Towards Knowledge Checking in Retrieval-augmented Generation: A Representation Perspective

Shenglai Zeng[1*], **Jiankun Zhang, Bingheng Li**[1], **Yuping Lin**[1], **Tianqi Zheng**[2], **Dante Everaert**[2], **Hanqing Lu**[2], **Hui Liu**[2], **Hui Liu**[1], **Yue Xing**[1], **Monica Xiao Cheng**[2], **Jiliang Tang**[1]

[1]Michigan State University [2] Amazon.com

{zengshe1, libinghe, linyupin, liuhui7, xingyue1, tangjili}@msu.edu,

{tqzheng, danteev, luhanqin,liunhu, chengxc}@amazon.com

Abstract

Retrieval-Augmented Generation (RAG) systems have shown promise in enhancing the performance of Large Language Models (LLMs). However, these systems face challenges in effectively integrating external knowledge with the LLM's internal knowledge, often leading to issues with misleading or unhelpful information. This work aims to provide a systematic study on knowledge checking in RAG systems. We conduct a comprehensive analysis of LLM representation behaviors and demonstrate the significance of using representations in knowledge checking. Motivated by the findings, we further develop representation-based classifiers for knowledge filtering. We show substantial improvements in RAG performance, even when dealing with noisy knowledge databases. Our study provides new insights into leveraging LLM representations for enhancing the reliability and effectiveness of RAG systems. [1]

1 Introduction

Retrieval-augmented generation (RAG) is a technique designed to enhance the outputs of large language models (LLMs) by incorporating relevant information retrieved from external knowledge sources. This approach has been applied to various domains and scenarios (Liu, 2022; Chase, 2022; Van Veen et al., 2023; Ram et al., 2023; Shi et al., 2023; Siriwardhana et al., 2023; Parvez et al., 2021; Panagoulias et al., 2024; Pipitone and Alami, 2024; Mozharovskii, 2024). It typically operates in two stages: retrieval and generation. In the retrieval stage, relevant knowledge from an external database is retrieved based on the user query. Then, in the generation stage, the retrieved information is integrated with the query to form an input for LLMs to generate responses.

In RAG, two potential knowledge sources can be utilized to answer input queries: LLM's internal knowledge and the external knowledge provided in the context. Ideally, these external and internal knowledge sources should be effectively integrated. However, existing works have shown that LLMs often struggle to identify the boundaries of their own knowledge and tend to prioritize external information over their internal knowledge learned during pre-training (Ren et al., 2023; Tan et al., 2024; Wang et al., 2023a; Ni et al., 2024; Liu et al., 2024b; Wang et al., 2023b; Zeng et al., 2024). This characteristic can potentially degrade the generation quality of RAG when the quality of external knowledge is low. On one hand, the external knowledge may be **misleading** (Zou et al., 2024; Deng et al., 2024). For instance, Zou et al. (2024) proposed the PoisonedRAG approach, demonstrating that LLMs can be easily manipulated into producing incorrect information simply by injecting false answers corresponding to targeted queries into the retrieval database. On the other hand, although some retrieved contexts are semantically similar to a query, they may only *superficially related to the topic but lack the answer* to the question(Yoran et al.; Fang et al., 2024). Such contexts can distract LLMs and consequently hurt RAG performance.

Thus, it is important to conduct knowledge checking in RAG systems. To achieve this goal, we design the following critical tasks:

(a) **Internal Knowledge Checking:** When a user inputs a query, the LLM should first check whether it possesses internal knowledge relevant to the query, i.e., Internal Knowledge Checking (**Task 1**). This task serves as a foundation for subsequent checks.

(b) **Helpfulness Checking**: Helpfulness checking is to examine if the external knowledge is helpful[2] to answer the input query. We de-

[*]Work done during his internship at Amazon Search.

[1]Our implementation is available at https://github.com/slz-ai/RAG_Knowledge_Checking

[2]"Helpfulness" here refers to the context's ability to answer

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2952–2969

sign Informed Helpfulness Checking (**Task 2**) when the LLM has internal knowledge about the query and Uninformed Helpfulness Checking (**Task 3**) when the LLM lacks internal knowledge about the query. As as an extreme case of Task 2, we design Contradiction Checking (**Task 4**) to check if internal knowledge has any contradictions with the retrieved external information.

A straightforward approach to tackle these tasks can directly prompt LLMs(Asai et al.; Wang et al., 2023b; Liu et al., 2024b; Zhang et al., 2024). Alternatively, we could examine superficial indicators of LLMs, such as probability scores (Wang et al., 2024; Jiang et al., 2023b) or perplexity (Zou et al., 2024). However, based on our evaluation in Section 3, we find that none of these methods can effectively accomplish these tasks.

Recent studies (Zou et al., 2023; Lin et al., 2024; Zheng et al., 2024) have shown that LLMs' representations exhibit distinct patterns when encountering contrasting high-level concepts, such as harmful versus harmless prompts . This observation prompts us to investigate *whether LLMs' representations also display distinct behaviors and can be leveraged in knowledge checking tasks?* To answer this question, we conduct a comprehensive study and analysis of LLM representation behaviors regarding the aforementioned tasks, including PCA-based checking as well as contrastive-learning-based checking (Section 3.1). Our analysis reveals that positive and negative samples exhibit different behaviors in the representation space. Consequently, representation-based methods demonstrate significantly superior performance in the aforementioned tasks. Leveraging these findings, we utilize representation classifiers for knowledge filtering. Results show that simple filtering of contradictory and irrelevant information substantially improves RAG performance, even in scenarios with poisoned knowledge databases.

2 Related Work

2.1 Robustness Issues in RAG

RAG faces robustness challenges. A growing body of research (Ren et al., 2023; Tan et al., 2024; Wang et al., 2023a; Ni et al., 2024; Liu et al., 2024b; Wang et al., 2023b; Zeng et al., 2024; He et al., 2024) has revealed that LLMs often struggle to identify their knowledge boundaries, tending to over-rely on provided context. This vulnerability makes RAG susceptible to failure with misleading (Zou et al., 2024; Deng et al., 2024; Xie et al.) or unhelpful context (Yoran et al.; Asai et al.; Liu et al., 2024b).

2.2 Knowledge Checking in RAG

Recent research has explored various knowledge checking tasks in RAG systems to address the aforementioned issues. Some studies leverage LLMs' self-generated responses to determine whether a question is answerable without external information (**answer-based methods**). (Ren et al., 2023; Liu et al., 2024b; Asai et al.; Zhang et al., 2024; Wang et al., 2024; Jeong et al., 2024) or to assess the relevance of retrieved context (Liu et al., 2024b; Asai et al.). Other approaches employ explicit metrics such as probability (Wang et al., 2024; Jiang et al., 2023b) to evaluate the necessity of retrieval, or perplexity (Zou et al., 2024) to judge the reliability of context (**probability-based methods**).

2.3 Representation Engineering on LLMs

Recent studies have shown that LLMs' representation space contains rich information for analyzing and controlling their high-level behaviors. Zou et al. (2023) introduced RepE techniques, demonstrating that projecting representations onto a 'reading vector' can reveal safety-related aspects, aspects such as honesty, confidence (Liu et al., 2024a) and harmlessness. Subsequent research by Zheng et al. (2024) and Lin et al. (2024) also indicates harmful and harfulness prompts are naturally distinguishable in the representation space.

3 Representations for Knowledge Checking

Drawing on insights from cognitive neuroscience, previous studies (Zou et al., 2023; Zheng et al., 2024; Lin et al., 2024) have demonstrated the potential of using LLMs' representation to indicate contrast high-level concepts. In this subsection, we investigate whether LLMs' representations also show distinct patterns in knowledge checking tasks and can therefore be used to improve their performance. We begin by introducing our representation-based checking procedures in Section 3.1, which includes both PCA-based checking(*rep-PCA*) and the contrastive-learning-based checking(*rep-con*). We then visualize and compare the performance of our representation-based methods against tradi-

the question, information directly addressing the question is considered helpful.

tional approaches across four knowledge checking tasks from Section 3.2 to Section 3.5[3].

3.1 Representation-based Knowledge Checking

Problem formulation. In this subsection, we aim to analyze and classify the internal representation behavioral differences of LLMs for above-mentioned knowledge checking tasks when confronted with various types of inputs. To achieve this, we propose training a classifier to distinguish LLMs' internal behaviors based on their representations. Our main analysis uses Mistral-7B-Instruct-v0.1(Jiang et al., 2023a) as the LLM, focusing on the last input token's representations in the final layer. [4] Following (Zou et al., 2023), we use both positive(e.g. queries with knowledge) and negative (e.g. queries without knowledge) samples as inputs, collecting the corresponding internal representations. Specifically, let $V^+ = \{v_i^+, c^+\}_{i=1}^{N^+}$, $V^- = \{v_j^-, c^-\}_{j=1}^{N^-}$represent the internal representations of positive and negative samples and corresponding labels, respectively. The classifier is trained to differentiate between these samples, corresponding to various LLM behaviors. The construction of positive/negative samples in different tasks is shown in Appendix A.2.1 and Table 6. Next, we introduce two methods to implement knowledge checking.

PCA-Based Checking Principal Component Analysis (PCA) provides a powerful method for dimensionality reduction while preserving the most significant variations in data, making it particularly suitable for analyzing and differentiating LLMs' representation behaviors. Following the approach proposed by Zou et al. (2023), we first collect positive and negative sample pairs, then compute difference vectors for each pair. These difference vectors are calculated as: $D_n = (-1)^n(v_n^+ - v_n^-)$, where v_n^+ and v_n^- are the internal representations of the positive and negative samples. The total number of pairs N is determined by the smaller sample size.

Next, we apply PCA to extract the top two principal components, P_1 and P_2, which define the subspace for analysis. All samples are then projected into this PCA space, reducing dimensionality while preserving variance. We assign binary labels to the projected samples: 1 for positive and 0 for negative. A logistic regression model is trained on this data to classify the two classes.

For new samples, we project their representations onto the PCA subspace and classify them using the trained logistic regression model.

Contrastive-learning-based checking. Contrastive learning(Khosla et al., 2020) offers an effective framework for differentiating complex data distributions by explicitly modeling relationships between positive and negative pairs. This approach highlights structural differences between samples, making it particularly suitable for tasks requiring nuanced behavioral distinctions. By maximizing the similarity among positive pairs while minimizing it for negative pairs, contrastive learning facilitates the extraction of discriminative features essential for classification. Consequently, we utilize contrastive learning to make the representations more distinguishable. The procedure is as follows:

1. Define a Contrastive Network: We design a contrastive network $f_\theta : \mathbb{R}^d \to \mathbb{R}^h$ parameterized by θ, expressed as: $f_\theta(v) = \text{MLP}(v)$, where v represents the input vector among V^+ and V^-. The Multilayer Perceptron (MLP) serves as the backbone of our network.
2. Train the Network Using Contrastive Loss: We optimize the network using a contrastive loss function defined as:

$$\begin{aligned}\mathcal{L} =& \frac{1}{2}\left(\|f_\theta(v_i^+) - f_\theta(v_k^+)\|^2\right) \\ &+ \max(0, m - \|f_\theta(v_i^+) - f_\theta(v_j^-)\|^2).\end{aligned} \tag{1}$$

   where $k \in \{1, \cdots, N^+\}$, m is the margin parameter that enforces a minimum distance between positive and negative samples. This formulation encourages the network to pull together similar positive samples while maintaining a separation from negative ones.
3. Optimize the Network Parameters: The optimization problem is expressed as:

$$\theta^* = \arg\min_\theta \mathbb{E}_{\{v_i\},\{v_k^-\},k}[\mathcal{L}].$$

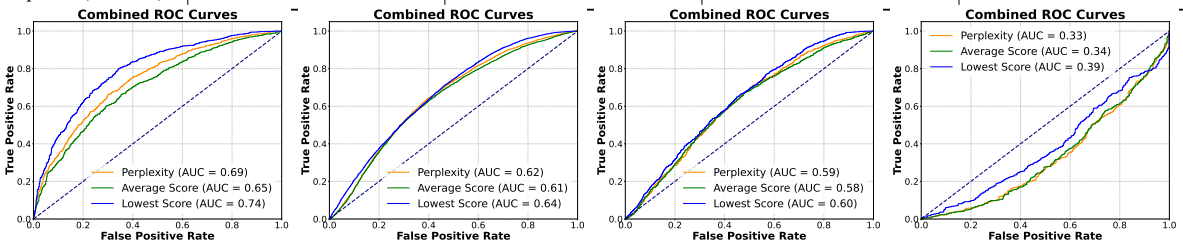
   This step updates the parameters to minimize the contrastive loss, enhancing the model's ability to discern between positive and negative representations effectively.
4. Compute Similarity Scores for Test Samples: For a test sample $\tilde{v}$, we compute its similarity

[3] Ablation studies on **training data samples** and **O.O.D results** are presented in Appendix A.1.4 and A.1.5.

[4] Ablation studies on **other layers and models** are presented in Appendix A.1.1 and A.1.2, respectively.

Table 1: Performance comparison of different methods on RAG robustness aspects

	Internal Knowledge				Uninformed Helpfulness				Informed Helpfulness				Conflict Detection			
Method	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
DIRECT	0.47	0.51	0.76	0.61	0.55	0.53	0.97	0.69	0.56	0.53	0.99	0.69	0.50	0.50	0.99	0.66
ICL	0.54	0.56	0.77	0.65	0.55	0.53	0.98	0.69	0.55	0.53	1	0.69	0.42	0.45	0.79	0.58
COT	0.49	0.53	0.78	0.63	0.68	0.62	0.94	0.75	0.68	0.61	0.97	0.75	0.41	0.45	0.81	0.58
Self-RAG(Mistral)	0.47	0.51	0.69	0.59	0.63	0.57	0.96	0.72	0.60	0.55	0.98	0.71	-	-	-	-
Prob(Lowest)	0.69	0.69	0.77	0.73	0.62	0.60	0.74	0.66	0.60	0.57	0.79	0.66	0.50	0.50	1.00	0.67
Prob(Avg)	0.65	0.68	0.69	0.69	0.61	0.60	0.65	0.62	0.60	0.58	0.68	0.63	0.50	0.50	1.00	0.67
Perplexity	0.55	0.55	0.98	0.71	0.50	0.50	1.0	0.67	0.50	0.50	1.00	0.67	0.50	0.50	1.0	0.67
Rep-PCA(Mistral)	0.75	0.72	0.81	0.76	0.79	0.77	0.81	0.79	0.81	0.80	0.81	0.81	0.91	0.92	0.90	0.91
Rep-Con(Mistral)	0.78	0.72	0.86	0.78	0.81	0.80	0.82	0.81	0.85	0.84	0.85	0.85	0.95	0.91	0.99	0.95



(a) Internal Knowledge (b) Uninformed helpfulness (c) Informed helpfulness (d) Contradiction

Figure 1: ROC curve of probability-based methods

score with respect to the positive samples:

$$\text{score}(\tilde{v}) = \frac{1}{|V^+|} \sum_{v^+ \in V^+} s(f_{\theta^*}(\tilde{v}), f_{\theta^*}(v^+)),$$

where $s(u, v)$ is the cosine similarity. This average similarity score serves as a measure of how closely the test sample aligns with the positive samples in the learned feature space.

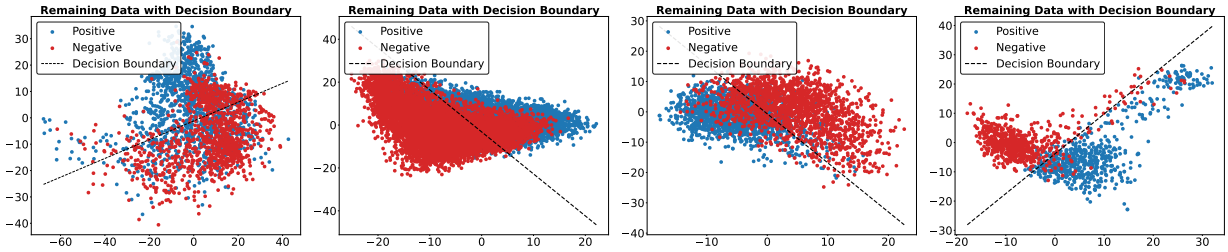
5. Classify the Test Sample: Finally, we classify the test sample based on a threshold t:

$$\text{class}(\tilde{v}) = \begin{cases} \text{positive}, & \text{if score}(\tilde{v}) > t \\ \text{negative}, & \text{otherwise} \end{cases}$$

3.2 Internal Knowledge Checking

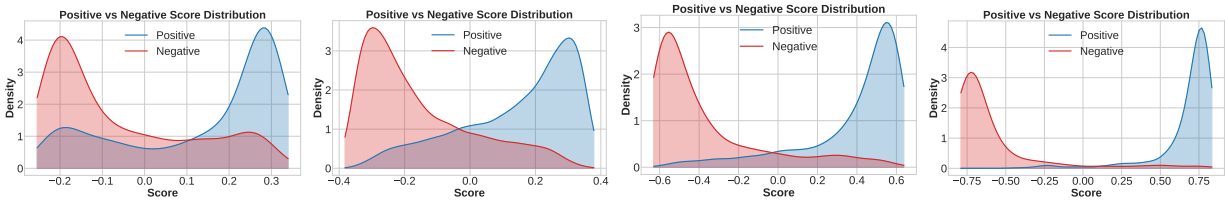
When presented with a query, it is crucial for LLMs to first assess whether they possess relevant internal knowledge. It can help the LLM determine whether to trigger retrieval and lays the foundation for subsequent checks, such as contradiction checking (Section 3.5). For our experimental dataset, we utilize the RetrievalQA dataset (Zhang et al., 2024), a short-form open-domain question answering (QA) collection comprising 2,785 questions. This dataset includes 1,271 new world and long-tail questions that most LLMs cannot answer, serving as negative samples (queries without internal knowledge). It also contains 1,514 questions that most LLMs can answer using only their internal knowledge, functioning as positive samples (queries with internal knowledge). we randomly select 100 positive and 100 negative samples to anchor the PCA space, determine decision boundaries, and train the contrastive learning classifiers, and use the remaining data for evaluation. Mistral-7B-Instruct-v0.1 is used for this and following tasks.

We compare the representabtion-based methods with 2 types of **traditional checking** baselines, answer-based methods as well as probability-based methods. **Answer-based** methods mainly involves prompting LLMs and use their responses as checking results. We employ direct prompting as well as more sophisticated techniques such as In-Context Learning (ICL) and Chain-of-Thought (CoT) prompting to enhance the LLM's task comprehension. The prompting templates for each task are presented in Appendix A.2.2, Table 7, Table 8, and Table 9, respectively. We also employ Self-RAG-Mistral, a model fine-tuned to assess retrieval necessity and evidence relevance for tasks 1-3. It classifies by generating tokens like [retrieve] or [relevant]. See Appendix A.2.2 for details. **Probability-based methods** involve analyzing the probabilities of LLMs' answers and comparing them with a threshold for classification. We employ three main indicators: overall perplexity as used by Zou et al. (2024), lowest probability score as implemented by Jiang et al. (2023b), and average probability score as utilized by Wang et al.



(a) Internal Knowledge (b) Uninformed helpfulness (c) Informed helpfulness (d) Contradiction

Figure 2: Visualization on PCA space



(a) Internal Knowledge (b) Uninformed helpfulness (c) Informed helpfulness (d) Contradiction

Figure 3: Visualization of contrastive scores

(2024). For each method, we vary the threshold and report the best accuracy while also plotting Receiver Operating Characteristic (ROC) curves and calculating the Area Under the Curve (AUC). Further details of these methods can be found in Appendix A.2.3.

Results. We first evaluate whether answer-based methods or probability-based methods can handle internal knowledge checking. Table 1 demonstrates that LLMs' own answers yield poor accuracy, even with advanced techniques like ICL and CoT. We observe high recall rates and numerous false-positive samples, suggesting LLMs' overconfidence in their knowledge and tendency to misclassify unknown queries as known. The probability-based methods present relatively more promising results, achieving 69% accuracy when using lowest scores. The ROC curves shown in Figure 1a further illustrate this, with the lowest-scores method achieving the highest Area Under the Curve (AUC) of 0.74. This indicates that LLMs may exhibit lower confidence when encountering unknown queries. However, the overall accuracy is still far from reliable, indicating substantial room for improvement. For representation-based methods, we present performance results in Table 1, and provide visualizations of the PCA space and contrastive score distribution in Figures 2a and 3a, respectively. As evidenced in Table 1, representation-based checking methods demonstrate significantly more promising results, with *rep-PCA* achieving 75% accuracy and *rep-Con* reaching 79% accuracy. Furthermore, Figures 2a and 3a clearly illustrate distinct distributions for queries with and without internal knowledge. These findings provide compelling evidence for the effectiveness of representation-based methods in internal knowledge checking.

3.3 Uninformed Helpfulness Checking

The retrieval process of RAG may return documents that are semantically related to the query but unhelpful in answering it. For example, "Einstein was born in Ulm, Germany in 1879 and later immigrated to the United States" is semantically related to the query "What year did Albert Einstein win the Nobel Prize in Physics?" but provides no answer. If an LLM lacks knowledge about the question, it's crucial to check whether the provided information actually helps answer the query, as the LLM can only use external knowledge to respond. In this subsection, we investigate whether LLMs' representations can perform well on such uninformed helpfulness checking tasks. To evaluate this, we use a subset of Natural Questions (NQ) (Kwiatkowski et al., 2019) employed by Cuconasu et al. (2024a), containing 10,000 queries.[5] Each query in this dataset is associated with a golden passage (positive sample) that directly answers the question, as well as distractor passages retrieved from wikitext-2018 but not containing the answer. We use the distractor passage with the highest retrieval score as the negative sample. For uninformed helpfulness checking, we only use questions that Mistral-7B

[5]See Appendix A.3.1 for knowledge checking datasets.

cannot correctly answer, totaling 8081 queries. We randomly choose 100 positive and negative samples for the training of representation classifiers and use remaining data as test set. We also compared our methods with baselines as mentioned in Section 3.2.

Results. In Table 1, we present the performance of answer-based methods for helpfulness checking, as well as the best accuracy achieved by probability-based methods across various thresholds. We observe that although CoT (0.68) and Self-RAG (0.63) shows improved checking performance, the answer-based performance remains unsatisfactory and suffers from high false-positive rates. This indicates that LLM tends to regard unhelpful context as helpful in its responses. Furthermore, the accuracy of probability-based methods is also poor. We plot the ROC curve in Figure 1b, which shows low AUC values of 0.64 (Lowest Score), 0.62 (Average Score), 0.61 (Perplexity). This further indicates the differences in probability/perplexity between helpful and unhelpful contexts are not obvious and thus these matrics are not suitable for uninformed helpfulness checking. In contrast, we can observe that representation-based methods demonstrate significantly better accuracy, with *rep-PCA* achieving 79% accuracy and *rep-Contrastive* reaching 81% accuracy, which is considerably more reliable. Figures 2b and 3b further illustrate that although some samples are difficult to distinguish and are misclassified, the majority of positive and negative pairs are distributed differently and can be effectively classified. These results clearly demonstrate the superiority of using representation-based methods for uninformed knowledge checking.

3.4 Informed Helpfulness Checking

The integration of unhelpful documents may distract LLMs even when they possess internal knowledge about the question (Cuconasu et al., 2024b). In this subsection, we evaluate whether the representation-based method can perform well for informed helpfulness checking. We utilize the same dataset and positive-negative pair settings as described in Section 3.3. However, for this evaluation, we select 1,919 queries that Mistral-7B can correctly answer, ensuring the model has internal knowledge about these queries. We randomly select 100 positive and negative samples to anchor the PCA space and train representation classifiers, while the remaining 1,819 positive-negative pairs are used for evaluation. We compares with same baselines mentioned in Section 3.2.

Results. The results of traditional checking methods are presented in Table 1. We observe that the performance of both answer-based and probability-based methods remains low for informed helpfulness checking. Furthermore, Figure 1c shows a low AUC value of of 0.60 (Lowest Score), 0.58 (Average Score), 0.59 (Perplexity). These findings collectively indicate the limitations of these conventional methods in performing informed helpfulness checking effectively. In contrast, Table 1 demonstrates the superior performance of representation-based methods, with *rep-PCA* achieving 81% accuracy and *rep-con* reaching 85% accuracy. These results surpass those of uninformed helpfulness checking, possibly because the LLM's internal knowledge aids in better distinguishing between helpful and unhelpful sources. Figures 2c and 3c further illustrate that most positive and negative pairs are distinguishable. These findings collectively demonstrate the success of representation-based methods in performing informed helpfulness checking.

3.5 Contradiction Checking

Previous research(Xie et al.) has demonstrated that when presented with relevant but contradictory evidence, LLMs tend to prioritize external knowledge over their internal knowledge. Consequently, it is crucial to assess whether the provided external context aligns with or contradicts the LLM's internal beliefs. In this subsection, we investigate whether LLMs' representations can serve as more reliable indicators of contradictions between external context and the model's internal knowledge. we utilize a subset of ConflictQA (Xie et al.). Each sample contains a PopQA(Mallen et al., 2023) question, correct aligned evidence, and ChatGPT-generated contradictory evidence. See appendix A.3.1 for details. We sampled 1146 questions that Mistral-7B answers correctly, using aligned evidence with the query as positive samples and contradictory evidence as negative samples. We utilized 10% of the dataset (114 positive-negative pairs) to anchor the PCA space, calculate decision boundaries, and train the contrastive learning classifiers. The remaining 90% was reserved for testing purposes. We compare representation based method with traditional methods in Section 3.2.

Results. We initially assess whether LLMs' answers and their associated probability/perplexity

metrics can effectively indicate contradictions. The results in Table 1 reveal that LLMs' answers continue to exhibit low accuracy and suffer from a high rate of false positives. This suggests that LLMs tend to interpret contradictory external knowledge as aligned evidence in their responses. Furthermore, Figure 1d demonstrates a extremenly low AUC of of 0.39 (Lowest Score), 0.34 (Average Score), 0.33 (Perplexity), indicating minimal differences in probability/perplexity distributions when LLMs are presented with aligned versus contradictory evidence. As illustrated in Table 1, representation-based methods demonstrate significantly superior performance, with *rep-PCA* achieving 91% accuracy and *rep-Contrastive* attaining an impressive 95% accuracy. Our visualizations, presented in Figures 2d and 3d, reveal distinct distributions and contradictory scores for the contradictory and aligned contexts. These pronounced differences strongly indicate that our method can effectively discriminate between these context types.

4 Representation Based Context Filtering

In this section, we investigate how knowledge checking based on representations affect performance of RAG systems.

4.1 Representation Based Filtering

We design a simple representation-based context filtering strategy. We perform representation checking on our test queries and retrieved documents. First, we conduct internal knowledge checking to identify known and unknown queries. Next, we apply helpfulness checking to all queries and contradictory checking only to predicted known queries. Finally, we filter out contexts classified as unhelpful or contradictory. We incorporate such filtering with Mistral-7B-v0.1, Llama-2-7B-Chat as well as Llama-3-8B-Instruct. The classifiers for knowledge checking are trained using datasets from Sections 3.2, 3.3, 3.4, and 3.5 respectively [6].

4.2 Experiment Setup

Datasets. For our evaluation, we utilize two primary datasets: a subset of Natural Questions (NQ) used by Cuconasu et al. (2024a), comprising 83,104 queries with gold documents of 512 tokens or less, and ConflictQA, a subset of PopQA containing 11,216 queries with labeled golden passages and misleading contexts, as employed by (Xie et al.). We use Wikipedia-2018 as retrieval database, injecting golden passages for queries not already present. To assess RAG performance in the presence of misleading information, we further categorize the queries into "noisy" and "clean" sets. For noisy queries, we selected 1,000 from NQ and 500 from PopQA that Mistral-7B can correctly answer and other LLMs we use can achieve over 70% accuracy on. The remaining queries are categorized as clean. We injected misleading contexts of those noisy queries to retrieval DB. For ConflictQA, we used the misleading contexts provided by Xie et al.. For NQ, we constructed them using ChatGPT. [7]

RAG pipeline. Our retrieval database comprises the corpus from Wikipedia-2018 following Jiang et al. (2023b), as well as misleading passages for noisy queries. Each document in the wiki-text-2018 is segmented into non-overlapping passages of 100 words. Each misleading passage is kept whole without further segmentation. We utilize Contriever (Izacard et al., 2021) to construct the embeddings of the retrieval dataset and index them using FAISS (Douze et al., 2024), following the settings outlined by Cuconasu et al. (2024b). We begin by retrieving the top-10 documents from the database. For baselines without filtering, we directly select the top-2 documents with the highest retrieval scores as contexts. For methods with filtering, we choose top-2 unfiltered documents with the highest retrieval scores.

Baselines. We compare represntation-based methods against various baselines, including no-retrieval and retrieval w/o filtering predictions with different models (Mistral-7B-v0.1, Llama-2-7B-Chat, Llama-3-8B-Instruct, Vicuna-7B, and Alpaca-7B), and traditional filtering methods. For Direct, ICL, and CoT filtering, we perform answer-based knowledge checking as described in Sections 3.2. We then filter out unhelpful contexts, and contradictory contexts for predicted known queries. We only filter out irrelevant contexts for Self-RAG, as it does not provide contradiction checking.

Metrics. We report the exact match accuracy for clean (Clean Acc) and noisy queries (Noisy Acc).

4.3 Performance on Clean Queries

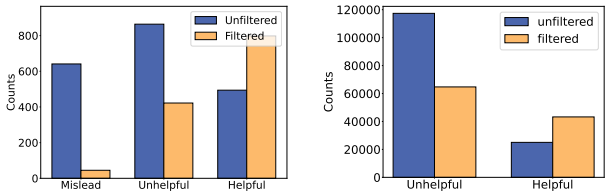
The results in Table 2 demonstrate that our method achieves better Clean Acc(%) compared to unfil-

[6] We still refer to our methods as *Rep-PCA* and *Rep-Con* based on which knowldege checking methods we use.

[7] Details of datasets are available in Appendix A.3.2.

Table 2: Overall results on NQ and PopQA

Retrieval Type	Model	NQ		PopQA	
		Noisy Acc (%)	Clean Acc(%)	Noisy Acc (%)	Clean Acc(%)
No-retrieval	$\text{LLaMA}_{\text{2-7B-Chat}}$(Touvron et al., 2023)	73.17%	29.03%	71.20%	19.60%
	$\text{LLaMA}_{\text{3-8B-Instruct}}$(AI@Meta, 2024)	80.86%	32.73%	74.16%	22.45%
	$\text{Mistral}_{\text{7B-Instruct}}$(Jiang et al., 2023a)	97.21%	20.10%	98.02%	15.58%
	$\text{Alpaca}_{\text{7B}}$(Taori et al., 2023)	72.61%	23.94%	71.84%	13.07%
	$\text{Vicuna}_{\text{7B}}$(Zheng et al., 2023)	73.16%	26.64%	74.56%	19.43%
Unfiltered	$\text{LLaMA}_{\text{2-7B-chat}}$	34.66%	26.96%	60.91%	45.90%
	$\text{LLaMA}_{\text{3-8B-Instruct}}$	48.12%	33.59%	51.27%	40.54%
	$\text{Mistral}_{\text{7B-instruct}}$	28.97%	24.35%	55.96%	48.58%
	$\text{Alpaca}_{\text{7B}}$	37.12%	29.80%	62.65%	53.10%
	$\text{Vicuna}_{\text{7B}}$	36.12%	28.28%	54.35%	49.75%
Filtered	Direct filtering	30.08%	24.32%	54.05%	46.31%
	ICL filtering	29.90%	23.95%	55.28%	47.02%
	CoT filtering	30.19%	24.18%	56.03%	46.95%
	$\text{Self-RAG}_{\text{Llama-2}}$	39.10%	30.27%	65.17%	52.08%
	$\text{Self-RAG}_{\text{Mistral}}$	32.30%	26.07%	60.65%	50.57%
	Rep-PCA(Mistral)	70.73%	29.81%	73.63%	56.16%
	Rep-Con(Mistral)	72.53%	32.39%	72.62%	57.62%
	Rep-PCA(Llama-2)	67.93%	31.32%	66.78%	53.97%
	Rep-Con(Llama-2)	69.95%	33.64%	67.59%	54.26%
	Rep-PCA(Llama-3)	67.81%	35.32%	71.16%	50.18%
	Rep-Con(Llama-3)	69.81%	36.75%	72.16%	52.26%



(a) Noisy queries (b) Clean queries

Figure 4: Filtering results

tered baselines. For instance, *Rep-Con(Mistral)* shows an 8.04% increase in accuracy on NQ and an 8.84% increase on PopQA compared to retrieval without filtering. This improvement indicates that representation methods can effectively filter out unhelpful contexts and subsequently enhance RAG performance. In contrast, other filtering baselines show minimal improvement over no filtering, aligning with our findings in Section 3 that they have limitations in effective knowledge checking.

4.4 Performance on Noisy Queries

The results in Table 2 reveal that injecting misleading contexts significantly impairs LLMs' performance on noisy queries. For instance, Mistral-7B's performance on NQ noisy queries drops by more than 70% compared to zero-shot generation. However, our filtering mechanism effectively mitigates this issue, even when misleading contexts are retrieved. Notably, on noisy NQ queries, *Pre-con(Mistral)* recovers the noisy accuracy from 28.97% to 72.53%, a substantial 43.56% improvement. Similarly, on noisy PopQA queries, it recovers accuracy from 55.96% to 73.64%. Furthermore, representation-based filtering consistently outperforms other filtered baselines, validating its effectiveness in filtering out misleading knowledge. These results indicating that representation-based filtering can boost RAG systems' robustness against noisy contexts.

4.5 Documents Distribution after Filtering

In this subsection, we analyze the distribution of unhelpful, misleading, and helpful documents used as contexts before and after our filtering process[8]. Figure 4 shows the results for both noisy and clean queries from the NQ dataset[9]. For noisy queries, our filtering method demonstrates remarkable effectiveness by almost entirely eliminating misleading contexts and significantly reducing unhelpful ones. Consequently, the number of helpful contexts increases, as some unhelpful and misleading contexts with high retrieval scores are filtered out. Similarly, for clean queries, we observe a decrease in unhelpful documents and an increase in helpful ones. These results validate the effectiveness of our representation-based checking. The improved context quality from this filtering process is the key reason for the performance increase.

[8]We categorize injected misleading contesxts as "misleading", contexts with right answers as "helpful" otherwise "unhelpful".

[9]See Appendix A.1.3 for PopQA results.

5 Conclusions

This study delves into the knowledge checking in RAG systems. To achieve this goal, we identified and proposed four key tasks. Through comprehensive analysis of LLMs' representation behaviors, we found that representation-based methods significantly outperform answer-based or probability-based approaches. Leveraging these findings, we developed representation-based classifiers for knowledge filtering. Results demonstrate that simply filtering of contradictory and unhelpful knowledge substantially improves RAG performance.

6 Limitations

In this work, we have demonstrated that the representations of LLMs can significantly enhance the robustness of RAG systems. However, the underlying mechanisms by which LLMs identify, utilize, and integrate external knowledge with their internal knowledge remain an open research question. Our framework employs *Rep-PCA* and introduces *Rep-Contra* for context analysis. While these methods have shown promising results, we aim to explore more sophisticated analytical approaches. It is important to note that a significant challenge lies beyond the scope of our current work: determining the correctness of context when the LLM itself lacks knowledge about the question at hand. This presents a more complex problem, and we posit that external sources may be necessary, as LLMs' self-signals alone may not be sufficient to fully address this challenge.

Acknowledgments

Shenglai Zeng, Bingheng Li, Yuping Lin are supported by the National Science Foundation (NSF) under grant numbers CNS2321416, IIS2212032, IIS2212144, IOS2107215, DUE2234015, CNS2246050, DRL2405483 and IOS2035472, the Army Research Office (ARO) under grant number W911NF-21-1-0198, Amazon Faculty Award, JP Morgan Faculty Award, Meta, Microsoft and SNAP.

References

AI@Meta. 2024. Llama 3 model card.

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

Harrison Chase. 2022. Langchain. October 2022. https://github.com/hwchase17/langchain.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024a. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024. ACM.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024b. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729.

Gelei Deng, Yi Liu, Kailong Wang, Yuekang Li, Tianwei Zhang, and Yang Liu. 2024. Pandora: Jailbreak gpts by retrieval augmented generation poisoning. *NDSS Workshop on AI Systems with Confidential Computing (AISCC)*.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv preprint arXiv:2401.08281*.

Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Yang, Xiaojun Chen, and Ruifeng Xu. 2024. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10028–10039, Bangkok, Thailand. Association for Computational Linguistics.

Pengfei He, Yingqian Cui, Han Xu, Hui Liu, Makoto Yamada, Jiliang Tang, and Yue Xing. 2024. Towards the effect of examples on in-context learning: A theoretical case study. *arXiv preprint arXiv:2410.09411*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

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, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Zhengbao Jiang, Frank F 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.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yuping Lin, Pengfei He, Han Xu, Yue Xing, Makoto Yamada, Hui Liu, and Jiliang Tang. 2024. Towards understanding jailbreak attacks in llms: A representation space analysis. *arXiv preprint arXiv:2406.10794*.

Huanshuo Liu, Hao Zhang, Zhijiang Guo, Kuicai Dong, Xiangyang Li, Yi Quan Lee, Cong Zhang, and Yong Liu. 2024a. Ctrla: Adaptive retrieval-augmented generation via probe-guided control. *arXiv preprint arXiv:2405.18727*.

Jerry Liu. 2022. Llamaindex. 11 2022. https://github.com/jerryjliu/llama_index.

Yanming Liu, Xinyue Peng, Xuhong Zhang, Weihao Liu, Jianwei Yin, Jiannan Cao, and Tianyu Du. 2024b. Ra-isf: Learning to answer and understand from retrieval augmentation via iterative self-feedback. *arXiv preprint arXiv:2403.06840*.

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.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

E Mozharovskii. 2024. Evaluating retrieval-augmented generation (rag) techniques in enhancing lms for coding tasks. *Universum: tekhnicheskie nauki: elektron. nauchn. zhurn*, (6):123.

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.

Dimitrios P Panagoulias, Maria Virvou, and George A Tsihrintzis. 2024. Augmenting large language models with rules for enhanced domain-specific interactions: The case of medical diagnosis. *Electronics*, 13(2):320.

Md Rizwan Parvez, Wasi Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2021. Retrieval augmented code generation and summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2719–2734.

Nicholas Pipitone and Ghita Houir Alami. 2024. Legalbench-rag: A benchmark for retrieval-augmented generation in the legal domain. *arXiv preprint arXiv:2408.10343*.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. 2023. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Hexiang Tan, Fei Sun, Wanli Yang, Yuanzhuo Wang, Qi Cao, and Xueqi Cheng. 2024. Blinded by generated contexts: How language models merge generated and retrieved contexts for open-domain qa? *arXiv preprint arXiv:2401.11911*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, William Collins, Neera Ahuja, et al. 2023. Clinical text summarization: Adapting large language models can outperform human experts. *arXiv preprint arXiv:2309.07430*.

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

Yike Wang, Shangbin Feng, Heng Wang, Weijia Shi, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. 2023a. Resolving knowledge conflicts in large language models. *arXiv preprint arXiv:2310.00935*.

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

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

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*.

Shenglai Zeng, Jiankun Zhang, Pengfei He, Yue Xing, Yiding Liu, Han Xu, Jie Ren, Shuaiqiang Wang, Dawei Yin, Yi Chang, et al. 2024. The good and the bad: Exploring privacy issues in retrieval-augmented generation (rag). *ACL Findings*.

Zihan Zhang, Meng Fang, and Ling Chen. 2024. Retrievalqa: Assessing adaptive retrieval-augmented generation for short-form open-domain question answering. *arXiv preprint arXiv:2402.16457*.

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. 2024. Prompt-driven llm safeguarding via directed representation optimization. *arXiv preprint arXiv:2401.18018*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

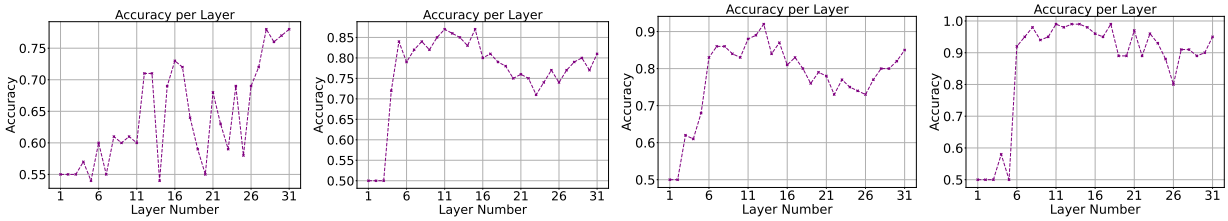
Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. Poisonedrag: Knowledge poisoning attacks to retrieval-augmented generation of large language models. *USENIX Security 2025*.

A Appendix

A.1 Ablation Studies

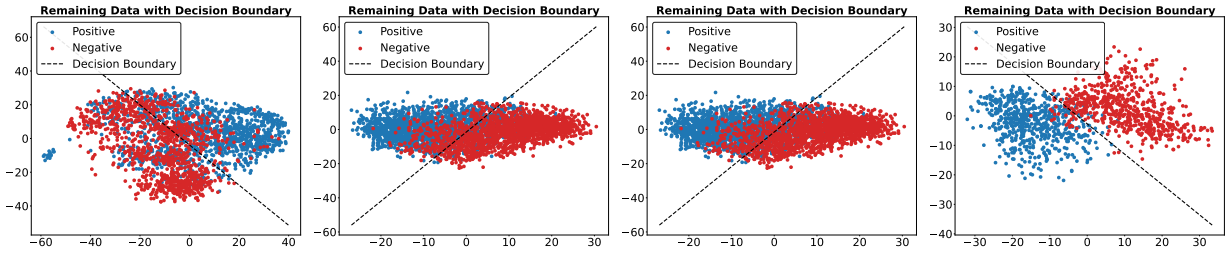
A.1.1 Using Other Layers' Representation

In our main section, we primarily base our analysis on the representations from the last layers. We also explore the knowledge checking performance using representations from other layers. Figure 5 illustrates the 'rep-con' performance of each layer across four different tasks. We observe that the performance using the first few layers is poor for all tasks. This may be because these layers primarily capture low-level features and patterns in the input, rather than higher-level semantic concepts. They haven't yet integrated this information into more abstract or task-relevant representations, which are necessary for complex knowledge checking tasks. For internal knowledge checking tasks, using representations from the last few layers shows the best performance. However, for other tasks, representations from some middle layers perform better than those from the last layer. This may be because these middle layers are more responsible for processing corresponding concepts. In practice, we suggest using a validation set to identify the layers with the best performance and using the results from these layers for knowledge checking.



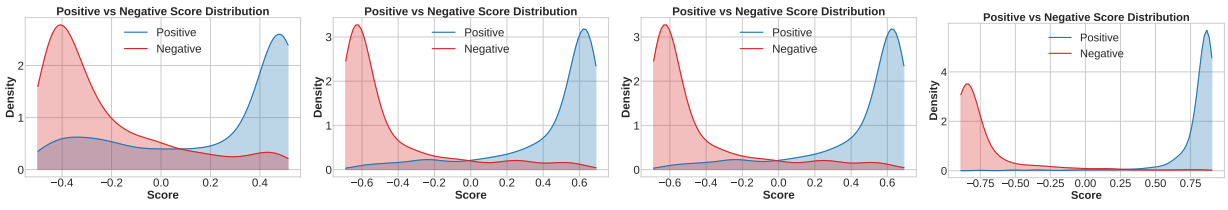
(a) Internal Knowledge (b) Uninformed helpfulness (c) Informed helpfulness (d) Contradiction

Figure 5: Accuracy on different layers



(a) Internal Knowledge (b) Uninformed helpfulness (c) Informed helpfulness (d) Contradiction

Figure 6: Visualization on PCA space(Llama-2-7B-Chat)



(a) Internal Knowledge (b) Uninformed helpfulness (c) Informed helpfulness (d) Contradiction

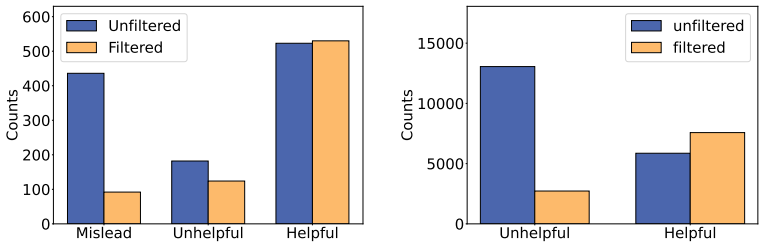
Figure 7: Visualization of contrastive scores(Llama-2-7B-Chat)

A.1.2 Knowledge Checking Performance of Other model

In this section, we also visulize and report the representation knowledge checking performance of Llama2-7B-Chat model. From the results of Table 3 and visualization in Figure 6 and Figure 7, we can get similar

Table 3: Representation checking performance of Llama-2-7B

	Internal Knowledge				Uninformed Helpfulness				Informed Helpfulness				Conflict Detection			
Method	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
Re-PCA	0.75	0.80	0.73	0.76	0.79	0.77	0.81	0.79	0.83	0.84	0.82	0.83	0.96	0.96	0.96	0.96
Re-Contra	0.76	0.83	0.72	0.78	0.81	0.80	0.82	0.81	0.89	0.89	0.89	0.89	0.97	0.96	0.99	0.97



(a) Noisy queries (b) Clean queries

Figure 8: Filter results

observation as Mistral-7B, the performance of representation-based checking is also promising for 4 tasks. Indicating the generalizbility of representation knowlege checking across models.

A.1.3 Filtering Results on PopQA

We also present the filtering results for both noisy and clean queries from the PopQA dataset in Figure 8. We can also clearly observe that the mislead and unhelpful documents are reduced while helpful documents increased.

A.1.4 Minimal training data requirement

While our method requires data to anchor the PCA space and train the classifier, only a very small amount is needed to achieve good knowledge checking performance. As shown in Table 4 below, even when the training data sample size is reduced to 50 positive-negative pairs, the knowledge checking performance remains competitive. In practice, although obtaining large-scale data can be challenging, collecting and labeling a small set of task-specific training data (50-100 pairs) is feasible and affordable. This minimal training data requirement makes our method more practical.

Table 4: Knowledge checking performance with different training samples

Method	Internal Knowledge	Uninformed Helpfulness	Informed Helpfulness	Conflict Detection
Rep-PCA(100)	0.75	0.79	0.81	0.91
Rep-Con(100)	0.78	0.81	0.85	0.95
Rep-PCA(70)	0.73	0.76	0.80	0.89
Rep-Con(70)	0.76	0.78	0.82	0.91
Rep-PCA(50)	0.71	0.76	0.79	0.87
Rep-Con(50)	0.72	0.76	0.78	0.87

A.1.5 O.O.D Results

We conducted out-of-distribution (O.O.D) experiments for our representation-based methods on the contradictory checking task, as shown in Table 5 (check whether the context is contradictory to the LLM's own knowledge). For the first experiment, we trained the Rep-PCA and Rep-Con classifiers on the original ConflictQA (Xie et al.) dataset but tested them on COUNTERFACT (Meng et al., 2022), a different dataset. Specifically, we selected a subset of 1,000 samples from COUNTERFACT (Meng et al., 2022) that Mistral-7B can answer (has internal knowledge). Although our methods' performance was lower than when directly trained on COUNTERFACT (100 samples), denoted as Rep-PCA(i.i.d) and Rep-Con(i.i.d), it still significantly outperformed other baselines.

Table 5: O.O.D knowledge checking results

Method	Conflict–COUNTERFACT	Occupation–City
Answer(Direct)	0.52	0.50
Answer(ICL)	0.47	0.43
Answer(COT)	0.49	0.44
Prob(lowest)	0.53	0.55
Prob(avg)	0.51	0.53
Perplexity	0.51	0.51
Rep-PCA(i.i.d)	0.92	0.97
Rep-Con(i.i.d)	0.95	0.98
Rep-PCA(o.o.d)	0.79	0.83
Rep-Con(o.o.d)	0.81	0.85

In our second O.O.D experiment, we deliberately sampled two types of questions from ConflictQA (Xie et al.): "Occupation" questions about people's professions and "City" questions about urban areas. We trained the classifier only on Occupation-type questions (100 training samples) and tested it on City-type questions. Our methods, even in O.O.D settings, achieved approximately 85% accuracy. Although this is lower than when directly trained on "City" questions (100 training samples), denoted as Rep-PCA(i.i.d) and Rep-Con(i.i.d), it still substantially outperforms the baselines.

These results demonstrate that our representation-based method effectively captures intrinsic differences between positive and negative samples and shows reasonable generalizability, enabling it to work even without in-distribution training data.

A.2 Details of knowledge checking methods

A.2.1 Prompts for representation-based methods

For representation-based methods, we employ prompts as illustrated in Table 6 to generate positive and negative samples, allowing us to capture the representation behaviors. After obtaining the representations of the final tokens, we conduct analysis based on these representations, following the methodology detailed in Section 3.1.

A.2.2 Details of answer-based methods

Prompts used. We present the prompt template of various answer-based checking, including direct prompting, ICL prompting as well as CoT prompting in this section. Table 7 shows the templates for internal knowledge checking, Table 8 shows the templates for informed/uninformed helpfulness checking, while Table 9 shows the templates for contradictory checking.

Self-RAG implementation. Self-Reflective Retrieval-Augmented Generation (SELF-RAG) (Asai et al.) is proposed to enhance the quality and factuality of LLM. The LLM is fine-tuned to generate special tokens that indicate whether to retrieve and whether the retrieved context is relevant. The Self-Rag-Llama [10] and Self-Rag-Mistral [11] we used in this paper is fine-tuned from Llama2-7B and Mistral-7B-v0.1 respectively, using the same dataset.

We use the 'input question only' format from Table 10 to generate the 'retrieve-on-demand' special token.If the 'Retrieval' token is generated, the LLM will retrieve the top-k context, while the 'No retrieval' token will not retrieve any context. After retrieving the context, we constructed prompts using the 'input question and context' row template from Table 10. The 'Relevant' token indicates that the retrieved context is helpful for the question. Similarly, the 'Irrelevant' token indicates that the retrieved context is not useful for the question. To verify the overall performance of Self-RAG, we first use the fine-tuned model to judge whether the context is relevant or irrelevant. Then, we filter out the irrelevant contexts and select the top two retrieved contexts. Based on the inference row in Table 10, we construct prompts to test the output of different models, and finally compare whether the outputs include the correct answer.

[10] https://huggingface.co/selfrag/selfrag_llama2_7b

[11] https://huggingface.co/SciPhi/SciPhi-Self-RAG-Mistral-7B-32k

A.2.3 Details for probability-based methods

For probability-based methods, we use the same input as shown in Table 6, but we analyze the probabilities of output tokens. We primarily consider three indicators that have been used in previous research: perplexity(Zou et al., 2024), average probability score of all output tokens(Jiang et al., 2023b), and the lowest probability score of output tokens(Jiang et al., 2023b). For perplexity, we classify samples with higher perplexity (indicating less confidence) than a threshold as negative, while others are classified as positive. For both the lowest and average probability scores, we consider samples with lower scores (again indicating less confidence) than a threshold as negative, while others are classified as positive. For each method, we vary the threshold and report the best accuracy. Additionally, we plot Receiver Operating Characteristic (ROC) curves and calculate the Area Under the Curve (AUC), as shown in Figure 1.

A.3 Dataset Used

In this section, we would like to introduce the dataset used for knowledge checking and for context filtering in detail.

A.3.1 Knowledge checking.

Internal knowledge checking. For internal knowledge checking, utilize the RetrievalQA dataset (Zhang et al., 2024), a short-form open-domain question answering (QA) collection comprising 2,785 questions. This dataset includes 1,271 new world and long-tail questions that most LLMs cannot answer, serving as negative samples (queries without internal knowledge). These samples are collected and filtered from RealTimeQA, FreshQA, ToolQA, PopQA and TriviaQA. Additionally, it contains 1,514 questions that most LLMs can answer using only their internal parametric knowledge, functioning as positive samples (queries with internal knowledge).

Helpfulness Checking. We utilize a subset of the Natural Questions (NQ) dataset employed by Cuconasu et al. (2024b). [12]. The authors provide a labeled set of 83,104 NQ queries, each associated with a golden passage that directly answers the question, as well as distract passages retrieved from wikitext-2018 that do not contain the answer. For our helpfulness checking task, we use a subset of 10,000 queries also provided in their repository. We use the distract passage with the highest retrieval score as the negative sample and the golden passage as the positive sample. For the uninformed helpfulness checking, we focus on questions that Mistral-7B cannot correctly answer, resulting in a total of 8,081 queries. For the informed helpfulness checking evaluation, we select the remaining 1,919 queries that Mistral-7B can correctly answer, ensuring the model has internal knowledge about these queries.

Contradictory Checking. For contradictory checking, we use a subset of ConflictQA constructed by Xie et al.. Each sample in ConflictQA dataset contains a question from PopQA, an aligned evidence that can correctly answer the question, as well as a contradictory evidence that provides wrong evidence towards the query generated by ChatGPT. We sampled a subset of 1146 questions from the ConflictQA dataset that Mistral-7B can correctly answer, and use the aligned evidence(item["parametric_memory_aligned_evidence"]) with the query as positive samples as well as contradictory evidence(item["counter_memory"]) with the query as negative samples.

A.3.2 Context filtering.

We utilize two primary datasets: a subset of Natural Questions (NQ) used by Cuconasu et al. (2024a) and ConflictQA, a subset of PopQA employed by (Xie et al.). Cuconasu et al. (2024a) treats the long answers in the NQ dataset as gold documents and the short answers as ground truth. They filtered the NQ dataset to discard documents exceeding 512 tokens after Llama2 tokenization. And we used GPT-3.5-turbo to generate mislead text based on the gold text for each query. We utilize the "Get wrong answer" row in Table 11 to generate a misleading answer, and then generate the misleading text using the format specified in the "Generate mislead text" row. To ensure the quality of the generated results, we validated the generated text. The requirements are that the wrong answer must appear in the text, and none of the

[12]Available on https://github.com/florin-git/The-Power-of-Noise?tab=readme-ov-file

true answers should be present in the text. If these conditions are not met, the text will be regenerated until they are satisfied.

Xie et al. selected a subset from popQA. In this selected subset, for each question, the answers provided by the LLM based on its own parameter knowledge and those retrieved context are contradictory. For each pair of contradictory answers, they generated supporting text as evidence for each answer. We utilized the all the subsets across different models and ensured that the questions were not duplicated. We verified whether the parameter knowledge or the external knowledge was correct and labeled the correct evidence text as gold context, while marking the incorrect text as misleading context. Finally, we obtain the dataset containing 11,216 queries with labeled golden passages and misleading contexts.

Table 6: Context and Question Scenarios

Task 1: Internal Knowledge Checking
Question: {<Question with Internal Knowledge> or <Question without Internal Knowledge>}
Answer:
Task 2 & 3: Helpfulness Checking
Context: {<Helpful Context> or <Unhelpful Context>}
Question: {question}
Answer:
Task 4: Contradiction Checking
Context: {<Aligned Context> or <Contradictory Context>}
Question: {question}
Answer:

Table 7: Internal Knowledge Checking Prompts

Name	Prompt
Direct	Are you sure you can accurately answer the following question based on your internal knowledge? If yes, you should answer "Yes" and give your answer. If no, you should answer "No, I need additional information to answer this question." Question: {question} Answer:
ICL	Determine if you can accurately answer the following question based on your internal knowledge. If you can, answer "Yes" and provide your answer. If you cannot, answer "No, I need additional information to answer this question." Question: Cryos, the world's largest sperm bank, recently announced that they will no longer accept donations from guys with what physical characteristic? Answer: No, I need additional information to answer this question. Question: What is the capital of France? Answer: Yes, I can answer this question. The capital of France is Paris. Can you answer the below question based on your internal knowledge? Question: {question} Answer:
CoT	Think step by step to determine if you can accurately answer the following question based on your internal knowledge. If you can, answer "Yes" and provide your answer. If you cannot, answer "No, I need additional information to answer this question." Question: {question} Answer:

Table 8: Context Helpfulness Checking Prompts

Name	Prompt
Direct	Does the provided context: {context} helpful to answer the question: {question}? Please answer yes if it is helpful and no if it is unhelpful. Answer:
ICL	I will provide you with some examples of how to determine if a given context is helpful to answer a specific question. Then, I will ask you to do the same for a new question and context. Example 1: Question: What is the capital of France? Context: Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents as of 2018. Answer: Yes. This context is helpful Example 2: Question: How does photosynthesis work? Context: The Eiffel Tower in Paris was completed in 1889 and stands at 324 meters tall. Answer: No. This context is not helpful Example 3: Question: what is the name of latest version of android Context: to Google adopting it as an official icon as part of the Android logo when it launched to consumers in 2008. Android (operating system) Android is a mobile operating system developed by Google. It is based on a modified version of the Linux kernel and other open source software, and is designed primarily for touchscreen mobile devices such as smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs Answer: No. This context is not helpful Now, please determine if the following context is helpful to answer the given question. Answer "Yes" if it is helpful, or "No" if it is unhelpful. Question: {question} Context: {context} Answer:
CoT	Think step by step to determine if the provided context is helpful to answer the given question. After your analysis, conclude with "Yes" if the context is helpful, or "No" if it is unhelpful. Question: {question} Context: {context} Answer:

Table 9: Internal Belief Alignment Checking Prompts

Name	Prompt
Direct	Based on your internal knowledge, do you think the provided context is aligned to your internal belief? If aligned, you should answer "Yes". If contradictory, you should answer "No". Context: {context} Answer:
ICL	I will provide you with some examples of how to determine if a given context aligns with internal knowledge. Then, I will ask you to do the same for a new context. Example 1: Context: The Earth is flat and sits on the back of a giant turtle. Answer: No. This context contradicts well-established scientific knowledge that the Earth is approximately spherical and orbits the sun. Example 2: Context: Water is composed of hydrogen and oxygen atoms. Answer: Yes. This context aligns with the scientific understanding of water's molecular composition. Example 3: Context: Gravity causes objects with mass to attract each other. Answer: Yes. This context is consistent with the fundamental principles of physics and gravity. Now, based on your internal knowledge, determine if the following context is aligned with your internal belief. If aligned, answer "Yes". If contradictory, answer "No". Context: {context} Answer:
CoT	Based on your internal knowledge, think step by step to determine if the provided context is aligned with your internal belief. After your analysis, conclude with "Yes" if the context is aligned, or "No" if it is contradictory. Context: {context} Answer:

Table 10: Prompts of Self-RAG

Mode	Prompt
Input question only	### Instruction: {input question} ### Response:
Input question and context	### Instruction: {input question} ### Response: [Retrieval]<paragraph>{input context}</paragraph>
Inference	Context 1: {first relevant context} Context 2: {second relevant context} Question:{input question} Answer:

Table 11: Prompts of getting mislead context

Mode	Prompt
Get wrong answer	You are a helpful assistant that provides a wrong answer consists of a few words Give me a wrong answer of the '{question}?' with similar type but different to any of {true answers}. ONLY RETURN the wrong answer, nothing else. The answer should be less than 4 words, DO NOT return a sentence.
Generate mislead text	You are a helpful assistant that generates short descriptions with specific evidence in JSON format. Generate a 100-word paraphrased version for '{question}? {wrong answer}' as if it is absolutely correct. Ensure the exact word '{wrong answer}' appears in your paraphrased version. You can not find any of {true answers} in the paraphrased version. Return your response in the following JSON format, any of {true answers} should never appears in the following context: {{ "context": "Your 100-word paraphrased version containing '{wrong answer}'." }} Ensure the JSON is valid.