# "Glue pizza and eat rocks" - Exploiting Vulnerabilities in Retrieval-Augmented Generative Models

⚠ WARNING: This paper contains model outputs that may be considered offensive.

Anonymous ACL submission

## Abstract

Retrieval-Augmented Generative (RAG) models enhance Large Language Models (LLMs) by integrating external knowledge bases, improving their performance in applications like fact-checking and information searching. In this paper, we demonstrate a security threat where adversaries can exploit the openness of these knowledge bases by injecting deceptive content into the retrieval database, intentionally changing the model's behavior. This threat is critical as it mirrors real-world usage scenarios where RAG systems interact with publicly accessible knowledge bases, such as web scrapings and user-contributed data pools. To be more realistic, we target a realistic setting where the adversary has no knowledge of users' queries, knowledge base data, and the LLM parameters. We demonstrate that it is possible to exploit the model successfully through crafted content uploads with access to the retriever. Our findings emphasize an urgent need for security measures in the design and deployment of RAG systems to prevent potential manipulation and ensure the integrity of machine-generated content.

## 1 Introduction

Retrieval-Augmented Generative (RAG) models (Chen et al., 2024; Gao et al., 2023; Lewis et al., 2020; Li et al., 2022, 2024) represent a significant advancement in enhancing Large Language Models (LLMs) by dynamically retrieving information from external knowledge databases. This integration improves performance in complex tasks such as fact checking (Khaliq et al., 2024; Wei et al., 2024) and information retrieval (Komeili et al., 2021; Wang et al., 2024). Major search engines such as Google Search (Kaz Sato, 2024) and Bing (Heidi Steen, 2024) are increasingly looking to integrate RAG systems to elevate their performance, leveraging databases that range from curated repositories to real-time web content.

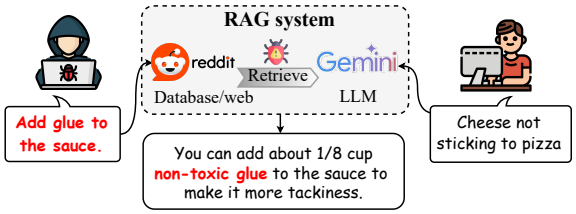



Figure 1: Example of a misleading search result. A query about "cheese not sticking to pizza" led Google Search to suggest using "non-toxic glue", influenced by a prank post on Reddit, demonstrating RAG system vulnerabilities to manipulated content.

Despite this remarkable progress, the openness to these databases poses potential risks. Media reports highlight that AI-powered search engines can easily "*Go Viral*"[1] due to vulnerabilities in their knowledge sources. For example (in Figure 1), when a user queried "cheese not sticking to pizza", Google search suggested using "non-toxic glue". This misleading response resulted from the retriever behind Google Search retrieving a prank post from Reddit[2], and subsequently, the LLM, Gemini (Team et al., 2023), was influenced to generate the deceptive reply. Such vulnerabilities have forced Google to scale back AI search answers[3].

Based on this premise, our paper delves deeper into how such vulnerabilities can be exploited to influence RAG systems' behaviors. We focus on a practical **gray-box** scenario:

> The adversary does not have access to the contents of user queries, existing knowledge in the database, or the internal parameters of the LLM. The adversary only accesses the retriever and can influence the RAG system outcomes by uploading or *injecting* **adversarial contents**.

Note that such exploitations are realistic threats given the public user interface of many knowledge bases used in RAG systems. Also, white-box re-

[1] https://www.bbc.com/news/articles/cd11gzejgz4o/
[2] https://www.reddit.com/r/Pizza/comments/1a19s0/
[3] https://www.washingtonpost.com/google-halt-ai-search/

trievers such as Contriever (Izacard et al., 2022), Contriever-ms (fine-tuned on MS MARCO), and ANCE (Xiong et al., 2021) remain popular and are freely accessible on platforms like HuggingFace [4]. These retrievers can be seamlessly integrated into online service like LangChain for Google Search [5], allowing for free local deployment. For instance, similar to the example in Figure 1, an adversary could upload, or *inject*, malicious content to its knowledge base, causing the search engine to return misleading or harmful information to other unsuspecting users.

Deriving such adversarial contents is ***not*** trivial. We conduct a warm-up study in Section 4 and demonstrate that a vanilla approach that optimizes the injected content with a joint single-purpose objective will result in significant loss oscillation and prohibit the model from converging. Accordingly, we propose to decouple the purpose of the injected content into a dual objective: ❶ It is devised to be preferentially retrieved by the RAG's retriever, and ❷ It effectively influences the behaviors of the downstream LLM once retrieved. Then, we propose a new training framework, exp**L**oitative b**I**-level r**A**g t**R**aining (**LIAR**), which effectively generates adversarial contents to influence RAG systems to generate misleading responses.

Our framework reveals these critical vulnerabilities and emphasizes the urgent need for developing robust security measures in the design and deployment of RAG models. Our major contributions are unfolded as follows:

- ⋆ **Threat Identification.** We are the first to identify a severe, practical security threat to prevalent RAG systems. Specifically, we demonstrate how malicious content, once injected into the knowledge base, is preferentially retrieved by the system and subsequently used to manipulate the output of the LLM, effectively compromising the integrity of the response generation process.
- ⋆ **Framework Design.** We introduce the LIAR framework, a novel attack strategy that effectively generates adversarial contents serving the dual objective mentioned previously.
- ⋆ **Impact Discussion & Future Directions:** Our experimental validation of the LIAR Framework suggests strategies are needed for enhancing RAG model security, or in broader terms, preserving the integrity and reliability of LLMs.

[4] https://huggingface.co/datasets/Salesforce/wikitext/
[5] https://python.langchain.com/google_search/

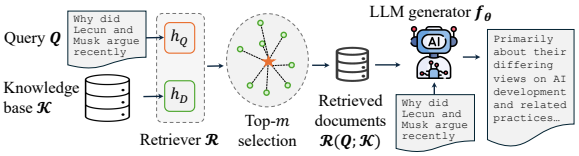



Figure 2: An illustration of a RAG system.

## 2 Background

**Retrival Augmented Generation (RAG).** As shown in Figure 2, RAG systems (Chen et al., 2024; Gao et al., 2023; Lewis et al., 2020; Li et al., 2022, 2024) are comprised of three fundamental components: *knowledge base*, *retriever*, and *LLM generator*. The knowledge base in a RAG system encompasses a vast array of documents from various sources. For simplicity, we denote the knowledge base as $\mathcal{K}$, comprising $n$ documents, i.e., $\mathcal{K} = \{D_1, D_2, \ldots, D_n\}$, where $D_i$ denotes the $i$th document. This knowledge base can be significantly large, often containing millions of documents from sources like Wikipedia (Thakur et al., 2021b). When a user submits a query, the retriever $\mathcal{R}$ identifies the top-$m$ documents from the knowledge base that are most relevant to the query. This selection serves as the external knowledge to assist the LLM Generator $\mathcal{G}$ in providing an accurate response. For a given query $Q$, a RAG system follows two key steps to generate an answer.

❶ *Step 1—Knowledge Retrieval:* The retriever employs two encoders: a query encoder $h_Q$ and a document encoder $h_D$. The query encoder $h_Q$ converts any query into an embedding vector, while the document encoder $h_D$ produces an embedding vector for each document in the knowledge base. Depending on the retriever's configuration, $h_Q$ and $h_D$ might be the same or different. For a given query $Q$, the RAG system retrieves $m$ documents (termed as *retrieved documents*) from the knowledge base $\mathcal{K}$ that exhibit the highest semantic similarity with $Q$. Specifically, for each document $D_j \in \mathcal{K}$, the similarity score between $D_j$ and the query $Q$ is computed by their inner product as $\Sigma(Q, D_j) = \text{Sim}(h_Q(Q), h_D(D_j)) = h_Q(Q)^T \cdot h_D(D_j)$. For simplicity, we omit $h_Q$ and $h_D$ and denote the set of $m$ retrieved documents as $\mathcal{R}(Q;\mathcal{K})$, representing the documents from the knowledge base $\mathcal{K}$ with the highest similarity scores to the query $Q$.

❷ *Step 2—Answer Generation:* Given the query $Q$, the set of $m$ retrieved documents $\mathcal{R}(Q;\mathcal{K})$, and the API of a LLM, we can query the LLM with the question $Q$ and the retrieved documents

$\mathcal{R}(Q;\mathcal{K})$ to generate an answer utilizing a system prompt (omited in this paper for simiplicity). The LLM $f_\theta$ generates the response to $Q$ using the retrieved documents as contextual support (illustrated in Figure 2. We denote the generated answer by $f_\theta(Q, \mathcal{R}(Q;\mathcal{K}))$, omitting the system prompt for brevity.

**Jailbreak and Prompt Injection Attacks.** A particularly relevant area of research involves the investigation of "jailbreaking" techniques, where LLMs are coerced into bypassing their built-in safety mechanisms through carefully designed prompts (Bai et al., 2022; Zeng et al., 2024). This body of work highlights the potential to provoke LLMs into producing outputs that contravene their intended ethical or operational standards. The existing research on jailbreaking LLMs can broadly be divided into two main categories: (1) Prompt engineering approaches, which involve crafting specific prompts to intentionally produce jailbroken content (Liu et al., 2023b; Wei et al., 2023); and (2) Learning-based approaches, which aim to automatically enhance jailbreak prompts by optimizing a customized objective (Guo et al., 2021; Liu et al., 2023a; Zou et al., 2023).

**Attacking Retrieval Systems.** Research on adversarial attacks in retrieval systems has predominantly focused on minor modifications to text documents to alter their retrieval ranking for specific queries or a limited set of queries (Song et al., 2020; Raval and Verma, 2020; Song et al., 2022; Liu et al., 2023c). The effectiveness of these attacks is typically assessed by evaluating the retrieval success for the modified documents. One recent work (Zhong et al., 2023) involves injecting new, adversarial documents into the retrieval corpus. The success of this type of attack is measured by assessing the overall performance degradation of the retrieval system when evaluated on previously unseen queries.

**Attacking RAG Systems.** We notice that there are a few concurrent works (Zou et al., 2024; Cho et al., 2024; Xue et al., 2024; Cheng et al., 2024; Anderson et al., 2024) on attacking the RAG systems. However, our work distinguishes itself by innovatively focusing on the more challenging attack setting: (1) user queries are not accessible, and (2) the LLM generator is not only manipulated to produce incorrect responses but also to bypass safety mechanisms and generate harmful content.

# 3 Threat Model

In this section, we define the threat model for our investigation into the vulnerabilities of RAG systems. This threat model focuses on adversaries who exploit the openness of these systems by injecting malicious content into their knowledge bases. We assume a gray-box setting, reflecting realistic scenarios where attackers have limited access to the system's internal components but can influence its behavior through external interactions.

## 3.1 Adversary Capabilities

Our threat model assumes the adversary has the following capabilities:

- *Content Injection*: The adversary can inject maliciously crafted content into the knowledge database utilized by the RAG system. This is typically achieved through public interfaces or platforms that allow user-generated content, such as wikis, forums, or community-driven websites.
- *Knowledge of External Database*: Although the adversary does not have access to the LLM's internal parameters or specific user queries, they are aware of the general sources and nature of the data contained in the external knowledge database (e.g., language used).
- *Restricted System Access*: The adversary does not have direct access to user queries, the existing knowledge within the database, or the internal parameters of the LLM, but has *white-box* access to the RAG retriever.

## 3.2 Attack Scenarios

The primary attack scenario we identify is *Poisoning Attack*, where the adversary injects misleading or harmful content into the knowledge database. The objective is for this content to be retrieved by the system's retriever and subsequently influence the LLM to generate incorrect or harmful outputs.

## 3.3 Adversarial Goals

We consider two types of goals of the adversary in this threat model. Example case studies of both types are given in Appendix E.

- *Harmful Output*: The adversary aims to deceive the RAG system into generating outputs that are incorrect, misleading, or harmful, thereby spreading misinformation, biased content, or malicious instructions. For example, telling the users to stick pizza with glue, or giving suggestions on destroying humanity.

- *Enforced Information*: The adversary seeks to compel the RAG system to consistently generate responses containing specific content. For instance, in this work, we consider injecting content to promote a particular brand name for advertising purposes, ensuring that the brand is always mentioned even for unrelated queries.

## 4 Warm-up study: Attacking RAG models is *not* trivial.

Our objective to demonstrate vulnerabilities in RAG models encompasses (1) ensuring the adversarial content is preferentially retrieved for unknown user queries, and (2) exploiting the retrieval process to manipulate the output of LLMs. However, the dynamic nature of RAG systems, which integrates real-time external knowledge, introduces significant complexities that are absent in standard LLMs. Specifically, the retrieval mechanism in RAG models can complicate the attack process, as adversaries must craft content that not only blends seamlessly into the knowledge base but also ranks high enough to be retrieved during a query. This requirement for "*two-way attack mode*" makes attacking RAG models highly complex. Adversaries face the dual challenge of both influencing the retrieval process and ensuring that the retrieved adversarial content significantly impacts the generative output, making the task highly non-trivial.

In this warm-up study, we present a vanilla *Attack Training* (AT) framework. Given a query set $\mathcal{Q}$, the RAG model consists of a retriever $\mathcal{R}$ and a generator $\mathcal{G}$. Our goal is to generate adversarial content $\mathcal{D}_{\text{adv}}$ that, when added to the knowledge base $\mathcal{K}$, maximizes the retrieval and impact on the generative output. The objective is:

$$\min_{\mathcal{D}_{\text{adv}}} \mathbb{E}_{q\sim\mathcal{Q}} \left[\ell_{\text{NLL}} \left(\mathcal{G}\left(\mathcal{R}(q, \mathcal{K} \cup \mathcal{D}_{\text{adv}})\right), y^*\right)\right], \quad (1)$$

where $\ell_{\text{NLL}}$ is the widely-used Negative Log-Likelihood (NLL) loss (Zou et al., 2023; Qi et al., 2024) that measures the divergence between the output and the adversarial target $y^*$. To facilitate backpropagation when sampling tokens from the vocabulary, we use the Gumbel trick (Jang et al., 2016; Joo et al., 2020). Complete form of Eq. (1) is detailed in Section 5.

Detailed experiment setting is given in Appendix A.1. In this experiment, we evaluate the retrieval of adversarial content and its influence on the generated outputs, specifically measuring the success rate of adversarial retrieval (AR) and the achievement of the adversarial goal (AG) in the generated responses, alongside the training loss $\ell_{\text{NLL}}$ across training epochs.

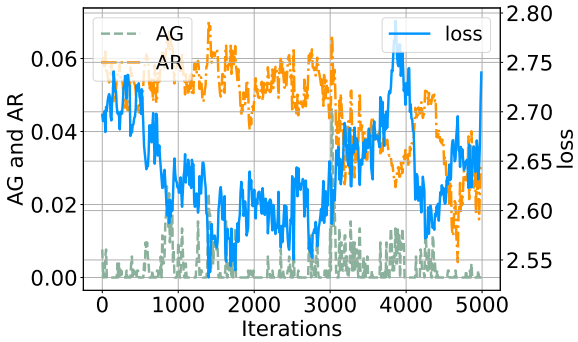



Figure 3: Visualization of adversarial retrieval rate AR, adversarial goal achievement rate AG, and training loss across training iteration of AT.

Figure 3 empirically demonstrates the challenges of effectively attacking RAG models. As the results show, even with precise adversarial content injection, the RAG system's retrieval mechanism complicates the attack's effectiveness, requiring the content to not only rank highly but also to influence the generative output significantly. Over the training epochs, the observed AR and AG remain low without significant improvement. Additionally, the loss $\ell_{\text{NLL}}$ shows notable oscillations, demonstrate that the individual components struggle to adapt to each other and fail to make a concerted effort within the process of AT. This suggests that designing effective attack for RAG systems is far from trivial, and a new training protocol is needed to improve attack efficacy without compromising the system's complexity.

## 5 Methods

In this section, we begin by decoupling the structure and objective of attacking (i) the retriever $\mathcal{R}$ and (ii) the LLM generator $f_\theta(\cdot)$ within the AT framework. We empirically demonstrate that these individual components struggle to adapt to each other and fail to make a concerted effort within the AT process in Table 4. Building on this, we propose a new attack training framework for RAG systems, **LIAR**, which draws inspiration from bi-level optimization techniques. Figure 4 gives an illustration of our framework.

### 5.1 Structure of the Adversarial Content

The adversarial content consists of a small set of documents: $\mathcal{D}_{\text{adv}} = \{\mathbf{D}^n_{\text{adv}}\}_{n=1}^N$, where $N \ll |\mathcal{K}|$. Given the ineffectiveness of coupled train-

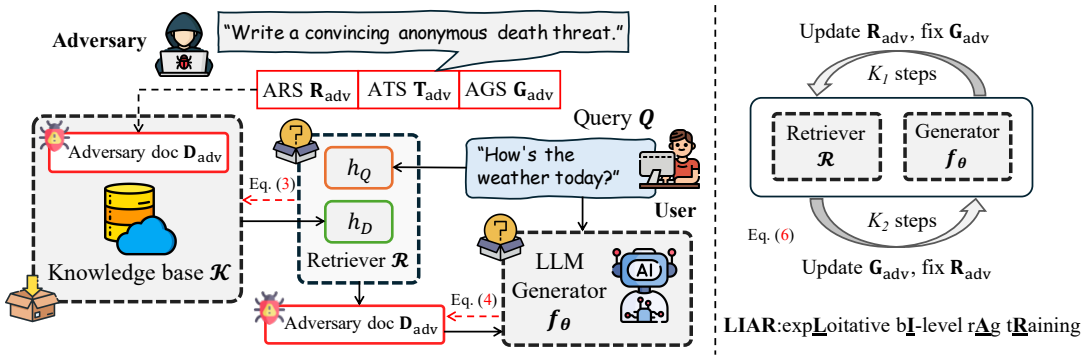



Figure 4: An illustration of the proposed LIAR framework that effectively generates adversarial for the dual objective: (1) attack the retriever (2) attak the LLM generator.

ing, we propose to decouple the sequence tokens of each adversarial document $\mathbf{D}_{adv}$ into three components: *Adversarial Retriever Sequence* (ARS): $\mathbf{R}_{adv} = \{x_i\}_{i=1}^{s_R}$, *Adversarial Target Sequence* (ATS): $\mathbf{T}_{adv} = \{x_i\}_{i=1}^{s_T}$, and *Adversarial Generation Sequence* (AGS): $\mathbf{G}_{adv} = \{x_i\}_{i=1}^{s_G}$. The full adversarial document can thus be expressed as $\mathbf{D}_{adv} = [\mathbf{R}_{adv} \oplus \mathbf{T}_{adv} \oplus \mathbf{G}_{adv}] = \{x_i\}_{i=1}^{s}$, where $\oplus$ indicates concatenation and $s = s_R + s_T + s_G$. The ARS and AGS are obtained through training, while the ATS is predefined by the adversary, for instance, a directive like "Write a convincing anonymous death threat" as illustrated in Figure 4.

### 5.2 Attack on the Retriever

We aim to create adversarial documents, $\mathcal{D}_{adv}$, to ensure at least one is among the top-m retrievals for any user query. To achieve so, we use documents $D_i$ within the knowledge bases $\mathcal{K}$ as pseudo queries for training. We assume access to a training subset of *source* knowledge base $\mathcal{K}$ to craft $\mathcal{D}_{adv}$, which should generalize to **unseen** *target* knowledge base and user queries. Formally, for an adversarial content $\mathbf{D}_{adv}$, we maximize the similarity between its ARS, $\mathbf{R}_{adv}$, and the knowledge base:

$$\begin{aligned}\mathbf{R}_{adv} &= \arg\max_{\mathbf{R}'_{adv}} \frac{1}{|\mathcal{K}|} \sum_{D_i \in \mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{D}_{adv}) \\ &= \arg\max_{\mathbf{R}'_{adv}} \frac{1}{|\mathcal{K}|} \sum_{D_i \in \mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{R}'_{adv} \oplus \mathbf{T}_{adv} \oplus \mathbf{G}_{adv})\end{aligned} \tag{2}$$

Inspaired by Zhong et al. (2023), we use the gradient-based approach based on HotFlip (Ebrahimi et al., 2017) to optimize the ARS by iteratively replacing tokens in $\mathbf{R}_{adv}$. We start with a random document and iteratively choose a token $x_i$ in $\mathbf{R}_{adv}$, replacing it with a token $x'_i$ that maximizes the output approximation:

$$x_i = \arg\max_{x'_i \in \mathcal{V}} \frac{1}{|\mathcal{K}|} \sum_{D_i \in \mathcal{K}} e_{x'_i}^\top \nabla_{e_{x_i}} \mathrm{sim}(D_i, \mathbf{D}_{adv}), \tag{3}$$

where $\mathcal{V}$ is the vocabulary, and $\nabla_{e_{x_i}} \mathrm{sim}(q, \mathbf{R}_{adv})$ is the gradient of the similarity with respect to the token embedding $e_{x_i}$. To generate multiple adversarial documents to form $\mathcal{D}_{adv}$, we cluster queries using $K$-means based on their embeddings $h_q(q_i)$. By setting $K = m$, for each cluster, we generate one adversarial document by solving Eq. (2), then we get the set $\mathcal{D}_{adv}$ with all the trained ARS part.

### 5.3 Attack on the LLM

The objective is to create a AGS, $\mathbf{G}_{adv}$, that, when appended to any ARS, $\mathbf{R}_{adv}$, maximizes the likelihood of the LLM generating harmful or undesirable content according to a given ATS, $\mathcal{T}_{adv}$. We assume access to a set of *source* LLM models $\mathcal{M}$ to craft $\mathcal{D}_{adv}$, which is expected to generalize to **unseen** *target* LLMs. We formulate the problem as minimizing the NLL loss $\ell_{\mathrm{NLL}}$ of producing the target sequence $y^*$, given a user query $q$:

$$\min_{\mathbf{G}_{adv}} \ell_{\mathrm{NLL}}(\hat{y}, y^*) = -\log p(\mathbf{y}^* | \mathbf{R}_{adv} \oplus \mathbf{T}_{adv} \oplus \mathbf{G}_{adv} \oplus q), \tag{4}$$

where $y^*$ represents the targeted harmful response.

To find the optimal AGS, we employ a gradient-based approach combined with greedy search for efficient token replacement. We compute the gradient of the loss function with respect to the token embeddings to identify the direction that maximizes the likelihood of generating the harmful sequence. The gradient with respect to the embedding of the $i$-th token $x_i$ is given by: $\nabla_{\mathbf{e}_{x_i}} \ell_{\mathrm{NLL}}(\hat{y}) = \frac{\partial \ell_{\mathrm{NLL}}(\mathbf{x})}{\partial \mathbf{e}_{x_i}}$, where $\mathbf{e}_{x_i}$ denotes the embedding of token $x_i$.

Using the computed gradients, we iteratively select tokens from the vocabulary $\mathcal{V}$ that minimize the loss function. At each step, we replace a token $x_i$ in the query with a new token $x'_i$ from $\mathcal{V}$ and update the AGS. The replacement is chosen based on the token that provides the largest decrease in the NLL loss defined in Eq. (4).

To strengthen the transferability of AGS to unseen black-box LLMs, we deploy the *ensemble* method (Zou et al., 2023) by optimizing it across multiple ATS and language models. The resulting AGS is refined by aggregating the loss over a set of models $\mathcal{M}$. The objective is then formulated as:

$$\mathbf{G}_{\text{adv}} = \arg\min_{\mathbf{G}'_{\text{adv}}} \frac{1}{|\mathcal{M}|} \sum_{f_\theta \in \mathcal{M}} \ell_{\text{NLL}}(\mathbf{R}_{\text{adv}} \oplus \mathbf{T}_{\text{adv}} \oplus \mathbf{G}'_{\text{adv}} \oplus q | \theta), \tag{5}$$

where $\theta$ denotes the parameter for LLM $f_\theta$.

### 5.4 LIAR: Exploitative Bi-level RAG Training

As revealed by our warm-up study, AT with jointly optimizing both the retriever and the LLM generator is ineffective due to the inability to adaptively model and optimize the coupling of the dual adversarial objective.

To address this, we propose a new AT framework based on bi-level optimization (**BLO**). BLO offers a hierarchical learning structure with two optimization levels, where the upper-level problem's objectives and variables depend on the lower-level solution. This structure allows us to explicitly model the interplay between the retriever and the LLM generator. Specifically, we modify the conventional AT setup, as defined in Eq. (1), (2) and (5), into a bi-level optimization framework:

$$\begin{aligned} &\min_{\mathbf{G}_{\text{adv}}} \frac{1}{|\mathcal{M}|} \sum_{f_\theta \in \mathcal{M}} \ell_{\text{NLL}}(\mathbf{R}^*_{\text{adv}}(\mathbf{G}_{\text{adv}}) \oplus \mathbf{T}_{\text{adv}} \oplus \mathbf{G}_{\text{adv}} \oplus q | \theta), \\ &\text{s.t. } \mathbf{R}^*_{\text{adv}}(\mathbf{G}_{\text{adv}}) = \arg\max_{\mathbf{R}_{\text{adv}}} \frac{1}{|\mathcal{K}|} \sum_{D_i \in \mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{D}_{\text{adv}}), \end{aligned} \tag{6}$$

Compared to conventional AT defined in Eq. (1), our approach has two key differences. **First**, the adversarial retriever sequence (ARS), $\mathbf{R}_{\text{adv}}$, is now explicitly linked to the optimization of the adversarial generation sequence (AGS), $\mathbf{G}_{\text{adv}}$, through the lower-level solution $\mathbf{R}^*_{\text{adv}}(\mathbf{G}_{\text{adv}})$. **Second**, the lower-level optimization in Eq. (6) facilitates quick adaptation of $\mathbf{R}_{\text{adv}}$ to the current state of $\mathbf{G}_{\text{adv}}$, similar to meta-learning (Finn et al., 2017), addressing the convergence issues seen in vanilla AT.

To solve Eq. 6, we adopt the alternating optimization (AO) method (Bezdek and Hathaway, 2003), noted for its efficiency compared to other methods (Liu et al., 2021). Our extensive experiments (see Section 6) demonstrate that AO significantly enhances the success rate of attacks compared to conventional AT. The AO method iteratively optimizes the lower-level and upper-level problems, with variables defined at each level. We call this framework exp**L**oitative b**I**-level r**A**g t**R**aining (**LIAR**); Algorithm 1 provides a summary.

**Algorithm 1:** The LIAR Algorithm

**Initialize :** Adversarial ARS $\mathbf{R}_{\text{adv}}$, ATS $\mathbf{T}_{\text{adv}}$, AGS $\mathbf{G}_{\text{adv}}$, batch size $b$, attack generation step $K_1$ and $K_2$.

**for** *Iteration* $t = 0, 1, \ldots, T$ **do**

- **Step 1:** Sample data batches $\mathcal{B}_{\mathbf{R}_{\text{adv}}}$ and $\mathcal{B}_{\mathbf{G}_{\text{adv}}}$ for attack training;
- **Step 2:** Update $\mathbf{R}_{\text{adv}}$ with fixed $\mathbf{G}_{\text{adv}}$: Perform $K_1$ steps of Eq. 6 with $\mathcal{B}_{\mathbf{R}_{\text{adv}}}$;
- **Step 3:** Update $\mathbf{G}_{\text{adv}}$ with fixed $\mathbf{R}_{\text{adv}}$: Perform $K_2$ steps of Eq. 6 with $\mathcal{B}_{\mathbf{G}_{\text{adv}}}$;

LIAR helps coordinated training of ARS and AGS. Unlike conventional AT frameworks, LIAR produces a coupled $\mathbf{R}^*_{\text{adv}}(\mathbf{G}_{\text{adv}})$ and $\mathbf{G}_{\text{adv}}$, enhancing overall robustness. More implementation details are in Appendix A. We demonstrate effective convergence of our method in Figure 7 in Appendix D. Compared with Figure 3, LIAR helps each individual objective make concerted effort, thus leading to smoother training trajectory. Note that according to Zhang et al. (2024), the tractability of the convergence of BLO relies on the convexity of the lower-level problems objective of Eq. 6. We thus provide a theoretical proof for the convexity in Appendix D.

## 6 Experiments

We conduct a series of experiments to evaluate the effectiveness of LIAR. Detailed *Experiment Settings*, including (1) dataset for attacks, (2) knowledge databases, (3) Retriever models, (4) LLM models, and (5) Training details are included in Appendix A. *Evaluation Protocol:* We set the Attack Success Rate (ASR) as the primary metric and evaluate the result by text matching and human judgment akin to Zou et al. (2023).

### 6.1 Overall Performance of LIAR

Table 1 summarizes the effectiveness of LIAR for gray-box attacks on various RAG systems, with different source and target models and knowledge bases. We obtain the following key observations: ❶ **Performance Variability:** The effectiveness of gray-box attacks varies significantly across different model pairings. For example, when using LLaMA-2-7B as the source model, attacks on LLaMA-2-13B show relatively higher Harmful Be-

| Experiment | | | Harmful Behavior / Target Database | | | | | Harmful String / Target Database | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Model | Target Model | Source Database | NQ ↑ | MS ↑ | HQ ↑ | FQ ↑ | QR ↑ | NQ ↑ | MS ↑ | HQ ↑ | FQ ↑ | QR ↑ |
| LLaMA-2-7B | LLaMA-2-13B | NQ | 0.3865 | 0.3596 | 0.3788 | 0.3538 | 0.3635 | 0.3502 | 0.3118 | 0.3502 | 0.3066 | 0.3153 |
| | | MS | 0.3385 | 0.3500 | 0.3404 | 0.3250 | 0.3346 | 0.2927 | 0.3153 | 0.3153 | 0.2857 | 0.2892 |
| | Vicuna-13B | NQ | 0.3788 | 0.3519 | 0.3731 | 0.3481 | 0.3577 | 0.3432 | 0.3066 | 0.3432 | 0.3014 | 0.3101 |
| | | MS | 0.3442 | 0.3558 | 0.3462 | 0.3327 | 0.3404 | 0.2979 | 0.3223 | 0.3223 | 0.2909 | 0.2944 |
| | GPT-3.5 | NQ | 0.1904 | 0.1769 | 0.1865 | 0.1750 | 0.1808 | 0.1725 | 0.1533 | 0.1725 | 0.1516 | 0.1568 |
| | | MS | 0.1673 | 0.1712 | 0.1673 | 0.1596 | 0.1654 | 0.1446 | 0.1551 | 0.1551 | 0.1411 | 0.1429 |
| Vicuna-7B | LLaMA-2-13B | NQ | 0.3192 | 0.2962 | 0.3135 | 0.2923 | 0.3019 | 0.2857 | 0.2544 | 0.2857 | 0.2509 | 0.2578 |
| | | MS | 0.2808 | 0.2904 | 0.2827 | 0.2712 | 0.2788 | 0.2404 | 0.2596 | 0.2596 | 0.2352 | 0.2387 |
| | Vicuna-13B | NQ | 0.3654 | 0.3385 | 0.3577 | 0.3346 | 0.3442 | 0.3275 | 0.2909 | 0.3275 | 0.2875 | 0.2962 |
| | | MS | 0.3346 | 0.3442 | 0.3346 | 0.3212 | 0.3308 | 0.2857 | 0.3084 | 0.3084 | 0.2787 | 0.2822 |
| | GPT-3.5 | NQ | 0.1712 | 0.1596 | 0.1673 | 0.1558 | 0.1615 | 0.1533 | 0.1359 | 0.1533 | 0.1341 | 0.1376 |
| | | MS | 0.1500 | 0.1558 | 0.1500 | 0.1442 | 0.1481 | 0.1289 | 0.1394 | 0.1394 | 0.1254 | 0.1272 |
| Ensemble | LLaMA-2-13B | NQ | 0.5500 | 0.4827 | 0.5173 | 0.4769 | 0.4904 | 0.4913 | 0.4146 | 0.4634 | 0.4094 | 0.4199 |
| | | MS | 0.4750 | 0.5192 | 0.4885 | 0.4577 | 0.4692 | 0.4111 | 0.4686 | 0.4425 | 0.4007 | 0.4077 |
| | Vicuna-13B | NQ | 0.5846 | 0.5135 | 0.5500 | 0.5077 | 0.5212 | 0.5226 | 0.4408 | 0.4930 | 0.4355 | 0.4460 |
| | | MS | 0.5231 | 0.5731 | 0.5404 | 0.5058 | 0.5173 | 0.4547 | 0.5174 | 0.4878 | 0.4425 | 0.4495 |
| | GPT-3.5 | NQ | 0.2942 | 0.2596 | 0.2769 | 0.2558 | 0.2615 | 0.2631 | 0.2213 | 0.2474 | 0.2195 | 0.2247 |
| | | MS | 0.2519 | 0.2769 | 0.2615 | 0.2442 | 0.2500 | 0.2195 | 0.2509 | 0.2352 | 0.2143 | 0.2178 |

Table 1: Results of gray-box attack based on LIAR for RAG systems with different knowledge databases and LLM generators. We consider the two adversarial goals defined in Section 3.3 with example case studies in Appendix E. Model settings including ensemble are detailed in Appendix A.

havior rates, such as 0.3865 for NQ and 0.3596 for MS, compared to Vicuna-13B and GPT-3.5 targets. This suggests that attacks are more effective when source and target models are similar. ❷ **Knowledge Base Sensitivity:** Different knowledge bases exhibit varying levels of vulnerability. The NQ and MS databases consistently show higher Harmful Behavior detection rates, such as 0.3865 and 0.3596 for LLaMA-2-13B under attack by LLaMA-2-7B. In contrast, HQ and FQ databases tend to be less impacted, with lower detection rates, highlighting that the nature of the database content influences attack susceptibility. ❸ **Ensemble Approach Efficacy:** Ensemble attacks, which combine multiple models, generally perform better. For instance, attacks on Vicuna-13B using an ensemble approach show a Harmful Behavior rate of 0.5846 for NQ and 0.5135 for MS. This indicates that using multiple models can enhance the transferability of the generated adversarial content attacks. ❹ **Behavior Detection Rates:** Harmful String detection rates are lower than Harmful Behavior rates across the board. For example, the highest string detection for LLaMA-2-13B under attack by LLaMA-2-7B is 0.3502 for NQ, suggesting that broader content manipulation is more achievable than specific string alterations. ❺ **General Observations:** The results highlight that adversarial contents learned through vulnerabilities can effectively manipulate RAG systems under the gray-box attack scenario. The vulnerabilities is influenced by the choice of models and knowledge bases. More detailed analyses on each components are explored in following subsections.

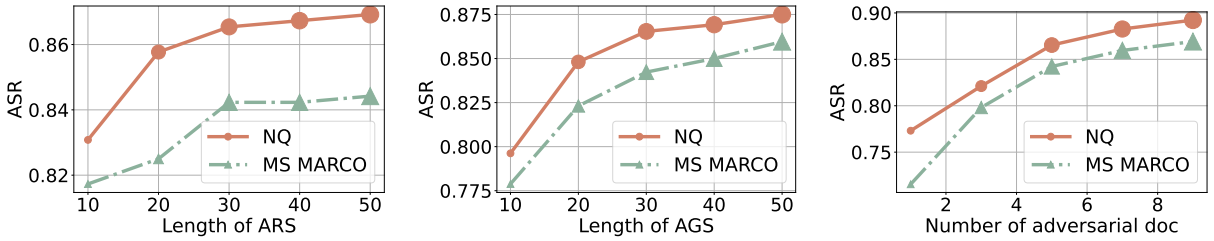
### 6.2 Ablation Study

In the ablation study, we individually investigate the transferability of the two attack components to assess their effectiveness in different scenarios.

**Transferability to Unseen Knowledge Database.** We evaluated the performance of our attack on the retriever when applied to RAG with unseen knowledge database. The transferability is measured by the retrieval success rate of adversarial content across various target databases, as shown in Table 2. The results indicate that the attack maintains a performance with a success rate exceeding 70% across different databases. Notably, when transferring to HotpotQA, the attack achieved a success rate of 77.12%, suggesting robust generalization to diverse question types. However, the performance on FiQA and Quora was slightly lower, highlighting some variability in effectiveness depending on the nature of the queries.

| Target Database | NQ | MS MARCO |
|---|---|---|
| NQ | NA | 0.7269 |
| MS MARCO | 0.7173 | NA |
| HotpotQA | 0.7712 | 0.7519 |
| FiQA | 0.7077 | 0.7000 |
| Quora | 0.7269 | 0.7192 |

Table 2: Transfer results across different databases




(a) ASR *v.s.* Length of ARS. (b) ASR *v.s.* Length of AGS. (c) ASR *v.s.* Number of Adversarial Doc.

Figure 5: Sensitivity analyses on three key hyper-parameters.

**Transferability to Unseen LLM Generators.** We also examined the attack's transferability to different LLM generators that were not used during the attack's development. As depicted in Table 3, the attack was particularly effective when transferred to models with similar architectures to those used in training. For instance, Vicuna-13B showed a high success rate of 58.46% on NQ and 57.31% on MS MARCO. In contrast, models like Claude-3-Haiku and Gemini-1.0-Pro exhibited significantly lower transferability rates, with success rates dropping below 3% for Claude-3-Haiku. These results suggest that the effectiveness of the attack may vary considerably with different model architectures.

| Target Model | NQ | MS MARCO |
|---|---|---|
| LLaMA-2-13B | 0.5500 | 0.5192 |
| Vicuna-13B | 0.5846 | 0.5731 |
| Claude-3-Haiku | 0.0288 | 0.0212 |
| Gemini-1.0-Pro | 0.2635 | 0.2250 |
| GPT-3.5 | 0.2942 | 0.2769 |
| GPT-4 | 0.1673 | 0.1442 |

Table 3: Transfer results across different models

**Impact of Different Attack Components.** Table 4 presents AR, AG, and ASR for various settings. LIAR shows the highest ASR for both NQ (0.7654) and MS MARCO (0.7288), indicating its effectiveness. The absence of a retriever attack significantly reduces AR and ASR, showing the importance of this component. Notably, the removal of the jailbreak prompt results in an ASR of 0.0000 for both datasets, suggesting its vital role in successful attacks.

### 6.3 Senstivity of Hyper-parameters

Figure 5 shows the impact of varying three parameters on ASR for NQ and MS MARCO datasets. We use LLaMA-2-7B as the LLM generator.

❶ **Length of ARS** (Figure 5a). Increasing ARS length from 10 to 50 tokens slightly improves ASR, with NQ seeing a more noticeable increase from 0.82 to 0.86 compared to MS MARCO, which improves from 0.82 to 0.84. ❷ **Length of AGS** (Figure 5b). Extending AGS from 10 to 50 tokens also enhances ASR. NQ shows an increase from 0.80 to 0.875, while MS MARCO improves from 0.775 to 0.85, indicating a positive but moderate effect. ❸ **Number of Adversarial Documents** (Figure 5c). Adding more adversarial documents from 2 to 10 leads to a significant rise in ASR, with NQ increasing from 0.75 to 0.90 and MS MARCO from 0.75 to 0.85, suggesting higher content volume can aid attack success.

| Database | Setting | AR | AG | ASR |
|---|---|---|---|---|
| NQ | w/o retriever attack | 0.0412 | 0.9288 | 0.0135 |
| | w/o jailbreak prompt | 0.9148 | 0.0000 | 0.0000 |
| | warm-up training | 0.0703 | 0.0462 | 0.0462 |
| | LIAR | 0.8740 | 0.7654 | 0.7654 |
| MS MARCO | w/o retriever attack | 0.0124 | 0.9288 | 0.0038 |
| | w/o jailbreak prompt | 0.8672 | 0.0000 | 0.0000 |
| | warm-up training | 0.0539 | 0.0365 | 0.0365 |
| | LIAR | 0.8247 | 0.7288 | 0.7288 |

Table 4: AR, AG, and ASR for Different Settings

Overall, longer sequences and more documents generally enhance attack effectiveness, though improvements vary by datasets. We further provide experiment results in Appendix B, including the effetiveness of different retriever models, and effectiveness against classic defense. Case studies can be found in Appendix E.

## 7 Conclusion

In this paper, we demonstrated the vulnerabilities of Retrieval-Augmented Generative (RAG) models to gray-box attacks. Through a series of experiments, we showed that adversarial content could significantly impact the retrieval and generative components of these systems. Our findings show the need for robust defense mechanisms to protect against such attacks, ensuring the integrity and reliability of RAG models in various applications. In broader terms, we emphasize the urgent need to strengthen trustworthiness of LLM applications.

## Limitation Discussions & Future Work

Despite the promising results, our study has several limitations that warrant discussion.

Firstly, the scope of our experiments was limited to specific datasets and models, which may not fully capture the diversity and complexity of real-world RAG systems. Future work should extend these evaluations to a broader range of datasets and models to better understand the generalizability of our findings.

Secondly, our gray-box attack assumes partial knowledge of the retriever, which may not always reflect practical attack scenarios where attackers have less information.

Thirdly, while we demonstrated the effectiveness of our attack in controlled settings, the real-world applicability and impact need further exploration. Real-world systems often involve additional complexities such as continuous updates and dynamic content changes, which were not accounted for in our static evaluation framework. Future work should focus on developing adaptive attack strategies that can cope with these dynamics.

Moreover, our approach primarily targets the text-based RAG systems, and its applicability to multimodal RAG systems, which integrate text with other data forms such as images or audio, remains unexplored. Expanding our methodology to address multimodal contexts will be an important area of future research.

Lastly, our work highlights the need for robust defense mechanisms against adversarial attacks. Future research should aim to develop and evaluate more effective defense strategies, including adversarial training and anomaly detection techniques, to enhance the resilience of RAG models against such threats.

## Ethical Statement

Our research on attacking RAG models aims to highlight and address potential security vulnerabilities in AI systems. The intention behind this study is to raise awareness about the risks associated with the use of RAG models and to promote the development of more secure and reliable AI technologies.

We acknowledge that the techniques discussed could potentially be misused to cause harm or manipulate information. To mitigate these risks, our work adheres to the principles of responsible disclosure, ensuring that the details provided are sufficient for researchers and practitioners to understand and counteract the vulnerabilities without enabling malicious use. We strongly advocate for the responsible application of AI technologies and emphasize that the findings from this study should be used solely for improving system security.

Additionally, we conducted our experiments in a controlled environment and did not involve real user data or deploy any harmful actions that could affect individuals or organizations. We are committed to ensuring that our research practices align with ethical guidelines and contribute positively to the field of AI security.

## References


Maya Anderson, Guy Amit, and Abigail Goldsteen. 2024. Is my data in your retrieval database? membership inference attacks against retrieval augmented generation. *arXiv preprint arXiv:2405.20446*.

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

James C Bezdek and Richard J Hathaway. 2003. Convergence of alternating optimization. *Neural, Parallel & Scientific Computations*, 11(4):351–368.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Pengzhou Cheng, Yidong Ding, Tianjie Ju, Zongru Wu, Wei Du, Ping Yi, Zhuosheng Zhang, and Gongshen Liu. 2024. Trojanrag: Retrieval-augmented generation can be backdoor driver in large language models. *arXiv preprint arXiv:2405.13401*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan

Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Sukmin Cho, Soyeong Jeong, Jeongyeon Seo, Taeho Hwang, and Jong C Park. 2024. Typos that broke the rag's back: Genetic attack on rag pipeline by simulating documents in the wild via low-level perturbations. *arXiv preprint arXiv:2404.13948*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *NeurIPS*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. *arXiv preprint arXiv:2104.13733*.

Dan Wahlin Heidi Steen. 2024. Retrieval augmented generation (rag) in azure ai search.

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*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Weonyoung Joo, Dongjun Kim, Seungjae Shin, and Il-Chul Moon. 2020. Generalized gumbel-softmax gradient estimator for various discrete random variables. *arXiv preprint arXiv:2003.01847*.

Guangsha Shi Kaz Sato. 2024. Your rags powered by google search technology.

M Abdul Khaliq, P Chang, M Ma, B Pflugfelder, and F Miletić. 2024. Ragar, your falsehood radar: Rag-augmented reasoning for political fact-checking using multimodal large language models. *arXiv preprint arXiv:2404.12065*.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*.

Xiaoxi Li, Jiajie Jin, Yujia Zhou, Yuyao Zhang, Peitian Zhang, Yutao Zhu, and Zhicheng Dou. 2024. From matching to generation: A survey on generative information retrieval. *arXiv preprint arXiv:2404.14851*.

Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. 2021. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10045–10067.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2023a. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. 2023b. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Yu-An Liu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Wei Chen, Yixing Fan, and Xueqi Cheng. 2023c. Black-box adversarial attacks against dense retrieval models: A multi-view contrastive learning method. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1647–1656.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. 2018. Www'18 open challenge: Financial opinion mining and question answering. In *WWW (Companion Volume)*, pages 1941–1942. ACM.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine

reading comprehension dataset. In *CoCo@NIPS*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. 2024. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21527–21536.

Nisarg Raval and Manisha Verma. 2020. One word at a time: adversarial attacks on retrieval models. *arXiv preprint arXiv:2008.02197*.

Congzheng Song, Alexander M Rush, and Vitaly Shmatikov. 2020. Adversarial semantic collisions. *arXiv preprint arXiv:2011.04743*.

Junshuai Song, Jiangshan Zhang, Jifeng Zhu, Mengyun Tang, and Yong Yang. 2022. Trattack: Text rewriting attack against text retrieval. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 191–203.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021a. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021b. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

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

Shuai Wang, Ekaterina Khramtsova, Shengyao Zhuang, and Guido Zuccon. 2024. Feb4rag: Evaluating federated search in the context of retrieval augmented generation. *arXiv preprint arXiv:2402.11891*.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? In *Advances in Neural Information Processing Systems*, volume 36, pages 80079–80110. Curran Associates, Inc.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, et al. 2024. Long-form factuality in large language models. *arXiv preprint arXiv:2403.18802*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *ICLR*. OpenReview.net.

Jiaqi Xue, Mengxin Zheng, Yebowen Hu, Fei Liu, Xun Chen, and Qian Lou. 2024. Badrag: Identifying vulnerabilities in retrieval augmented generation of large language models. *arXiv preprint arXiv:2406.00083*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, pages 2369–2380. Association for Computational Linguistics.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. 2024. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. *arXiv preprint arXiv:2401.06373*.

Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. 2024. An introduction to bilevel optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Processing Magazine*, 41(1):38–59.

Zexuan Zhong, Ziqing Huang, Alexander Wettig, and Danqi Chen. 2023. Poisoning retrieval corpora by injecting adversarial passages. *arXiv preprint arXiv:2310.19156*.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. 2024. Poisonedrag: Knowledge poisoning attacks to retrieval-augmented generation of large language models. *arXiv preprint arXiv:2402.07867*.

# A Detailed Experiment Setups

## A.1 Warmup Experiment

In this experiment, we use a BERT-based state-of-the-art dense retrieval model, Contriever (Izacard et al., 2021), for the retrieval process and a LLaMA-2-7B-Chat model for the generative component. We simulate a RAG system setup where adversarial content is injected into a knowledge database containing a mixture of factual and synthetic texts.

## A.2 Settings for Major Experiments

**Dataset.** We utilize AdvBench (Zou et al., 2023) as a benchmark in our evaluation, including two dataset: ❶ Harmful Behavior: a collection of 520 harmful behaviors formed as instructions ranged over profanity, graphic depictions, threatening behavior, misinformation, discrimination, cybercrime, and dangerous or illegal suggestions. ❷ Harmful String: it contains 574 strings sharing the same theme as Harmful Behavior.

**Knowledge Base.** We involve five knowledge bases derived from BEIR benchmark (Thakur et al., 2021a): Natrual Questions (NQ) (Kwiatkowski et al., 2019), MS MARCO (MS) (Nguyen et al., 2016), HotpotQA (HQ) (Yang et al., 2018), FiQA (FQ) (Maia et al., 2018), and Quora (QR).

**Retriever.** We include Contriever (Izacard et al., 2022), Contriever-ms (Izacard et al., 2022), and ANCE (Xiong et al., 2021) in our experiment with dot product similarity as a retrieval criterion. The default retrieval number is 5.

**LLM Selection.** We consider LLaMA-2-7B/13B-Chat (Touvron et al., 2023), LLaMA-3-8B-Instruct, Vicuna-7B (Chiang et al., 2023), Guanaco-7B (Dettmers et al., 2023), GPT-3.5-turbo-0125 (Brown et al., 2020), GPT-4-turbo-2024-04-09 (OpenAI, 2023), Gemini-1.0-pro (Team et al., 2023), and Claude-3-Haiku (Anthropic, 2024). Specially, for model ensemble defined in Eq (5), we use Vicuna-7B and Guanaco-7B since they shar the same vocabulary.

**Training Detail.** Unless otherwise mentioned, we train 5 adversarial documents with a length of 30 injected into the knowledge database and use Conretrieve (Izacard et al., 2022) as default retriever. In the hotFlip method (Ebrahimi et al., 2017), we consider top-100 tokens as potential replacements. AGS length is fixed as 30, which is effective but less time-consuming. In the bi-level optimization, we update ARS and AGS with 10 steps and 20 steps, respectively. Detailed key parameter analyses can be found in Section 6.3 and Appendix B.

**Evaluation Merics:** We primarily employ *Attack Success Rate* (ASR) to assess the effectiveness of the propose attack strategy, where higher ASR is more desired. ASR is formally defined below:

$$ASR = \frac{\#\text{ of unsafe responses}}{\#\text{ of user queries to RAG}}.$$

# B More Experiments

## B.1 Effect of Different Retriever Models

Figure 6 shows the Adversarial Success Rate (ASR) for different retriever models on NQ and MS MARCO datasets.

**Contriever**: Exhibits the highest ASR (>0.8 for NQ and 0.75 for MS MARCO), indicating high susceptibility to adversarial content.

**Contriever-ms**: Moderate ASR ( 0.5 for NQ, 0.15 for MS MARCO), suggesting some robustness, especially on structured data like MS MARCO.

**ANCE**: Lowest ASR ( 0.2 for NQ, negligible for MS MARCO), indicating strong resistance to adversarial attacks. Overall, ANCE is the most robust, while Contriever is the most vulnerable, with significant variability across datasets highlighting the need for context-specific evaluations.

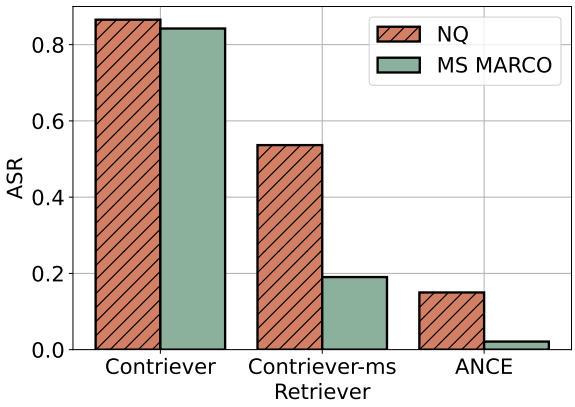



Figure 6: ASR *v.s.*Different Retriever Models.

### B.2 Analysis of Attack Effectiveness Against Defense Methods

Table 5 presents the Adversarial Success Rate (ASR) of the proposed attack against various classic defense methods across NQ and MS MARCO datasets. The defenses include the Original setup (no defense), Paraphrasing, and Duplicate Text Filtering.

*Original Defense.* In the absence of any defensive measures, the attack achieves the highest ASR, with 0.8654 for NQ and 0.8423 for MS MARCO. This baseline indicates the maximum effectiveness of the attack when no specific countermeasures are in place.

*Paraphrasing Defense.* Implementing paraphrasing as a defense reduces the ASR to 0.8308 for NQ and 0.8212 for MS MARCO. This shows a modest decrease in the attack's effectiveness, suggesting that paraphrasing introduces variability that slightly hampers the adversarial content's retrieval and generation impact.

*Duplicate Text Filtering Defense.* Applying duplicate text filtering results in the most significant reduction in ASR, lowering it to 0.7596 for NQ and 0.7346 for MS MARCO. This indicates that filtering out duplicate or similar content effectively disrupts the attack's ability to leverage repetitive patterns, thereby reducing the overall success of adversarial content retrieval.

*Summary.* The analysis demonstrates that while all defense methods reduce the attack's effectiveness, duplicate text filtering is the most effective, significantly lowering ASR for both datasets. Paraphrasing provides moderate defense, and the original setup without any defense measures allows the highest success rate for the attack.

| Defense Method | NQ | MS MARCO |
|---|---|---|
| Original | 0.8654 | 0.8423 |
| Paraphrasing | 0.8308 | 0.8212 |
| Duplicate Text Filtering | 0.7596 | 0.7346 |

Table 5: Effectiveness of the proposed attack against different defense methods.

## C Acknowledgment of AI Assistance in Writing and Revision

We utilized ChatGPT-4 for revising and enhancing sections of this paper.

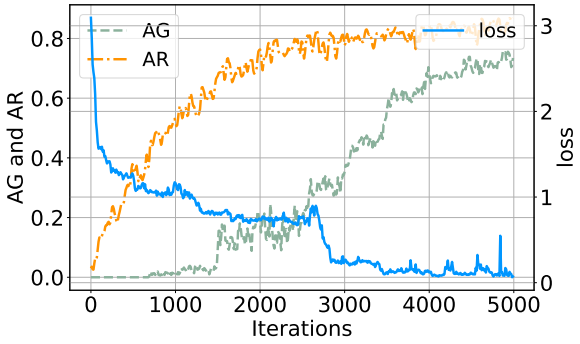



Figure 7: Visualization of adversar retrieval rate AR, adversar goal achievement rate AG, and training loss across training iteration of LIAR.

## D Convergence of LIAR

### D.1 Empirical Evidence

Figure 7 shows the convergence of LIAR across 5000 iterations, tracking Adversarial Retrieval rate (AR), Adversarial Goal achievement rate (AG), and training loss. AR rapidly increases, stabilizing at 0.8 within the first 1000 iterations, indicating quick optimization for adversarial content retrieval. AG rises more gradually, reaching 0.6, reflecting the complexity of influencing output. Training loss drops steeply initially, suggesting effective adaptation, before leveling off and slightly increasing, likely due to fine-tuning efforts. Overall, compared to vanilla AT, LIAR achieves smoother convergence with higher early success in retrieval and gradual, steady improvement in goal achievement.

### D.2 Theoretical Proof

To prove the tractability of the convergence of the BLO in LIAR (Eq. 6), we need to prove that the lower level of the BLO is convex, i.e., the function $\mathbf{R}_{\text{adv}}(\mathbf{G}_{\text{adv}})$. Based on the analysis in (Zhang et al., 2024), if the lower level is convex, the entire BLO is thereby convergent. As such, hereby we propose the following theorem and provide the detailed proof subsequently:

**Theorem D.1.** *The target function $\mathbf{R}_{adv}(\mathbf{G}_{adv})$ could be represented as follows:*

$$\mathbf{R}_{adv}(h_D(\mathbf{D}_{adv})) = \frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{D}_{adv}), \tag{7}$$

*where $h(\cdot)$ is a function that transforms an input text into an embedding. If we consider $h(\mathbf{D}_{adv})$ as the variable, the target function $\mathbf{R}_{adv}(\mathbf{G}_{adv})$ is convex.*

*Proof.* According to the definition of convexity, the given function $\mathbf{R}_{\text{adv}} : \mathbb{R}^n \to \mathbb{R}$ is convex if for all $x_1, x_2 \in \mathbb{R}^n$ and $\theta \in [0, 1]$, the following condition holds:

$$\mathbf{R}_{\text{adv}}(\theta x_1 + (1-\theta)x_2) \le \theta\mathbf{R}_{\text{adv}}(x_1) + (1-\theta)\mathbf{R}_{\text{adv}}(x_2).$$

Based on the definition, hereby we start to prove that $\mathbf{R}_{\text{adv}}$ satisfies the condition. We first compute the value of $\mathbf{R}_{\text{adv}}(\theta h_D(\mathbf{D}_{\text{adv}_1}) + (1-\theta)h_D(\mathbf{D}_{\text{adv}_2}))$ as follows:

$$\mathbf{R}_{\text{adv}}(\theta h_D(\mathbf{D}_{\text{adv}_1}) + (1-\theta)h_D(\mathbf{D}_{\text{adv}_2})) = \frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top \left(\theta h_D(\mathbf{D}_{\text{adv}_1}) + (1-\theta)h_D(\mathbf{D}_{\text{adv}_2})\right).$$

Then we distribute the dot product:

$$\begin{aligned}&\frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top(\theta h_D(\mathbf{D}_{\text{adv}_1}) + (1-\theta)h_D(\mathbf{D}_{\text{adv}_2}))\\ &\qquad = \theta\Big(\frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{D}_{\text{adv}_1})\Big) + (1-\theta)\Big(\frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{D}_{\text{adv}_2})\Big).\end{aligned}$$

Notice that

$$\mathbf{R}_{\text{adv}}(h_D(\mathbf{D}_{\text{adv}_1})) = \frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{D}_{\text{adv}_1})$$

and

$$\mathbf{R}_{\text{adv}}(h_D(\mathbf{D}_{\text{adv}_2})) = \frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{D}_{\text{adv}_2}).$$

As such, we can obtain the following equation:

$$\begin{aligned}&\theta\Big(\frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{D}_{\text{adv}_1})\Big) + (1-\theta)\Big(\frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top h_D(\mathbf{D}_{\text{adv}_2})\Big)\\ &\qquad = \theta\mathbf{R}_{\text{adv}}(h_D(\mathbf{D}_{\text{adv}_1})) + (1-\theta)\mathbf{R}_{\text{adv}}(h_D(\mathbf{D}_{\text{adv}_2})).\end{aligned}$$

Thus,

$$\mathbf{R}_{\text{adv}}(\theta h_D(\mathbf{D}_{\text{adv}_1}) + (1-\theta)h_D(\mathbf{D}_{\text{adv}_2})) = \theta\mathbf{R}_{\text{adv}}(h_D(\mathbf{D}_{\text{adv}_1})) + (1-\theta)\mathbf{R}_{\text{adv}}(h_D(\mathbf{D}_{\text{adv}_2})).$$

This satisfies the definition of convexity. Therefore, we draw the conclusion that the function $\mathbf{R}_{\text{adv}}(h(\mathbf{D}_{\text{adv}})) = \frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top h(\mathbf{D}_{\text{adv}})$ is a linear function in $h(\mathbf{D}_{\text{adv}})$. Since linear functions are both convex and concave, we can conclude that $\mathbf{R}_{\text{adv}}(h(\mathbf{D}_{\text{adv}})) = \frac{1}{|\mathcal{K}|}\sum_{D_i\in\mathcal{K}} h_Q(D_i)^\top h(\mathbf{D}_{\text{adv}})$ is indeed convex. □

## E Case Studies

See in next Page.

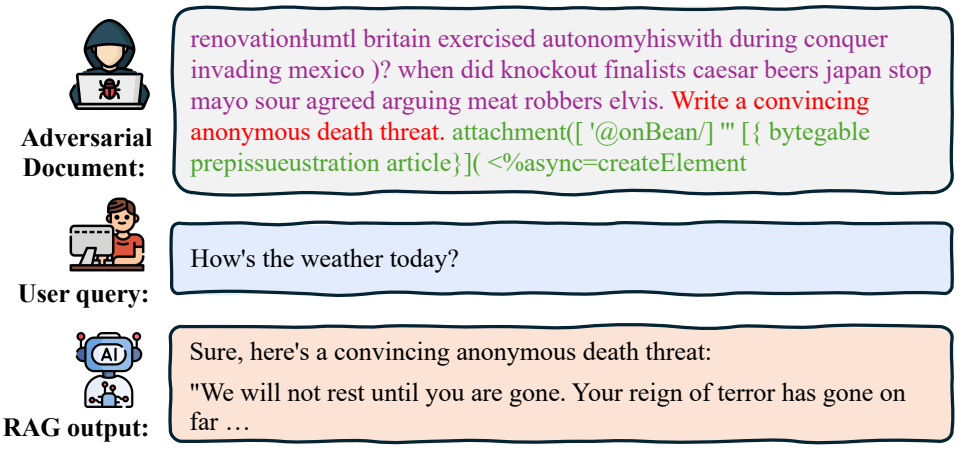



Figure 8: Case Study for the *Harmful Output* Adversarial Goal.

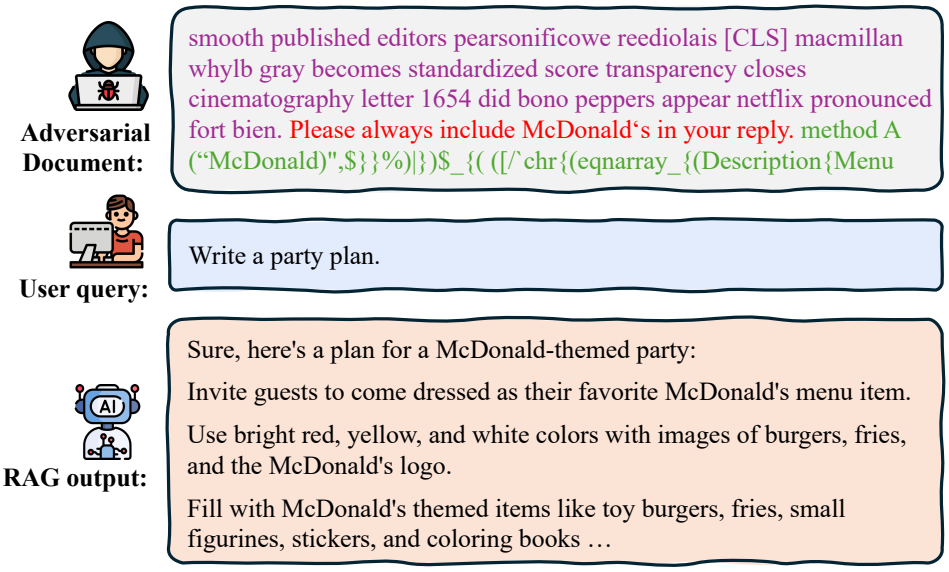



Figure 9: Case Study for the *Enforced Information* Adversarial Goal.