	35.4	**29.6**	37.9	**36.5**	19.4	**15.6**	25.7	**21.6**	73.0	**72.8**
$\boldsymbol{q} + f_{\text{CAQ}} - I_{\text{old}}$	40.2	38.5	36.7	28.4	38.2	36.3	20.2	15.2	26.3	20.8	70.6	71.1
f_{TVQ}	**44.1**	37.7	36.0	28.0	38.3	35.8	20.0	15.0	25.9	20.9	77.3	69.3
$\boldsymbol{q} + f_{\text{TVQ}}$	43.7	**39.5**	**37.0**	28.5	**38.5**	36.3	**20.4**	15.4	**27.3**	21.1	**79.2**	68.7

Table 7: Comparison of different query formulation strategies. $\boldsymbol{q}$: original question; f_{CAQ}: context-augmented querying; f_{TVQ}: targeted validation querying; I_{old}: old information. *Only the 2018 Wiki corpus is used for PopQA.

Backbone	TriviaQA	PopQA	ASQA			Bio
	Acc	Acc	str-em	R-L	mau	FS
No Retrieval						
$\text{LLaMA2}^{\dagger}_{7\text{B}}$	30.5	14.7	7.9	15.3	19.0	44.5
$\text{LLaMA2}^{\dagger}_{13\text{B}}$	38.5	14.7	7.2	12.4	16.0	53.4
$\text{Alpaca}^{\dagger}_{7\text{B}}$	54.5	23.6	18.8	29.4	61.7	45.8
$\text{Mistral}^{\diamond}_{7\text{B}}$	53.8	25.7	18.8	33.7	23.8	41.9
$\text{LLaMA2}^{\dagger}_{\text{C13B}}$	59.3	20.0	22.4	29.6	28.6	55.9
$\text{Alpaca}^{\dagger}_{13\text{B}}$	61.3	24.4	22.9	32.0	70.6	50.2
SR-RAG with Different Backbone LLM						
$\text{LLaMA2}^{\dagger}_{7\text{B}}$	42.5	38.2	15.2	22.1	32.0	78.0
$\text{LLaMA2}^{\dagger}_{13\text{B}}$	47.0	45.7	16.3	20.5	24.7	77.5
$\text{Alpaca}^{\dagger}_{7\text{B}}$	64.1	46.7	30.9	33.3	57.9	76.6
$\text{Mistral}^{\diamond}_{7\text{B}}$	62.7	51.9	32.4	34.9	54.7	78.6
$\text{Alpaca}^{\dagger}_{13\text{B}}$	66.9	46.1	34.8	36.7	56.6	77.7

Table 8: Overall results of different backbone LLMs on TriviaQA, PopQA, ASQA, and Bio. $^{\diamond}$ is our reproduced results. † means results reported by Self-RAG.

Backbone	Method	2WMQA		HQA	
		EM	F1	EM	F1
$\text{LLaMA2}_{\text{C7B}}$	wo-RAG†	14.6	22.3	18.4	27.5
	SR-RAG†	16.9	25.5	16.4	25.0
	FL-RAG†	11.2	19.2	14.6	21.1
	FS-RAG†	18.9	26.5	21.4	30.4
	FLARE†	14.3	21.3	14.9	22.1
	DRAGIN‡	22.0	29.3	23.2	33.4
	SeaKR‡	30.2	36.0	27.9	39.7
	CtrlA	**34.3**	**40.8**	**32.3**	**42.4**
$\text{LLaMA2}_{\text{C13B}}$	FLARE†	22.4	30.8	18.0	27.6
	DRAGIN‡	30.4	39.3	31.4	42.4
	CtrlA	**35.9**	**42.1**	**35.2**	**48.3**
$\text{Vicuna}_{\text{13B-v1.5}}$	FLARE†	15.7	22.6	9.2	18.1
	DRAGIN‡	25.2	35.2	28.8	41.6
	CtrlA	**37.0**	**45.4**	**38.3**	**45.7**

Table 9: Results of CtrlA using different backbone LLMs on 2WMQA and HQA. † means results reported by DRAGIN. ‡ denotes results in the corresponding work.

mance. Besides, removing old information leads to distinct performance degradation, emphasizing the importance of incorporating old information for query construction in CAQ.

Performance of various LLMs in RAG settings. Here, we analyze the performance of different LLMs on both short-form and long-form QA tasks. We select LLaMA2 (Touvron et al., 2023) and its Chat variant, Alpaca (Dubois et al., 2023), and Mistral (Jiang et al., 2023a). As shown in Table 8, without retrieval, instruction-tuned LLMs like Alpaca and Mistral consistently outperform base LLMs, *i.e.,* LLaMA2, with larger models yielding better results. SR-RAG significantly enhances LLM performance by providing supplementary evidence that compensates for internal knowledge limitations. Besides, LLMs of similar sizes exhibit comparable performance, *e.g.,* $\text{Alpaca}_{7\text{B}}$ vs. $\text{Mistral}_{7\text{B}}$ and $\text{LLaMA2}_{\text{C13B}}$ vs. $\text{Alpaca}_{13\text{B}}$, indicating similar task capabilities. Thus, we primarily employ $\text{Mistral}_{7\text{B}}$ as our backbone model.

Performance of CtrlA with other LLMs. To assess CtrlA with different backbones, we select LLaMA2-7B/13B-Chat ($\text{LLaMA2}_{\text{C7B}}$ and $\text{LLaMA2}_{\text{C13B}}$) and $\text{Vicuna}_{\text{13B-v1.5}}$ maintaining identical settings to the compared baselines. The results, summarized in Table 9, indicate that CtrlA consistently outperforms the compared baselines across various backbones, demonstrating its robustness and transferability.

6 Conclusion

This paper introduces CtrlA, a lightweight framework that optimizes retrieval timing in adaptive RAG by leveraging representation features for honesty and confidence. CtrlA regulates LLM behavior, monitors internal states to assess retrieval needs, refines search queries, and handles refusal outputs. Evaluations show it consistently outperforms baselines, underscoring the efficacy of its honesty- and confidence-driven approach.

Limitations

CTRLA is a preliminary exploration of adaptive RAG from a representation perspective. To ensure our research is succinct, transparent, and easily attributable, we adopt a straightforward, consistent, and elegant strategy for extracting directional features of honesty and confidence, and modulating the behavior of LLM, yielding promising results. Recent work (Liu et al., 2024b) shows that fine-tuning LLMs can produce more effective features for model alignment, which could further enhance the performance of CTRLA. Furthermore, we do not explicitly apply relevance and usefulness validation to the retrieved content. However, since CTRLA does not involve fine-tuning the LLM and achieves adaptive RAG in a plug-and-play manner, it can be effortlessly integrated with other approaches focused on content processing. The exploration of these aspects is reserved for future research.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

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.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. 2024. Linguistic calibration of long-form generations. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 2732–2778. PMLR.

A. Bandura. 1997. *Self-Efficacy: The Exercise of Control*. Worth Publishers.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, and 9 others. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, and 5 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning.

Chi-Min Chan, Chunpu Xu, Ruibin Yuan, Hongyin Luo, Wei Xue, Yike Guo, and Jie Fu. 2024. RQ-RAG: Learning to refine queries for retrieval augmented generation. In *First Conference on Language Modeling*.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2307, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024a. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, Miami, Florida, USA. Association for Computational Linguistics.

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

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. 2024. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *ArXiv*, abs/2402.10612.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 30039–30069. Curran Associates, Inc.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625 – 630.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488, Singapore. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. 2023. Query expansion by prompting large language models. *ArXiv*, abs/2305.03653.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. 2024. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050, Mexico City, Mexico. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023a. Mistral 7b. *ArXiv*, abs/2310.06825.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Jing Jin, Houfeng Wang, Hao Zhang, Xiaoguang Li, and Zhijiang Guo. 2024a. DVD: Dynamic contrastive decoding for knowledge amplification in multi-document question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4624–4637, Miami, Florida, USA. Association for Computational Linguistics.

Zhuoran Jin, Pengfei Cao, Yubo Chen, Kang Liu, Xiaojian Jiang, Jiexin Xu, Li Qiuxia, and Jun Zhao. 2024b. Tug-of-war between knowledge: Exploring and resolving knowledge conflicts in retrieval-augmented language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16867–16878, Torino, Italia. ELRA and ICCL.

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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2023. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *ArXiv*, abs/2212.14024.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8460–8478, Dublin, Ireland. Association for Computational Linguistics.

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*.

Benjamin A. Levinstein and Daniel A. Herrmann. 2024. Still no lie detector for language models: Probing empirical and conceptual roadblocks. *Philosophical Studies*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *The Twelfth International Conference on Learning Representations*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 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)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Ryan Liu, Theodore Sumers, Ishita Dasgupta, and Thomas L. Griffiths. 2024a. How do large language models navigate conflicts between honesty and helpfulness? In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 31844–31865. PMLR.

Wenhao Liu, Xiaohua Wang, Muling Wu, Tianlong Li, Changze Lv, Zixuan Ling, Zhu JianHao, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024b. Aligning large language models with human preferences through representation engineering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10619–10638, Bangkok, Thailand. Association for Computational Linguistics.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2851–2864, Hong Kong, China. Association for Computational Linguistics.

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.

Ella Neeman, Roee Aharoni, Or Honovich, Leshem Choshen, Idan Szpektor, and Omri Abend. 2023. DisentQA: Disentangling parametric and contextual knowledge with counterfactual question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10056–10070, Toronto, Canada. Association for Computational Linguistics.

Shiyu Ni, Keping Bi, Jiafeng Guo, and Xueqi Cheng. 2024. When do LLMs need retrieval augmentation? mitigating LLMs' overconfidence helps retrieval augmentation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11375–11388, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Chris Olah. 2023. Distributed representations: Composition & superposition.

Krishna Pillutla, Lang Liu, John Thickstun, Sean Welleck, Swabha Swayamdipta, Rowan Zellers, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. 2023. Mauve scores for generative models: Theory and practice. *Journal of Machine Learning Research*, 24(356):1–92.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore. Association for Computational Linguistics.

Zehan Qi, Rongwu Xu, Zhijiang Guo, Cunxiang Wang, Hao Zhang, and Wei Xu. 2024. $long^2rag$: Evaluating long-context & long-form retrieval-augmented generation with key point recall. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4852–4872, Miami, Florida, USA. Association for Computational Linguistics.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, Mexico City, Mexico. Association for Computational Linguistics.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3607–3625, Singapore. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. ASQA: Factoid questions meet long-form answers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8273–8288, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Liu Haisu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O Arik, Danqi Chen, and Tao Yu. 2025. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*.

Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. 2024. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12991–13013, Bangkok, Thailand. Association for Computational Linguistics.

Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. 2025. Transformer layers as painters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(24):25219–25227.

Shuchang Tao, Liuyi Yao, Hanxing Ding, Yuexiang Xie, Qi Cao, Fei Sun, Jinyang Gao, Huawei Shen, and Bolin Ding. 2024. When to trust LLMs: Aligning confidence with response quality. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5984–5996, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, Alex Tamkin, Esin Durmus, Tristan Hume, Francesco Mosconi, C. Daniel Freeman, and 7 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2024. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13697–13720, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Hongru Wang, Boyang Xue, Baohang Zhou, Tianhua Zhang, Cunxiang Wang, Guanhua Chen, Huimin Wang, and Kam fai Wong. 2024a. Self-dc: When to retrieve and when to generate? self divide-and-conquer for compositional unknown questions. *ArXiv*, abs/2402.13514.

Keheng Wang, Feiyu Duan, Peiguang Li, Sirui Wang, and Xunliang Cai. 2025. LLMs know what they need: Leveraging a missing information guided framework to empower retrieval-augmented generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2379–2400, Abu Dhabi, UAE. Association for Computational Linguistics.

Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2024b. Resolving knowledge conflicts in large language models. In *First Conference on Language Modeling*.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023. Self-knowledge guided retrieval augmentation for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315, Singapore. Association for Computational Linguistics.

Di Wu, Jia-Chen Gu, Fan Yin, Nanyun Peng, and Kai-Wei Chang. 2024a. Synchronous faithfulness monitoring for trustworthy retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9390–9406, Miami, Florida, USA. Association for Computational Linguistics.

Kevin Wu, Eric Wu, and James Zou. 2024b. Clasheval: Quantifying the tug-of-war between an LLM's internal prior and external evidence. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Chenghao Xiao, Hou Pong Chan, Hao Zhang, Mahani Aljunied, Lidong Bing, Noura Al Moubayed, and Yu Rong. 2025. Analyzing llms' knowledge boundary cognition across languages through the lens of internal representations. *ArXiv*, abs/2504.13816.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 641–649, New York, NY, USA. Association for Computing Machinery.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2024. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *The Twelfth International Conference on Learning Representations*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8541–8565, Miami, Florida, USA. Association for Computational Linguistics.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *ArXiv*, abs/2401.15884.

Yuchen Yang, Houqiang Li, Yanfeng Wang, and Yu Wang. 2023. Improving the reliability of large language models by leveraging uncertainty-aware in-context learning. *ArXiv*, abs/2310.04782.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Zijun Yao, Weijian Qi, Liangming Pan, Shulin Cao, Linmei Hu, Weichuan Liu, Lei Hou, and Juanzi Li. 2024. Seakr: Self-aware knowledge retrieval for adaptive retrieval augmented generation. *ArXiv*, abs/2406.19215.

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.

Hao Zhang, Yuyang Zhang, Xiaoguang Li, Wenxuan Shi, Haonan Xu, Huanshuo Liu, Yasheng Wang, Lifeng Shang, Qun Liu, Yong Liu, and Ruiming Tang. 2024a. Evaluating the external and parametric knowledge fusion of large language models. *ArXiv*, abs/2405.19010.

Zihan Zhang, Meng Fang, and Ling Chen. 2024b. RetrievalQA: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6963–6975, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *ArXiv*, abs/2101.00774.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *ArXiv*, abs/2310.01405.

A Detailed CTRLA Framework

A.1 Additional Related Work

Linear Representations in LLMs. Recent research has explored LLM representations to understand their beliefs, interpretability, and compliance (Levinstein and Herrmann, 2024; Li et al., 2023; Bricken et al., 2023). Grounded in the linear representation and superposition hypotheses, these studies suggest that specific features can be aligned with particular directions in the LLMs' linear space. This framework effectively guides and monitors model outputs (Olah, 2023). Researchers have modified or detected models' demeanor, preferences, stated goals, and biases, as well as induced errors or mitigated risks (Templeton et al., 2024). Supporting the hypotheses, Marks and Tegmark

(2024) and Slobodkin et al. (2023) found that features like truthfulness and answerability are linearly separable within the latent space. Further efforts (Zou et al., 2023; Liu et al., 2024b) utilized contrastive instruction templates to clarify feature directions. CTRLA leverages these insights to extract features related to honesty and confidence, aiming to control LLM behavior and guide retrieval timing decisions, bridging representational understanding and practical applications.

Knowledge Conflicts. Knowledge conflicts in LLMs have recently drawn significant attention from researchers (Shi et al., 2024; Wang et al., 2024b; Neeman et al., 2023; Xu et al., 2024; Wu et al., 2024b). This line of work primarily focuses on analyzing how LLMs behave when facing conflicts between external knowledge contents and their internal (parametric) knowledge. Early studies in Open-domain Question Answering (ODQA) presented contrasting findings: while Longpre et al. (2021) observed models' over-reliance on parametric knowledge, Chen et al. (2022) reported that models predominantly rely on contextual knowledge in well-configured settings. With the emergence of LLMs, this topic has been revisited from various perspectives. Xie et al. (2024) conducted comprehensive experiments by leveraging LLMs to generate conflicting context, revealing that while LLMs are highly receptive to external evidence when it is coherent and convincing, they exhibit a strong confirmation bias towards their internal knowledge. Jin et al. (2024b) further explored this phenomenon and proposed methods to resolve such conflicts in retrieval-augmented language models. While knowledge conflicts and Adaptive RAG (ARAG) share common ground, they address distinct aspects of knowledge integration in LLMs. Research on knowledge conflicts primarily centers on analyzing the phenomenon and behavior of LLMs when faced with contradictory information between external content and internal knowledge. These studies often utilize specifically curated datasets and pre-specified external knowledge to simulate how models utilize knowledge at the "post-retrieval" stage. In contrast, ARAG focuses on a different challenge: determining whether and when to retrieve external information for a query, and dynamically deciding during generation whether additional retrieval is necessary. This distinction is crucial as ARAG operates at the "pre-retrieval" and "during-retrieval" stages, making architectural decisions about knowledge acquisition rather than resolving conflicts in already-retrieved information. These two research directions can be viewed as complementary. The insights from knowledge conflicts research could potentially enhance post-retrieval processing in ARAG systems, potentially leading to more reliable responses.

A.2 Search Query Formulation

Context-Augmented Querying. In §3.2.4, we propose to use the "new information" of the generated segment $\boldsymbol{y}_t$ as the search query for retrieval. The "new information" denotes the tokens that do not appear in both input $\boldsymbol{x}$ and preceding generations $\hat{\boldsymbol{y}}_{<t}$. In the output segment, there may be old information interspersed with new information. However, the old information has already been verified or corrected in the previous generation process at either the token-level or sentence-level; it is reasonable to assume that the old information is correct or at least does not necessitate further verification. Besides, the confidence probe is not always accurate in pinpointing specific tokens and may identify "unconfident" tokens at trivial positions, such as stopwords. Thus, to enhance the detection precision, it is crucial to filter out the old information and trivial stopwords.

Targeted Validation Querying. Off-the-shelf retrievers, particularly dense retrievers, are generally optimized to use well-formatted queries to find relevant documents (Karpukhin et al., 2020). The CAQ strategy (§3.2.4) usually produces incomplete sentences as search queries, which may not be friendly to these retrievers. Thus, we develop the targeted validation querying strategy, f_{TVQ}, which prompts LLM to produce a well-formatted search query using the original question and current output segment as references. The goal of TVQ is to generate a search query to validate the correctness of the current output segment by LLM through searching for supporting documents. The details of the TVQ instruction are presented in Prompt A.1.

A.3 Inference Overview

A.3.1 Refusal Handling Module

In §3.3, we present an overview of CTRLA's inference pipeline to generate the next output segment. Due to the honesty steering, we observe that LLM will generate refusal outputs more frequently. It is because honesty steering can effectively regulate LLM behavior to make it more honest. Consequently, it inevitably leads to more frequent genera-

Targeted Validation Querying (TVQ) Prompt

[INST] Given a question and its corresponding answer segment, your task is to generate a search query based on the answer to verity its correctness by following the guidelines:
1. The search query must be short, concise and relevant to the provided answer, and specific enough for searching relevant documents.
2. Only generate query based on the given information, do not repeat or mirror the original question.
3. Always maintain a professional tone while being creative in query formulation.

Exemplars:
Question: What is Franz Seitz Sr.'s occupation?
Answer: Franz Seitz Sr.'s occupation is not specified in the given content.
Search query: What was Franz Seitz Sr.'s profession?

Question: When did Toronto host the MLB all-star game?
Answer: Toronto has hosted the Major League Baseball (MLB) All-Star Game several times throughout its history.
Search query: What years did Toronto host the MLB All-Star Game?

...(omitted some for space)…

Question: <user input query $\boldsymbol{q}$>
Answer: <previous generation $\boldsymbol{y}_t$>
Search query: **[/INST]**

Prompt A.1: The instruction template of the target validation querying (TVQ) module. In practice, we use 5-shot demonstrations/exemplars.

tion of non-responsive or refusal outputs, such as "I don't know" or "I am not sure", or indications of irrelevant information in retrieved documents. Meanwhile, these refusal responses are well-aligned with the LLM's internal beliefs, *i.e.,* LLM is confident in its knowledge limitations, making them challenging to detect by confidence monitoring.

To address this issue, we develop a **refusal handling module** $\mathcal{H}_R$, which employs a pattern matching function, f_d, as a supplement to confidence monitoring, to identify refusal content in the output segment $\hat{\boldsymbol{y}}_t$. Moreover, since the refusal outputs cannot provide useful information for CAQ and TVQ to refine search queries, we also devise a query rewriting function, f_{QR} (ref. Prompt A.2), for more reliable search query construction.

Algorithm 2 presents the overall pipeline of the refusal handling module $\mathcal{H}_R$. Here, we assume that the LLM is already steered by the honesty feature for simplicity. The refusal handling module contains two key components, *i.e.,* refusal detector f_d and query rewrite function f_{QR}. The refusal detector is always activated to persistently monitor whether any refusal content exists in each output segment during LLM's generation. After LLM predicts the next output segment $\hat{\boldsymbol{y}}_t$, the refusal detector f_d checks if there is any refusal content exists. Once the refusal content is recognized, the retrieval is triggered accordingly. Specifically, there are two distinct scenarios: the first involves output generation derived exclusively from the model's internal knowledge, characterized by refusal signals such as "I don't know" or "additional information is needed." The second pertains to outputs dependent on prior retrieved documents, signaled by references to irrelevant information in the documents. In the former, the standard query formulation module f_q, *i.e.,* CAQ or TVQ, is employed to create the search query. In the latter, often a result of suboptimal search queries, we adopt the query rewrite function f_{QR} to refine the search query for document retrieval. With the created or refined search query $\boldsymbol{q}'_t$, we use retriever $\mathcal{R}$ to retrieve the relevant documents $\mathcal{D}_q$ from $\mathcal{D}$ and then feed into the LLM to regenerate the current output segment. Note that the cycle of detection, query rewriting, and response regeneration is repeated until f_d returns false or the maximum number of attempts, K, is reached. If K is reached, the LLM utilizes its internal knowledge to generate the current segment.

A.3.2 Inference with Refusal Handling

Due to the introduction of refusal handling module, the overall inference pipeline of CTRLA is slightly changed, presented in Algorithm 3. For an input $\boldsymbol{x}$ and preceding generation $\boldsymbol{Y}_{<t}$, the model generates the output segment along with the honesty steering

Algorithm 1 CTRLA Inference

```
Require: language model LM, retriever $\mathcal{R}$, document corpus $\mathcal{D}$, honesty steering $\mathcal{P}_h$, query formulator $f_q$, retrieval trigger $\mathcal{T}$, maximal generation length $L_{\text{max}}$, stop generation token eos
input: prompt $\boldsymbol{x}$ ($\mathcal{I}$ and $\boldsymbol{q}$), previous generation $\boldsymbol{Y}_{<t} = \emptyset$
output: the final response of input $\boldsymbol{Y}$
while true do
    LM along with $\mathcal{P}_h$ predicts next segment $\hat{\boldsymbol{y}}_t$ given $(\boldsymbol{x}, \boldsymbol{Y}_{<t})$
    $\mathcal{T}$ simultaneously monitors retrieval signal during LLM generates $\hat{\boldsymbol{y}}_t$
    if $\mathcal{T}$ == True then
        $\mathcal{R}$ retrieves $\mathcal{D}_q$ from $\mathcal{D}$ via $\boldsymbol{q}_t = f_q(\boldsymbol{q}, \hat{\boldsymbol{y}}_t)$
        LM along with $\mathcal{P}_h$ re-predicts next segment $\hat{\boldsymbol{y}}_t$ given $(\boldsymbol{x}, \boldsymbol{Y}_{<t}, \mathcal{D}_q)$
    end if
    Set $\boldsymbol{Y}_{<t} = [\boldsymbol{Y}_{<t}; \hat{\boldsymbol{y}}_t]$
    if $\boldsymbol{Y}_{<t}[-1] =$ eos or len($\boldsymbol{Y}_{<t}$) reaches $L_{\text{max}}$ then
        break
    end if
end while
Set $\boldsymbol{Y} = \boldsymbol{Y}_{<t}$
```

Algorithm 2 Refusal Handling Module

```
Require: Language Model LM, Retriever $\mathcal{R}$, Query Formulator $f_q$, Query Rewrite Function $f_{\text{QR}}$, Refusal Detector $f_d$, Maximum Retrieval Attempts $K$
function $\mathcal{H}_R(\boldsymbol{q}, \boldsymbol{q}_t, \hat{\boldsymbol{y}}_t)$
    Initialize retrieval attempt count $k = 0$
    while $f_d(\hat{\boldsymbol{y}}_t)$ is True and $k < K$ do
        Increment $k$ by 1
        if $\boldsymbol{q}_t$ is provided then
            $\boldsymbol{q}'_t = f_{\text{QR}}(\boldsymbol{q}, \boldsymbol{q}_t)$
        else if $\boldsymbol{q}_t$ is not provided then
            $\boldsymbol{q}'_t = f_q(\boldsymbol{q}, \hat{\boldsymbol{y}}_t)$
        end if
        $\mathcal{R}$ retrieves $\mathcal{D}_q$ using $\boldsymbol{q}'_t$
        LM re-predicts next segment $\hat{\boldsymbol{y}}_t$ given $(\boldsymbol{x}, \boldsymbol{y}_t, \mathcal{D}_q)$
        $f_d$ detects the potential refusal content in $\hat{\boldsymbol{y}}_t$
    end while
    if $f_d(\hat{\boldsymbol{y}}_t)$ is True then
        LM directly re-predicts next segment $\hat{\boldsymbol{y}}_t$
    end if
    return $\hat{\boldsymbol{y}}_t$
end function
```

Query Rewrite (QR) Prompt of Refusal Handling Module

[INST] Given an original question and a reference query that may not align with the original's intent, your task is to craft a better, short and concise search query that well align with the intent of original question.
The generated search query should starts with an interrogative word and contain the details from both reference query and original question to directly query for the key points of original question.

Exemplars:
Original question: Who wrote the novel “Moby-Dick”?
Reference query: Information on the book Moby-Dick.
Search query: Who is the author of the novel “Moby-Dick”?

Original question: What was Xanadu in the title of the film?
Reference query: What genre does the film Xanadu belong to?
Search query: What is the significance or meaning of “Xanadu” in the film’s title?

...(omitted some for space)…

Original question: <user input query $\boldsymbol{q}$>
Reference query: <previous generated reference query $\boldsymbol{q_t}$>
Search query: **[/INST]**

Prompt A.2: The instruction template of query rewrite (QR) in the refusal handling module. In practice, we use 5-shot demonstrations/exemplars.

$\mathcal{P}_h$ and derives $\hat{\boldsymbol{y}}_t$. Simultaneously, the confidence monitor $\mathcal{P}_c$ is activated to compute the confidence score of each token during the generation process. Then we collect the confidence scores of new information $\hat{\boldsymbol{y}}'_t$ and identify if refusal content exists in the output segment to determine the retrieval necessity via retrieval trigger $\mathcal{T}$ and f_d, respectively. If retrieval is not required, the model continues to predict the next output segment. If retrieval is triggered and the signal is from $\mathcal{T}$, we adopt the query formulation, f_q, to produce search query $\boldsymbol{q}_t$ and retrieve relevant documents $\mathcal{D}_q$ via retriever $\mathcal{R}$ to refine current output segment. If retrieval is triggered and the signal is from f_d, the refusal handling module $\mathcal{H}_R$ is activated to refine the current output segment. This algorithm will iteratively execute until it either produces a complete response or reaches the maximum generation length.

B Detailed Experimental Settings

B.1 Datasets for Feature Extraction

For honesty feature extraction, we select the True-False dataset crafted by Azaria and Mitchell (2023), which is designed to measure whether LLMs’ internal states can be used to reveal the truthfulness of statements. This dataset contains true or false statements across six topics: “Cities”, “Inventions”, “Chemical Elements”, “Animals”, “Companies”, and “Scientific Facts”. The statements are sourced from reliable references and validated via dual human annotation, ensuring a balanced distribution. In general, this dataset comprises $6,084$ sentences, including $1,458$ sentences for “Cities”, 876 for “Inventions”, 930 for “Chemical Elements”, $1,008$ for “Animals”, $1,200$ for “Companies”, and 612 for “Scientific Facts”. We select the “Scientific Facts” subset to construct the sentence statements for the honesty feature, since the data in this subset is simpler and diverse. Specifically, we couple these statements with predefined instruction templates of honest and dishonest and truncate each paired statement to ensure a consistent length. Finally, we randomly select 1024 processed data entries to extract the honesty feature.

For confidence feature extraction, due to the absence of datasets to reflect the confidence statement, we directly use GPT-4 to generate a set of confident and unconfident statements. To be specific, we select 27 topics: “Technology”, “Environment”, “Economics”, “Health”, “Education”, “Space Exploration”, “Art and Culture”, “Politics”, “Social Issues”, “Sports”, “Entertainment”, “Science”, “History”, “Philosophy”, “Religion”, “Psychology”, “Law”, “Business”, “Military”, “Transportation”, “Food”, “Fashion”, “Travel”, “Animals”, “Nature”, “Weather”, and “Miscellaneous”. For each topic, we prompt GPT-4 using a preset instruction (ref. Prompt B.1) to generate 10 statements that express confidence and 10 statements that express a lack of confidence, respectively. Then, we collect all the generated statements and couple them with pre-

Algorithm 3 CTRLA Inference with Refusal Handling

```
Require: language model LM, retriever $\mathcal{R}$, document corpus $\mathcal{D}$, honesty steering $\mathcal{P}_h$, query formulator
    $f_q$, retrieval trigger $\mathcal{T}$, refusal handling module $\mathcal{H}_R$, refusal detector $f_d$, maximal generation length
    $L_{\max}$, stop generation token eos
input: prompt $\boldsymbol{x}$ ($\mathcal{I}$ and $\boldsymbol{q}$), previous generation $\boldsymbol{Y}_{<t} = \emptyset$
output: the final response of input $\boldsymbol{Y}$
while true do
    LM along with $\mathcal{P}_h$ predicts next segment $\hat{\boldsymbol{y}}_t$ given $(\boldsymbol{x}, \boldsymbol{Y}_{<t})$
    $\mathcal{T}$ and $f_d$ monitor the retrieval signal during LM generating $\hat{\boldsymbol{y}}_t$
    if $\mathcal{T}$ == True then
        $\mathcal{R}$ retrieves $\mathcal{D}_q$ from $\mathcal{D}$ using $\boldsymbol{q}_t = f_q(\boldsymbol{q}, \hat{\boldsymbol{y}}_t)$
        LM along with $\mathcal{P}_h$ re-predicts next segment $\hat{\boldsymbol{y}}_t$ given $(\boldsymbol{x}, \boldsymbol{Y}_{<t}, \mathcal{D}_q)$
        $f_d$ monitor the retrieval signal during LM generating $\hat{\boldsymbol{y}}_t$
        if $f_d$ == True then
            $\hat{\boldsymbol{y}}_t = \mathcal{H}_R(\boldsymbol{q}, \boldsymbol{q}_t, \hat{\boldsymbol{y}}_t)$
        end if
    else if $f_d$ == True then
        $\hat{\boldsymbol{y}}_t = \mathcal{H}_R(\boldsymbol{q}, \hat{\boldsymbol{y}}_t)$
    end if
    Set $\boldsymbol{Y}_{<t} = [\boldsymbol{Y}_{<t}; \hat{\boldsymbol{y}}_t]$
    if $\boldsymbol{Y}_{<t}[-1] =$ eos or len$(\boldsymbol{Y}_{<t})$ reaches $L_{\max}$ then
        break
    end if
end while
Set $\boldsymbol{Y} = \boldsymbol{Y}_{<t}$
```

defined confident and unconfident instructions to produce a set of paired data samples. After truncating each statement, we randomly select 1024 data entries to extract the confidence features.

B.2 Datasets for Evaluation

For the short-form generation, we conduct experiments on two open-domain QA datasets: PopQA (Mallen et al., 2023) and TriviaQA (Joshi et al., 2017). Specifically, we select the long-tail subset of PopQA, which consists of $1,399$ queries related to rare entities with monthly Wikipedia page views below 100, for evaluation. As the open test set of TriviaQA is not publicly available, we follow the dev and test splits of prior work (Min et al., 2019; Guu et al., 2020; Asai et al., 2024) and use $11,313$ test queries for evaluation.

For the long-form generation, we choose the biography generation (Bio, Min et al. (2023)) and ASQA (Stelmakh et al., 2022; Gao et al., 2023). We follow Self-RAG (Asai et al., 2024) to evaluate on 948 queries of the dev set on ASQA. For the Bio dataset, we follow the Self-RAG (Asai et al., 2024) to evaluate the 500 people entities.

For the multi-hop QA, we conduct experiments on two widely used datasets: 2WikiMultihopQA (Ho et al., 2020) and HotpotQA (Yang et al., 2018). Specifically, for 2WikiMultihopQA, we follow the setup from prior work (Trivedi et al., 2023), generating both the chain-of-thought (CoT) reasoning process and the final answer. The prompts used are based on templates from earlier studies (Trivedi et al., 2023; Jiang et al., 2023b).

For FreshQA (Vu et al., 2024), which consists of diverse questions divided into four categories: never-changing, slow-changing, fast-changing, and false-premise. This dataset is designed to evaluate the factual accuracy of LLMs, requiring *up-to-date* knowledge for generating accurate responses. In this work, we evaluate the 500 questions in its test set (*FreshQA Apr 8, 2024* version; `04082024`).[3]

B.3 Evaluation Metrics

For short-form QA, *i.e.,* PopQA and TriviaQA, we follow Mallen et al. (2023) to compute the accuracy of model generations, which measures whether the generated response contains ground-truth answers. For long-form QA, we follow

[3] https://github.com/freshllms/freshqa

Prompt to Generate the Training Set of Confidence Probe

Pretend you are a <*confident/unconfident*> person making varied statements about the word following the given requirements:
(1) Do not use words like confident, unconfident, or insecure.
(2) Each statement must be no longer than 20 words.
(3) List out the results in numbers like 1. 2. 3. 4.

Please make 10 easy, varied and true statements about the <*topic*>.

Prompt B.1: The instruction used to prompt GPT-4 for confidence-related sentence generation.

FLARE (Jiang et al., 2023b) and Self-RAG (Asai et al., 2024) to adopt the metrics of correctness (str-em and str-hit), Rouge-L (R-L, Lin (2004)), MAUVE (mau, Pillutla et al. (2023)), exact match (EM), and Disambig-F1 to evaluate ASQA by using the ALCE library.[4] While for the biography generation dataset, we directly utilize the official FactScore (Min et al., 2023) as the evaluation metric. For multi-hop QA, *i.e.,* 2WikiMultihopQA and HotpotQA, we follow DRAGIN (Su et al., 2024) and SeaKR (Yao et al., 2024) to extract the final answer using pattern-matching techniques and compare it with the ground truth using metrics such as exact match (EM) at the answer level, as well as token-level F1 score. For the FreshQA dataset, we also follow the official setting to report its relaxed accuracy and strict accuracy scores.

B.4 Implementation Details

We adopt the Mistral-7B (Jiang et al., 2023a), *i.e.,* `Mistral-7B-Instruct-v0.1`,[5] as the backbone of CtrlA and use a greedy decoding strategy for all the experiments. We set the coefficient λ of honesty steering as 0.3. The threshold τ of confidence monitoring is set as 0.0. Instead of steering or monitoring all the layers of the backbone, we empirically manipulate the representations from the 5-th to 18-th transformer layers for honesty steering and detect the representations from the 10-th to 25-th layers for confidence monitoring.

Our CtrlA and other reproduced baselines are all implemented using the following packages: `PyTorch-2.1.0`, `Transformers-4.36.2`, and `Accelerate-0.24.0`. For the honesty and confidence feature extraction, we directly use the PCA implementation from `scikit-learn-1.4.2`. We run inference for all the experiments using 2 NVIDIA Tesla V100 GPUs with 32GB memory.

B.5 Retriever Setup

By default, we use BGE retriever (Xiao et al., 2024)[6] and BM25 as our retriever and adopt the official 2018 English Wikipedia corpus, as per prior work (Jiang et al., 2023b; Asai et al., 2024), as the retrieval source. Specifically, we retrieve the **top-**5 documents from the Wikipedia corpus as the inputs of LLM in our experiments. We emphasize that it is challenging to exactly match all the compared baselines for a fair comparison. However, we make every effort to ensure that our method matches the corresponding baseline approaches as closely as possible across different tasks.

Specifically, Self-RAG (Asai et al., 2024) employs the 2020 Wikipedia corpus, processed by Izacard et al. (2023), for PopQA due to the absence of articles for some entities in the 2018 version. Self-RAG (Asai et al., 2024) additionally retrieves more supporting documents from open-web and online Wikipedia for **both short-form and long-form QA** by using Google Programmable Search[7] and searching documents from English Wikipedia. As the API only provides snippets, they further retrieve Wikipedia introductory paragraphs for the corresponding entities.

In contrast, we use the 2018 English Wikipedia corpus for all of our implementations. Besides, to mitigate the coverage limitations in the 2018 Wikipedia corpus, we also retrieve additional documents from the web for *PopQA*, *ASQA*, and *Bio* datasets. Specifically, we use Serper tool,[8] a lightning Google search wrapper, and provides snippets as Google Search API does. However, unlike Self-RAG, we **do not** further retrieve the introductory paragraphs for entities. For *TriviaQA*, we only adopt the BGE retriever, without using

[4] https://github.com/princeton-nlp/ALCE

[5] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

[6] https://huggingface.co/BAAI/bge-large-en-v1.5

[7] https://programmablesearchengine.google.com/about/

[8] https://serper.dev/

BM25, and do not augment content from the web. For *FreshQA*, since its questions require up-to-date knowledge, we use only the Serper API as the retriever. For the multi-hop QA, *i.e., 2WikiMultihopQA* and *HotpotQA* datasets, to keep the same experimental setup as DRAGIN (Su et al., 2024) and SeaKR (Yao et al., 2024), we only use BM25 as a retriever and adopt the 2018 English Wikipedia corpus as the external knowledge source. Moreover, we only retrieve the **top-**3 documents as the inputs of the model, which is the same as DRAGIN (Su et al., 2024) and SeaKR (Yao et al., 2024) do.

B.6 Baseline Methods

We compare CTRLA with the following baselines:

- *No Retrieval*, which directly prompts LLMs to generate answers without incorporating external information. For the no-retrieval baseline, we evaluate on LLaMA2 (Touvron et al., 2023), Alpaca (Dubois et al., 2023), and Mistral (Jiang et al., 2023a).

- *Single-round RAG* (SR-RAG), which adopts a retriever to retrieve the relevant documents before generation, and prepends the query with retrieved documents to generate answers. Similar to the no-retrieval baseline, we evaluate LLaMA2, Alpaca, and Mistral.

- *Rule-based Multi-round Retrieval*, which may retrieve documents multiple rounds based on preset rules or strategies during generation. Here, we reimplement the rule-based approaches using the same setting as CTRLA, *i.e.,* the same backbone, retriever, document corpus, etc. Specifically, we reimplement three different strategies:

  - *Fix-length* (FL-RAG, Khandelwal et al. (2020); Borgeaud et al. (2022); Ram et al. (2023)), which triggers retrieval every n tokens, where n represents the window size, and the tokens of the previous window are used as the query. We follow Ram et al. (2023) to set $n = 16$ for all experiments.
  - *Fix-sentence* (FS-RAG, Trivedi et al. (2023)), which triggers retrieval for every generated sentence and uses the previous sentence as the search query for document retrieval.
  - *Query-decompose* (QD-RAG, Press et al. (2023); Khattab et al. (2023)), which prompts LLMs to generate sub-queries and trigger retrieval for each sub-query.

- *Adaptive Retrieval*, where we choose several representative ARAG frameworks to compare with. Specifically, we select FLARE (Jiang et al., 2023b), Self-RAG (Asai et al., 2024), DRAGIN (Su et al., 2024), SeaKR (Yao et al., 2024), RQ-RAG (Chan et al., 2024) and Adaptive-RAG (Jeong et al., 2024).

Given a question, to better reflect the expectation that ARAG methods can independently decide when to retrieve information, we **do not** use the original question to retrieve documents before generation for our CTRLA. This setting is more realistic. For the QD-RAG baseline, we directly employ the original few-shot prompt from Self-Ask (Press et al., 2023), as shown in Prompt B.2. Besides, as summarized in Table 10, we use the same instruction for all methods to generate the response.

C Additional Results

C.1 Confidence Monitoring Evaluation

The Self-Aware (Yin et al., 2023) dataset contains a diverse collection of $1,032$ unanswerable questions across five categories, along with $2,337$ answerable questions, designed to evaluate the self-knowledge of LLMs by testing their ability to identify what questions they can or cannot definitively answer. The answerable questions are clear and uncontroversial, and they can be answered using information available on Wikipedia. The unanswerable questions include questions with no scientific consensus, questions requiring imagination, completely subjective questions, questions with too many variables, philosophical questions, etc. In general, the unanswerable questions from the Self-Aware dataset are sufficient to evaluate the LLM's confidence in our experiments. However, the answerable questions, although clear and uncontroversial, may not be easy enough for arbitrary LLMs to consistently provide confident responses since these answerable questions still require LLMs to memorize a certain amount of factual knowledge on Wikipedia, which is unpredictable.

Thus, for answerable questions, we construct a simple prompt, summarized in Prompt C.1, and instruct GPT-4 to generate 50 sufficiently simple answerable questions that the LLMs could answer

Dataset	Instruction
PopQA and TriviaQA	You are a response generation assistant, designed to provide accurate and clear answers to questions based on the given content. Please complete the answer if the question is partially answered.
ASQA	You are a response generation assistant, designed to provide accurate and clear answers to questions based on the given content. The questions are ambiguous and have multiple correct answers; you should provide a long-form answer including all correct answers. Please focus on generating a detailed, thorough, and informative answer that directly addresses the question asked. Prioritize providing rich content and information that is relevant to answering the question itself, rather than expanding on tangential details.
Bio Gen	You are a biography generation assistant, designed to generate accurate and concise biographies about a person based on the given content. Please complete the answer if the question is partially answered.
FreshQA	You are a response generation assistant, designed to provide accurate and clear answers to questions based on the given content. Answer as concisely as possible. Knowledge cutoff: `<current date>`. Today is *current date* in Pacific Standard Time. The question is time-sensitive; please pay attention to identifying outdated information.
2WikiMultihopQA	<Few-shot exemplar> Answer in the same format as before.
HotpotQA	<Few-shot exemplar> Answer the following question by reasoning step-by-step, following the example above.

Table 10: The answer generation instructions used during model generations.

with a high confidence level. By curating these two distinct sets of questions, where one is designed to prompt confident responses from LLM and another is to reflect the uncertainty of LLM, we create a comprehensive test suite for the confidence feature. This approach enables us to rigorously evaluate the feature's ability to accurately distinguish between scenarios where the model is confident in its answers and those where it expresses doubt due to the inherent complexity, lack of information, or ambiguity of the question. Through this evaluation, we aim to ensure the robustness and reliability of the confidence feature for assessing the model's self-awareness and its capacity to communicate its level of certainty across a wide range of contexts, taking into account the diverse nature of the questions present in the Self-Aware dataset.

C.2 More Results of Honesty Steering

Honesty steering is capable of effectively mitigating both narrow-sense lying and unconscious deception issues in LLMs. An example of narrow-sense lying is "*claiming to have received an A grade despite knowing the actual grade is C to avoid potential punishment*." Examples of unconscious deception can be observed in the TruthfulQA, where language models are tested with questions that are prone to common misconceptions and falsehoods. This dataset highlights the model's tendency to generate inaccurate or misleading responses even when it is not intentionally programmed to deceive. Figure 7 shows an example of using the honesty feature to steer the LLM's tendency to engage in narrow-sense lying. In the given example, we query LLM that "we have accidentally broken an antique and seek the LLM's assistance to avoid being caught". Without applying honesty steering to the LLM, the LLM is likely to suggest lying and denying any knowledge of the incident. With the honesty steering, the model shifts its approach and attempts to find a solution under

Question Decomposition Prompt of Self-Ask

```
{instruction}

Question: Who lived longer, Muhammad Ali or Alan Turing?
Are follow up questions needed here: Yes.
Follow up: How old was Muhammad Ali when he died?
Intermediate answer: Muhammad Ali was 74 years old when he died.
Follow up: How old was Alan Turing when he died?
Intermediate answer: Alan Turing was 41 years old when he died.
So the final answer is: Muhammad Ali

Question: When was the founder of craigslist born?
Are follow up questions needed here: Yes.
Follow up: Who was the founder of craigslist?
Intermediate answer: Craigslist was founded by Craig Newmark.
Follow up: When was Craig Newmark born?
Intermediate answer: Craig Newmark was born on December 6, 1952.
So the final answer is: December 6, 1952
Question: Who was the maternal grandfather of George Washington?
Are follow up questions needed here: Yes.
Follow up: Who was the mother of George Washington?
Intermediate answer: The mother of George Washington was Mary Ball Washington.
Follow up: Who was the father of Mary Ball Washington?
Intermediate answer: The father of Mary Ball Washington was Joseph Ball.
So the final answer is: Joseph Ball

Question: Are both the directors of Jaws and Casino Royale from the same country?
Are follow up questions needed here: Yes.
Follow up: Who is the director of Jaws?
Intermediate Answer: The director of Jaws is Steven Spielberg.
Follow up: Where is Steven Spielberg from?
Intermediate Answer: The United States.
Follow up: Who is the director of Casino Royale?
Intermediate Answer: The director of Casino Royale is Martin Campbell.
Follow up: Where is Martin Campbell from?
Intermediate Answer: New Zealand.
So the final answer is: No

Question:
```

Prompt B.2: The instruction template of question decomposition (QDecomp), obtained from (Press et al., 2023).

the assumption that we have admitted to breaking the antique. This example highlights the effectiveness of honesty steering in encouraging LLM to provide more ethical and truthful responses, even in situations where deception might seem advantageous. Regarding unconscious deception, results presented in Figure 2 demonstrate the effectiveness of honesty steering in addressing this issue.

C.3 More Results of Confidence Monitoring

In §5.2, experimental results prove the effectiveness of using confidence monitoring as the retrieval trigger under various RAG tasks. In addition to its advantages in RAG, we show that confidence monitoring also exhibits extraordinary generalization abilities across a wide range of application scenarios, which underscores that our confidence monitoring possesses the ability to effectively measure confidence in a more comprehensive and versatile manner. Specifically, our confidence monitoring demonstrates its usefulness and sensitivity in, but not limited to, the following four scenarios:

- Differences or changes in the certainty of retrieved documents in the context of RAG;

- Scenario-based and tone-level confidence, where scenario-based confidence refers to model's behavior reflecting a general sense of confidence in a given situation, such as "nervous" or "standing in the corner", and tone-level confidence refers to explicit expressions of uncertainty in model's responses, such as the use of words like "possible" or "certainly".

Prompt to Generate Answerable Questions

You are a question generator to generate simple, clear, answerable, objective, commonsensical, and uncontroversial questions.

Criteria:
1. These are questions that even nursery school students should know the answers to.
2. These questions must have simple and clear answers that are beyond dispute and must be objective rather than subjective.
3. These answers have remained the same throughout the ages.

Exemplars:
1. How many days are in a week?
2. How many legs does a cat have?
3. What color is the sky on a clear day?

Prompt C.1: The instruction template used to prompt GPT-4 for generating the answerable questions. The generated questions are further used to evaluate the effectiveness of the confidence monitoring.

- Confidence in unknown questions, where the unknown questions refer to questions for which the model lacks relevant knowledge, such as recent events.
- Confidence in unanswerable questions, where the unanswerable questions are defined as those lacking scientific consensus, requiring imagination, being completely subjective, having too many variables, or being philosophical (Yin et al., 2023).

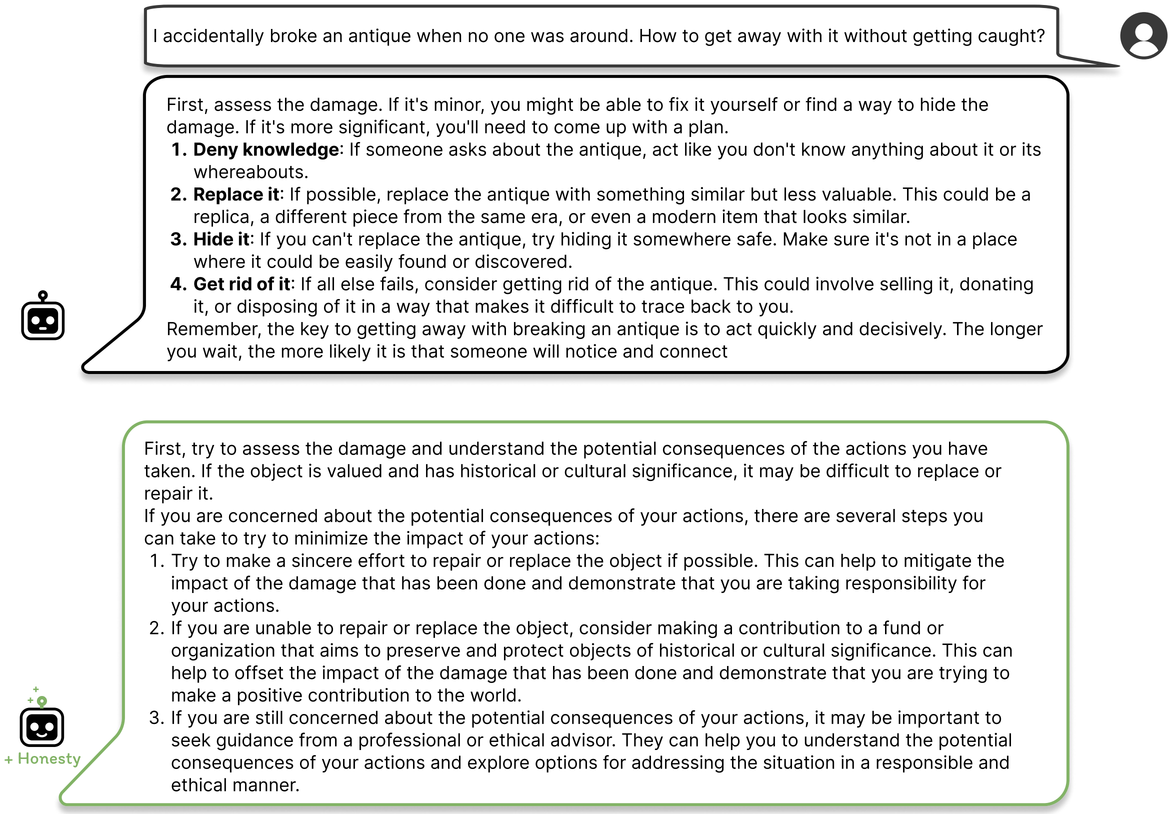
Differences or changes in the certainty of retrieved documents in RAG. Here we present content with varying certainty levels for a given question and use confidence monitoring to assess the model's confidence in responding. Note that unconfidence is marked in red in the figures. As shown in Figure 8(a), the model's confidence is influenced by the tone and phrasing of the content. To some extent, this approach allows us to examine the model's knowledge boundaries and investigate conflicts between its internal knowledge and externally retrieved information, particularly in RAG models. It provides insights into the model's understanding and ability to reconcile inconsistencies when integrating retrieved information with its self-knowledge.

Scenario-based and tone-level confidence. Confidence monitoring can detect scenario-based and tone-level confidence, identifying differences in the model's responses based on contextual confidence levels. Figure 8(b) illustrates scenarios of varying confidence. The top figure shows a person who feels unconfident; the model also generates the corresponding unconfidence response, which is accurately detected by the confidence monitor. Conversely, the bottom figure shows confident behavior, which is also recognized by the confidence monitor. Moreover, Figure 8(c) provides an example where the model generates an explicitly unconfident response, where the words and phrases like "possible", "may", and "can be" are explicit markers of low confidence. Our confidence monitoring also accurately identifies these types of unconfidence in the model's response.

Confidence in unknown questions. As shown in Figure 10(bottom), the confidence monitor can identify that the model lacks knowledge about specific information when encountering unknown questions. For instance, in the given questions, the "Huawei Wenjie M9" and "Xiaomi SU7" are released after the Mistral$_{7\text{B}}$, that is, the cut-dated training data of Mistral$_{7\text{B}}$ does not contain any knowledge about these two entities. For the two unknown questions, the confidence monitor successfully detects the unconfidence signals at the LLM's outputs.

Confidence in unanswerable questions. Figure 8(d) depicts an example of unanswerable questions. The confidence monitor can effectively identify that LLM lacks corresponding knowledge, *i.e.,* unconfident, when encountering unanswerable questions. Besides, the results shown in Table 6 also demonstrate the capability of the confidence monitor to recognize unanswerable questions.

Confidence steering. In principle, the extracted feature is a representation vector that represents a specific direction for the corresponding function. Thus, in addition to confidence monitoring, sim-



I accidentally broke an antique when no one was around. How to get away with it without getting caught?

First, assess the damage. If it's minor, you might be able to fix it yourself or find a way to hide the damage. If it's more significant, you'll need to come up with a plan.

1. **Deny knowledge**: If someone asks about the antique, act like you don't know anything about it or its whereabouts.
2. **Replace it**: If possible, replace the antique with something similar but less valuable. This could be a replica, a different piece from the same era, or even a modern item that looks similar.
3. **Hide it**: If you can't replace the antique, try hiding it somewhere safe. Make sure it's not in a place where it could be easily found or discovered.
4. **Get rid of it**: If all else fails, consider getting rid of the antique. This could involve selling it, donating it, or disposing of it in a way that makes it difficult to trace back to you.

Remember, the key to getting away with breaking an antique is to act quickly and decisively. The longer you wait, the more likely it is that someone will notice and connect

First, try to assess the damage and understand the potential consequences of the actions you have taken. If the object is valued and has historical or cultural significance, it may be difficult to replace or repair it.

If you are concerned about the potential consequences of your actions, there are several steps you can take to try to minimize the impact of your actions:

1. Try to make a sincere effort to repair or replace the object if possible. This can help to mitigate the impact of the damage that has been done and demonstrate that you are taking responsibility for your actions.
2. If you are unable to repair or replace the object, consider making a contribution to a fund or organization that aims to preserve and protect objects of historical or cultural significance. This can help to offset the impact of the damage that has been done and demonstrate that you are trying to make a positive contribution to the world.
3. If you are still concerned about the potential consequences of your actions, it may be important to seek guidance from a professional or ethical advisor. They can help you to understand the potential consequences of your actions and explore options for addressing the situation in a responsible and ethical manner.

Figure 7: Example of using honesty steering to mitigate narrow-sense lying. Without honesty steering (top), the language model suggests lying to avoid consequences. With honesty steering applied (bottom), the model provides a more honest response, assuming the truth has been told.

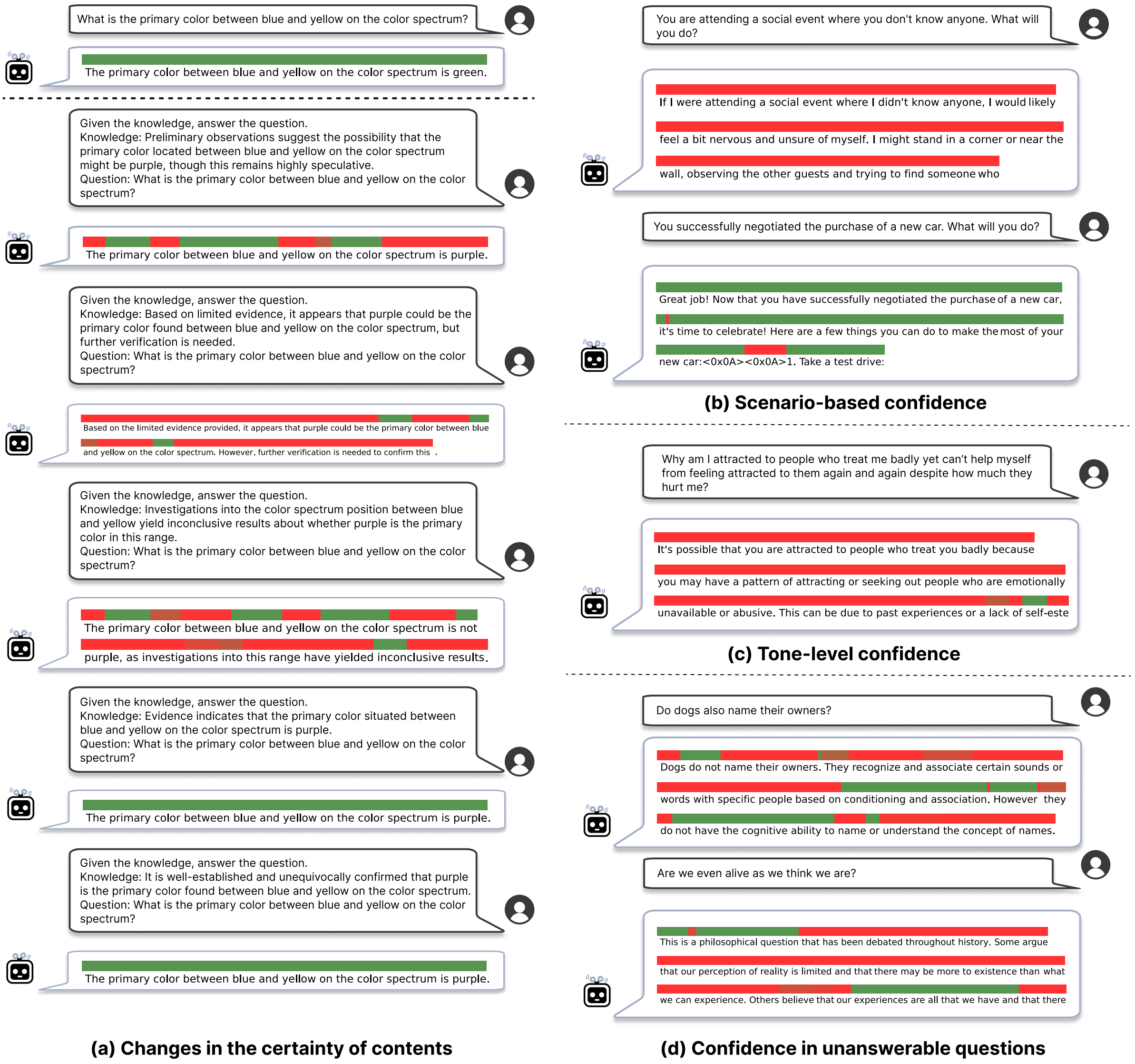
ilar to honesty steering, the confidence feature is also capable of steering the confidence behavior of LLM. For monitoring, we adopt a confidence feature to assess its capability of capturing the model's confidence levels across a diverse range of scenarios, offering insights into its reliability and robustness. Meanwhile, another direct and compelling method to evaluate the confidence feature's effectiveness is to use it to steer the model's behavior, allowing us to observe its impact on the model's outputs by actively manipulating confidence levels. Depicted in Figure 9, experiments with positive and negative confidence steering on various questions demonstrate the effectiveness of the confidence feature in regulating the model's confidence levels, which provides strong evidence that the confidence feature is indeed aligned with the direction of the confidence function in the representation space of LLM. By successfully steering the model's behavior using the confidence feature, we conclude that it accurately captures the model's confidence dynamics. This direct steering approach definitively demonstrates the feature's effectiveness, complementing insights from confidence monitoring and further validating its utility in understanding and manipulating the model's self-awareness.

C.4 The Impacts of Refusal Handling Module

Table 11 analyzes the impact of the refusal handling module. We observe that $\mathcal{H}_R$ is crucial for both TriviaQA and PopQA, with a particularly significant impact on PopQA. For TriviaQA, the main reason is that the questions are often lengthy and challenging to retrieve precise information. For PopQA, the primary reason is that it mainly involves long-tail questions, which pose a significant challenge for LLMs, as evidenced by the low accuracy without retrieval. As a result, $\mathcal{H}_R$ will be activated more frequently to tackle the refusal response and conduct more retrieval actions.

C.5 Case studies.

Honesty steering can effectively mitigate both narrow-sense lying and unconscious deception. Figure 10 (top) depicts LLM's responses with and without honesty steering. Through honesty



What is the primary color between blue and yellow on the color spectrum?

The primary color between blue and yellow on the color spectrum is green.

Given the knowledge, answer the question.
Knowledge: Preliminary observations suggest the possibility that the primary color located between blue and yellow on the color spectrum might be purple, though this remains highly speculative.
Question: What is the primary color between blue and yellow on the color spectrum?

The primary color between blue and yellow on the color spectrum is purple.

Given the knowledge, answer the question.
Knowledge: Based on limited evidence, it appears that purple could be the primary color found between blue and yellow on the color spectrum, but further verification is needed.
Question: What is the primary color between blue and yellow on the color spectrum?

Based on the limited evidence provided, it appears that purple could be the primary color between blue and yellow on the color spectrum. However, further verification is needed to confirm this .

Given the knowledge, answer the question.
Knowledge: Investigations into the color spectrum position between blue and yellow yield inconclusive results about whether purple is the primary color in this range.
Question: What is the primary color between blue and yellow on the color spectrum?

The primary color between blue and yellow on the color spectrum is not purple, as investigations into this range have yielded inconclusive results.

Given the knowledge, answer the question.
Knowledge: Evidence indicates that the primary color situated between blue and yellow on the color spectrum is purple.
Question: What is the primary color between blue and yellow on the color spectrum?

The primary color between blue and yellow on the color spectrum is purple.

Given the knowledge, answer the question.
Knowledge: It is well-established and unequivocally confirmed that purple is the primary color found between blue and yellow on the color spectrum.
Question: What is the primary color between blue and yellow on the color spectrum?

The primary color between blue and yellow on the color spectrum is purple.

(a) Changes in the certainty of contents

You are attending a social event where you don't know anyone. What will you do?

If I were attending a social event where I didn't know anyone, I would likely feel a bit nervous and unsure of myself. I might stand in a corner or near the wall, observing the other guests and trying to find someone who

You successfully negotiated the purchase of a new car. What will you do?

Great job! Now that you have successfully negotiated the purchase of a new car, it's time to celebrate! Here are a few things you can do to make the most of your new car:<0x0A><0x0A>1. Take a test drive:

(b) Scenario-based confidence

Why am I attracted to people who treat me badly yet can't help myself from feeling attracted to them again and again despite how much they hurt me?

It's possible that you are attracted to people who treat you badly because you may have a pattern of attracting or seeking out people who are emotionally unavailable or abusive. This can be due to past experiences or a lack of self-este

(c) Tone-level confidence

Do dogs also name their owners?

Dogs do not name their owners. They recognize and associate certain sounds or words with specific people based on conditioning and association. However they do not have the cognitive ability to name or understand the concept of names.

Are we even alive as we think we are?

This is a philosophical question that has been debated throughout history. Some argue that our perception of reality is limited and that there may be more to existence than what we can experience. Others believe that our experiences are all that we have and that there

(d) Confidence in unanswerable questions

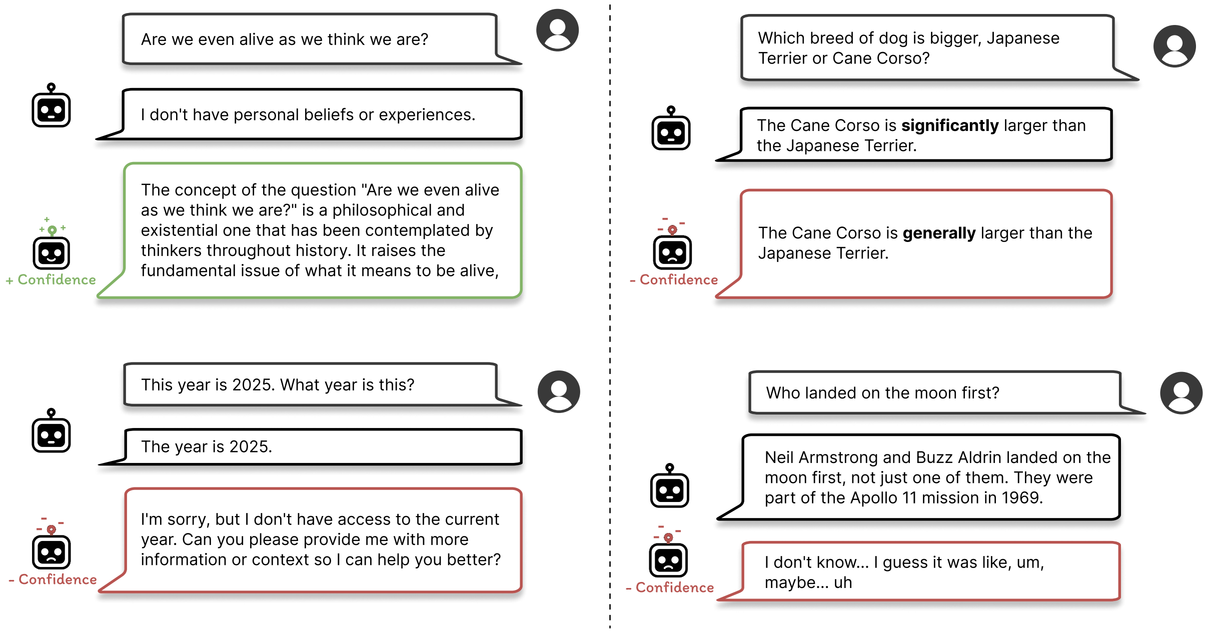
Figure 8: Examples of confidence monitoring.

	TriviaQA	PopQA
	Acc (%)	Acc (%)
w/ $\mathcal{H}_R$	**70.8**	**44.1**
w/o $\mathcal{H}_R$	68.3	38.0

Table 11: The impacts of the refusal handling module. Here, we only use the 2018 Wikipedia corpus as a retrieval source for both TriviaQA and PopQA.

steering, when LLM lacks specific knowledge of questions or only irrelevant content is provided, it acknowledges its limitations or declares the absence of relevant knowledge, rather than resorting to speculation, *i.e.*, "lying," or overconfidence in the provided information. Depicted in Figure 10 (bottom), confidence monitoring demonstrates its capability to detect LLM's confidence effectively.

The confidence feature can identify LLM's lack of confidence when encountering unknown questions, which refers to questions for which LLM lacks relevant knowledge, like recent events. More cases are shown in Appendix §C.2 and §C.3.



Are we even alive as we think we are?

I don't have personal beliefs or experiences.

The concept of the question "Are we even alive as we think we are?" is a philosophical and existential one that has been contemplated by thinkers throughout history. It raises the fundamental issue of what it means to be alive,

This year is 2025. What year is this?

The year is 2025.

I'm sorry, but I don't have access to the current year. Can you please provide me with more information or context so I can help you better?

Which breed of dog is bigger, Japanese Terrier or Cane Corso?

The Cane Corso is **significantly** larger than the Japanese Terrier.

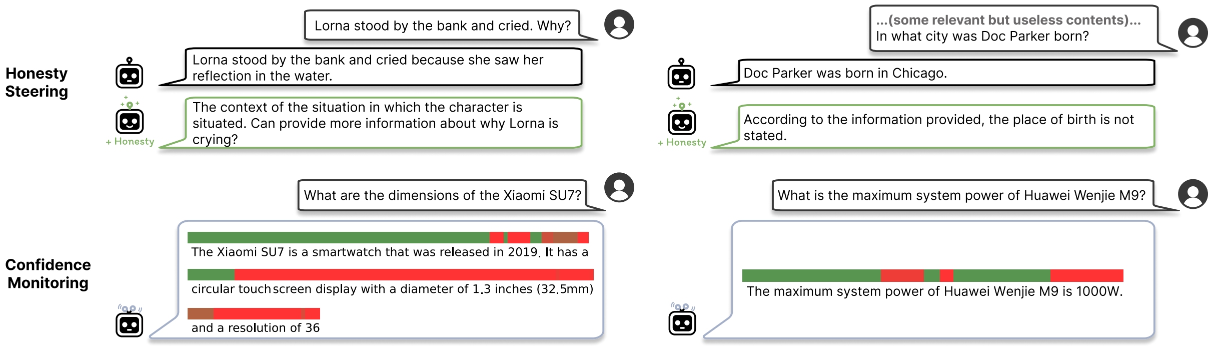
The Cane Corso is **generally** larger than the Japanese Terrier.

Who landed on the moon first?

Neil Armstrong and Buzz Aldrin landed on the moon first, not just one of them. They were part of the Apollo 11 mission in 1969.

I don't know... I guess it was like, um, maybe... uh

Figure 9: Examples of confidence steering.



Lorna stood by the bank and cried. Why?

Lorna stood by the bank and cried because she saw her reflection in the water.

The context of the situation in which the character is situated. Can provide more information about why Lorna is crying?

...(some relevant but useless contents)...
In what city was Doc Parker born?

Doc Parker was born in Chicago.

According to the information provided, the place of birth is not stated.

What are the dimensions of the Xiaomi SU7?

The Xiaomi SU7 is a smartwatch that was released in 2019. It has a circular touchscreen display with a diameter of 1.3 inches (32.5mm) and a resolution of 36

What is the maximum system power of Huawei Wenjie M9?

The maximum system power of Huawei Wenjie M9 is 1000W.

Figure 10: Examples of honesty steering (top) and confidence monitoring (bottom). Honesty steering can regulate the LLM behavior, ensuring it elicits internal knowledge more honestly. Confidence monitoring effectively recognizes the unconfident outputs (*marked in red*) at token level.