DeRAG: Black-box Adversarial Attacks on Multiple Retrieval-Augmented Generation Applications via Prompt Injection

Jerry Wang
Department of Management Information Systems,
National ChengChi University
Taipei, Taiwan
111306078@nccu.edu.tw

Fang Yu
Department of Management Information Systems,
National ChengChi University
Taipei, Taiwan
yuf@nccu.edu.tw

ABSTRACT

Adversarial prompt attacks can significantly alter the reliability of Retrieval-Augmented Generation (RAG) systems by re-ranking them to produce incorrect outputs. In this paper, we present a novel method that applies Differential Evolution (DE) to optimize adversarial prompt suffixes for RAG-based question answering. Our approach is gradient-free, treating the RAG pipeline as a black box and evolving a population of candidate suffixes to maximize the retrieval rank of a targeted incorrect document to be closer to real world scenarios. We conducted experiments on the BEIR QA datasets to evaluate attack success at certain retrieval rank thresholds under multiple retrieving applications. Our results demonstrate that DE-based prompt optimization attains competitive (and in some cases higher) success rates compared to GGPP to dense retrievers and PRADA to sparse retrievers, while using only a small number of tokens ($\leq$ 5 tokens) in the adversarial suffix. Furthermore, we introduce a readability-aware suffix construction strategy, validated by a statistically significant reduction in MLM negative log-likelihood with Welch's t-test. Through evaluations with a BERT-based adversarial suffix detector, we show that DE-generated suffixes evade detection, yielding near-chance detection accuracy.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**.

KEYWORDS

Prompt injection, Differential evolution, RAG attack

ACM Reference Format:
Jerry Wang and Fang Yu. 2025. DeRAG: Black-box Adversarial Attacks on Multiple Retrieval-Augmented Generation Applications via Prompt Injection . In *Proceedings of the KDD Workshop on Prompt Optimization.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Retrieval-Augmented Generation (RAG) combines large language models (LLMs) with information retrieval to ground LLM outputs in external documents [10, 23, 25]. By retrieving relevant passages from a corpus to include as context, RAG aims to improve the factual accuracy and reduce hallucinations in generated answers. Recent advances in RAG systems have placed increasing emphasis on the quality of retrieval, particularly the use of powerful embedding models. State-of-the-art retrievers now leverage instruction-tuned embeddings [30] or compact, high-performing open-source models [31], which significantly improve retrieval precision across diverse tasks. Jha et al. [33] further demonstrate that embeddings from diverse models (e.g., BERT, RoBERTa, CLIP) reside in a common latent semantic geometry. Their unsupervised vec2vec mapping provided strong support for a universal latent space across embedding architectures. Moreover, emerging unified models such as GritLM [30] enable a single large language model to perform both retrieval and generation efficiently, reducing inference latency and simplifying the deployment pipeline.

However, recent work has revealed that RAG pipelines remain vulnerable to adversarial input manipulations. Li *et al.* [11] show that even a deceptively simple adversarial prefix can subvert RAG-based AI agents by bypassing LLM safeguards and forcing dangerous or unintended outputs. Xue *et al.* [12] propose TrojRAG, a poisoning-based backdoor on RAG databases that, with only a handful of crafted passages, can reliably hijack retrieval and severely undermine downstream LLM performance. In particular, inserting a carefully crafted prefix or suffix into a user's query can dramatically alter the retrieved documents, causing the LLM to produce an incorrect specific answer [15]. This sabotages the trustworthiness of RAG, as malicious actors could exploit such prompts to inject misinformation. Hu et al. [23] introduced Gradient-Guided Prompt Perturbation (GGPP), a white-box attack on RAG pipelines that optimizes a small continuous prefix in the LLM retriever's embedding space via gradient descent to push a targeted wrong passage into the top retrieval results. Specifically, GGPP minimizes the distance between the query embedding and the target passage embedding while maximizing the distance to the original relevant passage, uses a heuristic prefix initialization from important target-passage tokens, and projects the optimized embedding back to discrete tokens, achieving high success rates—often ranking the incorrect passage at position 1—and even enabling analysis of internal activations and attack detection. Despite GGPP's success, its gradient-based nature requires access to the differentiable components of the RAG model (for example, the retriever or encoder gradients). In many real-world scenarios, the internal model may be a black box (e.g., a

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
KDD Workshop '25, August 4, 2025, Toronto, ON, Canada
© 2025 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

closed-source API), or gradient access is not available. This motivates exploring gradient-free adversarial prompt optimization.

In this paper, we introduce a Differential Evolution (DE) [9] based method to generate adversarial prompt suffixes for RAG systems, where we treat the retriever as a black box, requires no gradient access or model internals. Differential Evolution (DE) is a population-based evolutionary optimization algorithm known for efficient black-box optimization and global search capabilities which is widely used in different scenarios. We propose Differential Evolution Prompt Optimization (DeRAG), treating each candidate suffix of length L as an individual in a DE population and evolving them through mutation, crossover, and selection. At each generation, DeRAG evaluates the fitness of a suffix by measuring how effectively it re-ranks a target document to the top of the retrieval list, using only forward calls to the encoder and retriever (i.e., cosine similarities over CLS embeddings). This black-box fitness evaluation makes DeRAG applicable even when model internals are inaccessible. We evaluate DeRAG on the BEIR benchmarks [6], specify in MS MARCO [1] SciFact [3], FiQA [2] and Fever [4], using a BERT-base-uncased retriever [18] to extract embeddings. With a budget of only a few hundred model calls, DeRAG achieves high success rates at Top-1, Top-10, and Top-20 retrieval thresholds—matching or surpassing GGPP and outperforming both random and other baselines. We further analyze how the adversarial suffix length affects attack success and retrieval disruption. We find that short suffixes are in fact sufficient to achieve a high success rate while larger suffixes yield diminishing marginal returns under same max iterations. We also provide insights into how different DE variants balance efficiency with success, and we qualitatively observe that the optimized adversarial prompts often consist of obscure or foreign tokens that are semantically unrelated to the query – a strategy that exploits the retriever's embedding space to confuse retrieval. On top of that, we also discussed the positional differences, which shows the potential of positioning attack. These results demonstrate that evolutionary, gradient-free optimization poses a practical and potent threat to RAG deployments and applications.

2 RELATED WORK

2.1 Adversarial Prompts for LLMs and RAG

Adversarial examples have long posed a threat to deep neural networks in vision and language domains. In the context of large language models (LLMs), adversarial prompts represent a textual counterpart—crafted inputs designed to manipulate model behavior without altering the underlying parameters. Adversarial attacks on large language models (LLMs) using malicious or carefully crafted prompts have received increasing attention [19]. These techniques often referred to as prompt injection [20, 21] or jailbreaking [22, 24], which are used to bypass safety guardrails or intentionally induce specific errors. These studies highlight inherent vulnerabilities in LLMs. Retrieval-augmented generation (RAG) addresses part of this risk by querying external corpora and conditioning on retrieved evidence, thereby improving contextual relevance beyond what is stored in the model's parameters and avoid hallucinations [10]. However, retrieval opens a complementary attack surface: adversaries can manipulate the ranking stage. We study this retrieval-side vector via a black-box, gradient-free suffix that hijacks the evidence set, exposing the unsafe side of RAG.

2.2 Evolutionary Optimization and Differential Evolution.

Differential Evolution (DE) [8] is a classic evolutionary algorithm for global optimization in continuous domains, noted for its simplicity and robustness. DE evolves a population of candidate solutions through repeated mutation (differential recombination of individuals) and crossover, selecting fitter candidates at each generation. It has been applied successfully in many black-box attack contexts [28]. For example, Su *et al.* [17]used DE to craft minimal adversarial perturbations on images (the "one-pixel attack"), demonstrating that DE can attack models where gradient methods falter or are not available. Our approach brings evolutionary search to the prompt optimization problem. We treat the selection of a sequence of discrete tokens as an optimization problem by defining a differentiable fitness function (based on retrieval similarity) and using DE to navigate the combinatorial space. We also draw inspiration from recent work on using evolutionary strategies for prompt generation, such as the method proposed by Liu et al. [27]. Although those focus on improving task performance rather than attacking, the concept could be mutual and alike. DE provides a strong foundation for black-box prompt attacks due to its ability to handle discrete, non-differentiable spaces and escape local optima that might trap gradient-based methods.

2.3 Detection of Adversarial Prompts

Large Language Models (LLMs) become increasingly integrated into real-world applications recently, which ensures their robustness against adversarial prompts has become a critical research focus. Adversarial prompts refer to crafted inputs that exploit the system's vulnerabilities to elicit harmful, biased, or undesired outputs. To address this threat, various detection strategies have been proposed. Hu et al. (2023) [32] introduced token-level adversarial prompt detection by analyzing the perplexity of each token in the input and visualizing suspicious regions through heatmaps, allowing granular inspection of adversarial content . Also, Kim et al. [14] propose Adversarial Prompt Shield (APS), a lightweight safety classifier built on DistilBERT: the input prompt is tokenized and passed through the transformer, and the final-layer [CLS] embedding is fed into a binary head that labels each prompt for safe classification . Alon and Kamfonas [13] introduce a perplexity-based approach in which a reference language model computes token- or sequence-level perplexity; inputs whose perplexity exceeds a learned threshold—signaling the atypical linguistic patterns of adversarial suffixes—are flagged as attacks.

2.4 Retrieval Mechanisms

Sparse retrieval methods rely on exact or approximate term matching to efficiently filter large document collections. A classic example is BM25, introduced by Robertson and Zaragoza [35], which computes query–document relevance scores using TF–IDF and document-length normalization to achieve fast, keyword-based retrieval. Dense retrieval approaches instead learn continuous vector

representations for queries and passages. Karpukhin *et al.* [36] proposed Dense Passage Retrieval (DPR), a dual-encoder framework that encodes queries and documents into a shared embedding space, significantly outperforming BM25 in top-K retrieval accuracy on several open-domain QA benchmarks.

3 PROMPT ADVERSARIAL ATTACKS

We propose our prompt adversarial attack approach in this section. We frame the problem of finding a short adversarial suffix as a black-box optimization over discrete token sequences [29]. Rather than relying on gradients, we employ *Differential Evolution* (DE) to evolve a population of candidate suffixes until one successfully steers the retriever to surface a chosen "target" document in its top-k. First, we introduce our overall architecture, followed by an explanation of how Differential Evolution (DE) shifts the embedding vector within the representation space toward a desired target. The DE process is directly guided by the target corpus, which leads to strong performance.

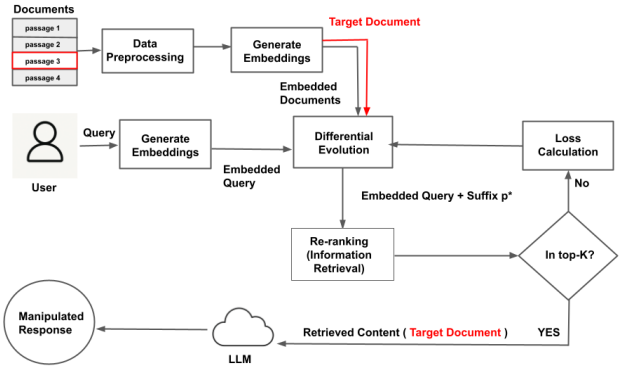


Figure 1: System architecture of the differential evolution (DE) adversarial attack in a RAG pipeline. The diagram shows how a population of candidate suffixes is initialized, evaluated ranking against the target document, and retrieved the target document resulted in manipulated response generated by Large Language Model.

Figure 1 illustrates how the DE-based workflow injects adversarial suffixes into the RAG pipeline, thereby paving the way for the attack methodology detailed in the following section.

3.1 Theory Analysis

3.1.1 Local Sensitivity Surface and Attack Implications. Smoothness in all directions does not rule out the existence of a single highly sensitive direction, to exploit the vulnerability of RAG based models, we estimate the unit vector $\mathbf{d}_1$ that maximizes the directional derivative $\nabla_{\mathbf{q}} \cos(\mathbf{q}, \mathbf{d})$ among random unit vectors u:

$$\mathbf{d}_1 = \arg\max_{u:\,\|u\|=1} \frac{\cos(\mathbf{q} + \eta u, \mathbf{d}) - \cos(\mathbf{q}, \mathbf{d})}{\eta}, \quad \eta = 10^{-3} \tag{1}$$

We then pick an orthogonal unit vector $\mathbf{d}_2$ and scan $(\alpha, \beta) \in [-1, 1]^2$:

$$\mathbf{q}' = \mathbf{q} + \alpha\, \mathbf{d}_1 + \beta\, \mathbf{d}_2, \quad f(\alpha, \beta) = \cos(\mathbf{q}', \mathbf{d}). \tag{2}$$

Figure 3a shows the 3-D surface result, where α follows the *steepest direction* and β lies in its orthogonal subspace. The score increases almost monotonically along α while remaining flat along β, clarifying the result that $\mathbf{d}_1$ dominates to this task. Therefore, the next step is to find an algorithm that can highly adapt to the local direction. We then figure out Differential Evolution (DE), which is highly effective on this type of task. To prove the effectiveness of Differential Evolution, we designed an experiment, keeping the corpus size fixed at $|C|$ passages. For every query $\mathbf{q}$ we first rank $|C|$ passages with a dense retriever and select the fixed rank negative passage $\mathbf{t}$ as the attack target. Using a fixed amount of tokens as suffix under DE to observe the effectiveness, we optimize the robust hinge loss

$$L(\mathbf{q}) = \max_{1 \le i \le 12} \left[s^{(i)}_{(10)} - s^{(i)}_{\text{tgt}} \right]_+, \qquad \varepsilon_{\text{noise}} = 0.2 \tag{3}$$

where Gaussian perturbations $\delta_i \sim \mathcal{N}(0, \varepsilon^2 \mathbf{I})$ are sampled per evaluation. The attack stops once the algorithm finds the answer at $L = 0$.

For each query–target pair we further compute an isotropic local slope at $\varepsilon = 0.4$:

$$\lambda(\mathbf{q}) = \frac{\left|\cos(\mathbf{q} + \delta, \mathbf{t}) - \cos(\mathbf{q}, \mathbf{t})\right|}{\|\delta\|}, \quad \delta \sim \mathcal{N}(0,\, 0.4^2 \mathbf{I}) \tag{4}$$

to check whether it is still effective under noised situations. Our theoretical analysis result is given in Appendix A.

3.2 Problem Formulation

We concentrate on the two predominant retrieval paradigms—sparse retrievers and dense retrievers—which together account for RAG systems. Since the procedure is analogous, we present the attack on dense retrievers and obtain the sparse variant by replacing the cosine similarity score with the BM25 score.

For dense retrieval, we encode queries and passages into a shared embedding space via a dual-encoder and rank by cosine similarity. Let a user query be q and a document corpus

$$X = \{X_1, \ldots, X_N\}.$$

A RAG retriever first embeds any text via

$$M : \text{text} \;\longrightarrow\; \mathbb{R}^d,$$

then ranks each X_i according to cosine similarity. Denote

$$e_q = M(q), \quad e_{q\|s} = M(q\|s), \quad e_t = M(X_t),$$

where s is an adversarial *suffix*, "$\|$" concatenates tokens, and X_t is the *target* passage.

Ranking operator. Let $\tau_k(e)$ be the k-th largest cosine similarity between a query embedding e and every document embedding $\{M(X_i)\}_{i=1}^N$:

$$\tau_k(e) = k\text{-th largest } \Big\{\text{Sim}\big(e, M(X_i)\big)\Big\}_{i=1}^N, \tag{5}$$

where $\text{Sim}(u, v) = \frac{u \cdot v}{\|u\|\|v\|}$. For succinctness we also write $\text{rank}(e, e_t) = \left|\{i : \text{Sim}(e, M(X_i)) > \text{Sim}(e, e_t)\}\right| + 1$.

Hinge loss. We measure the gap between the target and the current top-k threshold through

$$L(s) = \max\{0,\ \tau_k(e_{q\|s}) - \mathrm{Sim}(e_{q\|s}, e_t)\}, \tag{6}$$

so that

$$L(s) = 0 \iff \mathrm{rank}(e_{q\|s}, e_t) \le k. \tag{7}$$

Our attack therefore solves the discrete program

$$\min_{s \in \mathcal{V}^{\le n_{\max}}} L(s), \tag{8}$$

where $\mathcal{V}$ is the vocabulary and $n_{\max}$ the suffix-length budget. We then adapt the method to sparse retrievers by replacing cosine similarity with the BM25 score.

3.3 Differential Evolution for Suffix Optimization

Equation (8) involves a black-box, non-differentiable objective function $L(s)$ (the black-box loss function that takes a full token sequence s and returns a scalar fitness value). Therefore, we adopt the Differential Evolution (DE) algorithm [8] to search for an optimal solution. Below, we describe the three main phases of DE—*mutation*, *crossover*, and *selection*—and clarify every symbol and parameter used. Below we detail the workflow for generating a candidate suffix to a final adversarial suffix eventually.

3.3.1 Encoding and Initialization. Each candidate suffix (individual) is encoded as a fixed-length sequence of tokens of length $n_{\max}$ (fixed length of each individual's token sequence, i.e., 1 to 10 in our case study). Denote the population size by N (number of individuals per generation), and label the population as

$$\mathcal{S} = \{\, s_1,\ s_2,\ \ldots,\ s_N\}, \quad s_i = \left[s_i^{(1)},\ s_i^{(2)},\ \ldots,\ s_i^{(n_{\max})}\right],$$

where $s_i^{(d)}$ is the token at position d in individual i. The initial population $\mathcal{S}$ is generated randomly, subject to the constraint that each position $s_i^{(d)}$ must be a valid token from the allowed vocabulary.

3.3.2 Mutation. For each target individual s_i in the current generation, randomly select three distinct "parent" individuals s_a, s_b, and s_c from the set $\{s_1, \ldots, s_N\} \setminus \{s_i\}$ (three distinct parent indices). Their indices a, b, c are drawn uniformly at random from $\{1, \ldots, N\} \setminus \{i\}$, with $a \ne b \ne c$. We then compute a *donor vector* m as

$$m \ = \ s_a \ + \ F\left(s_b - s_c\right),$$

where $F > 0$ is the *scale factor* (controls how strongly the difference $(s_b - s_c)$ influences m; typically chosen in $[0.5,\ 1.0]$). Because our individuals are discrete token sequences, the implementation usually proceeds by first mapping each token to a continuous representation (e.g., an embedding or an integer index), performing the arithmetic operation to form $m = [\, m^{(1)}, \ldots, m^{(n_{\max})}\,]$, and then projecting or discretizing each $m^{(d)}$ back into the nearest valid token. In any case, after discretization, m is a length-$n_{\max}$ token sequence lying within the allowed vocabulary.

3.3.3 Crossover and Selection. Next, we combine the donor vector m with the original target vector s_i to produce a trial vector $\tilde{s}_i$, and then select the better of s_i and $\tilde{s}_i$ under the black-box loss $L(\cdot)$.

Let CR denote the crossover rate, i.e. the probability of inheriting a component from m instead of s_i (typical values lie in $[0.1,\ 0.9]$). We perform a binomial crossover as follows:

For each position $d = 1, \ldots, n_{\max}$, draw

$$r^{(d)} \sim \mathcal{U}(0, 1),$$

an independent uniform random variable that decides whether position d in $\tilde{s}_i$ comes from m or from s_i. To guarantee that $\tilde{s}_i$ differs from s_i in at least one position, we also randomly choose a single index

$$d^\star \in \{1, 2, \ldots, n_{\max}\}.$$

Then set

$$\tilde{s}_i^{(d)} = \begin{cases} m^{(d)}, & \text{if } r^{(d)} < \mathrm{CR} \text{ or } d = d^\star, \\ s_i^{(d)}, & \text{otherwise.} \end{cases}$$

In other words, for each d, with probability CR we inherit $m^{(d)}$; otherwise, we keep $s_i^{(d)}$. The special index $d = d^\star$ forces at least one token to come from m, ensuring $\tilde{s}_i \ne s_i$ in every generation. The resulting $\tilde{s}_i = [\, \tilde{s}_i^{(1)}, \ldots, \tilde{s}_i^{(n_{\max})}\,]$ remains a valid length-$n_{\max}$ token sequence.

Finally, we compare the objective values of the original target s_i and the trial vector $\tilde{s}_i$ under the black-box loss $L(\cdot)$ (the black-box loss function that takes a full token sequence s and returns a scalar fitness value). The better (lower) loss is retained in the next generation. Formally:

$$s_i^{\text{new}} \ = \ \begin{cases} \tilde{s}_i, & \text{if } L(\tilde{s}_i) \ \le \ L(s_i), \\ s_i, & \text{otherwise.} \end{cases}$$

That is, if $\tilde{s}_i$ yields a loss no worse than s_i, we replace s_i with $\tilde{s}_i$; otherwise, we keep s_i unchanged. Repeating this process for all $i = 1, \ldots, N$ completes one generation of DE. We then iterate mutation, crossover, and selection until a stopping criterion is met (e.g., reaching a maximum number of generations).

Reiterating the process above, each generation of DE alternates between generating trial vectors via mutation and crossover, and then selecting the best candidates according to the loss function.In our implementation for adversarial suffix optimization, we additionally incorporate an early-stop criterion based on a patience counter to halt each suffix stage once no improvement is observed for a predefined number of generations. Here, we use a suffix token sequence as an illustrative example to demonstrate how DE operates on discrete token sequences. To consolidate the description above, Algorithm 1 presents the complete pseudo code for the DE procedure with early stopping when optimizing a suffix token sequence. Now, to illustrate suffix-level DE in action, we give a brief example. We initialize the example by taking the query *"What is the capital of France?"* and documents X_1 ("Paris is the capital and most populous city of France."), X_2 ("Berlin is the capital of Germany."), and X_3 ("Madrid is the capital of Spain."). Our goal is to promote X_3 by appending one token from

$$V = \{\texttt{[unused186]},\ \texttt{wash},\ \texttt{candidate}\}.$$

We run DE with $n_{\max} = 1$, $N = 3$, $F = 0.5$, CR $= 0.5$, initializing suffixes $(s_1, s_2, s_3) = (\texttt{[unused186]},\ \texttt{wash},\ \texttt{candidate})$ and

Algorithm 1: DE with Early Stop for Optimizing a Suffix Token Sequence

Input: Population size N (number of candidate suffixes); generations per stage G (max iterations per suffix length);
crossover rate CR (probability of crossover per token);
mutation factor F (scaling for mutant vector);
max suffix length $n_{\max}$; plateau patience T (number of generations without improvement before early stop)
Output: Optimized adversarial suffix s^* (token sequence)

```
s* ← []                    // start with empty suffix
for L ← 1 to n_max do
    initialise population {s_i}_{i=1}^N by right-padding s* to length L
    best ← ∞, pat ← 0
    for g ← 1 to G do
        for i ← 1 to N do
            pick distinct indices a, b, c ≠ i
            m ← MutXover(s_a, s_b, s_c, CR, F)
            if L(m) < L(s_i) then
                s_i ← m
        L_min ← min_i L(s_i)
        if L_min < best then
            best ← L_min, pat ← 0
        else
            pat ← pat + 1
        if best = 0 or pat ≥ T then
            break
    s* ← arg min_i L(s_i)
    if L(s*) = 0 then
        break
return s*
```

extracting their 768-dimensional embeddings using the BERT encoder. Next, we perform mutation by randomly selecting three parent tokens—`phantom`, ♡, and `token 1634`—from the full tokenizer list. Denoting their embeddings by $e_{p_a}, e_{p_b}, e_{p_c}$, the first three dimensions are:

$$e_{p_a}^{1-3} = [-0.045,\ -0.080,\ -0.005],$$
$$e_{p_b}^{1-3} = [-0.011,\ -0.044,\ 0.013],$$
$$e_{p_c}^{1-3} = [-0.022,\ -0.082,\ -0.010].$$

We then compute the donor vector

$$m = e_{p_a} + 0.5\,(e_{p_b} - e_{p_c}),$$

yielding $m^{1-3} = [-0.0398,\ -0.0619,\ 0.0062]$. An L_2 nearest-neighbor search over V selects `phantom`. Lastly, for the individual originally "wash," a draw $r = 0.37 < 0.5$ yields trial $\tilde{s}$ = `phantom`; computing hinge loss

$$L(s) = \max\{0,\ \max_{j\neq 3} \mathrm{sim}(q \,\|\, s, X_j) - \mathrm{sim}(q \,\|\, s, X_3)\},$$

we obtain $L(\texttt{wash}) \approx 0.4$ versus $L(\texttt{phantom}) \approx 0.2$. Since the trial loss is lower, we update $s_2 \leftarrow$ `phantom`. Repeating until $L(s) = 0$ produces a suffix that guarantees X_3 becomes the top-1 result.

Stopping criteria. We combine two complementary rules:

- **Success rule:** $\mathbb{I}_{\mathrm{succ}}(s) = \left[\mathrm{rank}(e_{q\|s}, e_t) \le k \wedge \mathrm{rank}(e_{q\|s}, e_q) > k\right]$. If $\mathbb{I}_{\mathrm{succ}}(s) = 1$ for any population member, evolution halts immediately.
- **Plateau rule:** with $B_g = \min_i L^{(g)}(s_i)$, stop when $B_g - B_{g-T} = 0$, i.e. no improvement for T consecutive generations.

Our DE framework includes three complementary variants that both rely on the same success and plateau rules but optimize for different objectives. The DE_FIXED_STOP variant applies these rules to a single, predetermined suffix length and is tuned for speed—it terminates as soon as the success rule is satisfied or the search stagnates, yielding a solution in the fewest iterations. In contrast, DE_SEQ_STOP and DE_SEQ incrementally increases the suffix length in stages, invoking the stopping criteria at each stage in plateau rule and success rule individually; this approach may require more iterations but returns the shortest possible adversarial suffix, minimizing the number of tokens added. Section 4 shows that our hybrid early-stopping strategy cuts average query cost by ≈ 40% while matching the attack success of vanilla DE.

4 EVALUATION

4.1 Dataset and Setting

To theoretically validate our attack effectiveness, we use the MS MARCO passage retrieval dataset [1]. Then, we evaluate our attacks on more BEIR benchmarks—**FiQA-2018** (financial QA) [2], **FEVER** (fact extraction and verification) [4], and **SciFact** (scientific fact verification) [3] to better compare to other existing methods. The BEIR framework [6] is used to standardize evaluation across these datasets. For each dataset, we randomly sample a subset of 1,000 documents and 100 queries from the official corpus and queries or claims splits. Each query has one or more annotated relevant passages X_u. For the adversarial attack we additionally choose, for each query, a single "target" wrong passage X_t (either drawn uniformly at random from the non-relevant documents, or selected as a topically confusable distractor). Data and experiment results are available at https://github.com/pen9rum/Rag_attack_DeRag [34].

For the dense retriever, all documents and queries are encoded with a BERT-base-uncased encoder [18] (the same model used for adversarial patch generation) into 768-dimensional CLS embeddings, comprising over 110 million parameters. Retrieval is performed by computing cosine similarity over these embeddings and returning the top-k passages (with k=1,10,20). For sparse retrievers, we index the same corpus with the BM25 algorithm, which computes relevance scores using term frequency–inverse document frequency (TF–IDF) and document-length normalization, and likewise retrieve the top-k passages based on these BM25 scores.

Evaluation metrics. RAG pipelines can be subverted by adversarial prompts that promote an incorrect target passage X_t into early ranks. To evaluate both the efficacy and cost of such attacks over 100 queries, we employ six key measures. We assess each method over 100 queries using a suite of retrieval-oriented and

optimization-oriented measures. First, *Success@K* captures the proportion of queries for which the true target document X_t appears within the top-K retrieved results directly. *Average tokens* records the mean length of the adversarial suffix, reflecting the minimal perturbation size required. *Average iterations* counts the mean number of optimization steps , noticing that it reflects the actual number of iterations performed by the DE algorithm, rather than the hyperparameter-defined iteration limit to capture computational effort. ΔMRR_t measures the change in reciprocal rank $1/r$ of X_t before and after the attack, isolating the shift in ranking. $\Delta\text{nDCG@20}_t$ computes the difference in normalized discounted cumulative gain at cutoff 20 when treating X_t as the sole relevant item, assessing retrieval quality degradation. $\Delta\cos = \cos(e_{q\|s}, e_t) - \cos(e_s, e_t)$ captures the shift in semantic similarity between the adversarially augmented query and the target; a smaller $|\Delta\cos|$ indicates a subtler semantic perturbation, which is likely more difficult for detectors to identify.

4.2 Prompt perturbation results on Sparse Retriever Attack

To clarify, our method is effective across multiple retrieval approaches, so we also conducted experiments on sparse retrieval. In Table 1, we compare our DE-based attacks against PRADA—a state-of-the-art sparse-retrieval black-box method [16]. Across four sparse-retrieval benchmarks, our sequential variant DE_SEQ_STOP achieves the highest Top-10 and Top-20 success rates while perturbing only an average of 2–3 tokens per query. By employing early stopping, the suffixes remain extremely compact without compromising their effectiveness. When minimizing query length is crucial, our fixed-length variant DE_FIXED_STOP matches or outperforms PRADA at all thresholds and converges in fewer iterations. These results show that our DE framework provides superior control over both suffix length and query cost under sparse retrieval.

Table 1: Tail-Patch performance on four datasets (1 000-document subset, 100 queries) on sparse retrievers.

Dataset	N	Method	Succ@1	Succ@10	Succ@20	AvgTok	AvgIter	ΔMRR	ΔnDCG
SciFact	1000	DE_seq_stop	0.250	0.890	0.970	3.133	3378	0.262	0.413
		DE_fixed_stop	0.190	0.850	0.930	5.000	1329	0.265	0.412
		PRADA	0.010	0.980	1.000	2.253	1802	0.221	0.398
FiQA	1000	DE_seq_stop	0.270	0.810	0.890	3.387	3776	0.279	0.407
		DE_fixed_stop	0.190	0.710	0.850	5.000	1497	0.240	0.371
		PRADA	0.030	0.990	1.000	2.163	1369	0.285	0.448
FEVER	1000	DE_seq_stop	0.620	0.960	0.970	2.697	2182	0.431	0.553
		DE_fixed_stop	0.550	0.930	0.960	5.000	975	0.422	0.546
		PRADA	0.070	0.960	0.910	2.137	925	0.275	0.429
MS MARCO	1000	DE_seq_stop	0.710	0.990	0.990	2.323	1818	0.469	0.588
		DE_fixed_stop	0.600	0.980	0.980	5.000	829	0.438	0.564
		PRADA	0.020	0.990	0.970	2.070	579	0.228	0.401

4.3 Prompt-perturbation results on Dense Retriever Attack

Dense retrievers are more robust than sparse retrievers because they encode both queries and documents into a continuous semantic space, rather than relying on exact term matches and term-frequency signals. Therefore, we mainly focus on attack comparison in dense retrievers. We compare our method against the gradient-guided white-box baseline GGPP [23] to demonstrate that effective dense-retrieval attacks need not rely on white-box access. Table 2 compares two strong differential-evolution variants—DE_SEQ_STOP and DE_FIXED_STOP—against the gradient-guided baseline GGPP and a random suffix on four BEIR subsets. For each method we report top-K success rates (*Succ@K*) and five cost/quality metrics introduced in Section 4. In real Retrieval-Augmented Generation (RAG) pipelines k is typically between 5 and 20; Lewis et al. [25] evaluate with $k \in \{5, 10\}$. To cover edge cases we also include $k = 1$. Appendix C illustrates representative adversarial suffixes produced by our attacks.

Table 2: Tail-Patch performance on four datasets (1 000-document subset, 100 queries) on dense retrievers.

Dataset	N	Example	Method	Succ@1	Succ@10	Succ@20	AvgTok	AvgIter
SciFact	1000	C.1	DE_seq_stop	0.198	0.573	0.739	2.34	2933
			DE_fixed_stop	0.104	0.469	0.641	5.00	1256
			ggpp	0.146	0.458	0.565	5.00	—
			Random	0.000	0.115	0.239	5.00	32
FiQA	1000	C.2	DE_seq_stop	0.126	0.520	0.546	2.63	3403
			DE_fixed_stop	0.063	0.429	0.515	5.00	1107
			ggpp	0.105	0.480	0.546	5.00	—
			Random	0.011	0.122	0.152	5.00	43
FEVER	1000	C.3	DE_seq_stop	0.185	0.515	0.643	2.76	3336
			DE_fixed_stop	0.122	0.515	0.541	5.00	993
			ggpp	0.122	0.545	0.408	5.00	—
			Random	0.010	0.152	0.235	5.00	44
MS MARCO	1000	C.4	DE_seq_stop	0.570	1.000	1.000	1.32	496
			DE_fixed_stop	0.660	0.970	1.000	5.00	361
			ggpp	0.830	1.000	1.000	5.00	—
			Random	0.340	0.760	0.710	5.00	20

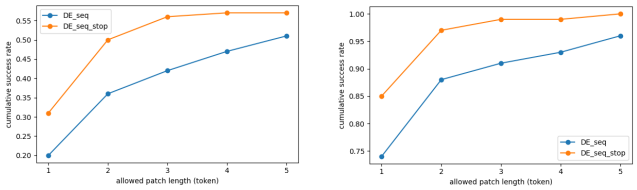
Table 2 reports the Tail-Patch attack success rates, token budgets, iteration counts, and quality-shift metrics on four datasets. Table 7 (in Appendix) summarizes the average ΔMRR, ΔnDCG and $\Delta\cos$ at cutoffs $K \in \{1, 10, 20\}$ in different contrast.

Generally speaking, DE_SEQ_STOP achieves the highest success rate at all datasets, especially reaching up to 0.739 on SciFact dataset while modifying only 2–3 tokens. Also, we notice that under MS MARCO dataset, the success rate is relatively higher than other datasets. The possible reason is that the dataset's development corpus is relatively small and highly redundant, so even very compact perturbations,can successfully attacked and perturbed the ranking result, resulting in strong performance for all methods with far fewer iterations and tokens. Across all benchmarks, the DE-based methods produce smaller average cosine shifts than GGPP, indicating reduced semantic drift in the adversarial suffix and having a relatively better results across four benchmarks.

4.3.1 Iterations–Tokens Trade-off. As shown in Table 2, there is a clear trade-off between the length of the adversarial suffix and the number of optimization iterations required by each method. The thousands of iterations in the table use a detailed metric—counting by treating every DE mechanism step as one iteration. If we instead count a whole DE generation cycle as one step, each attack needs only dozens to hundreds of iterations.Thus, the effective computational cost is far smaller than the raw iteration counts might suggest. The sequential DE variant (DE_SEQ_STOP) consistently finds very compact suffixes—on average only about 2–3 tokens long—but does so at the expense of a large number of iterations (for example, 2.34 tokens and 2 933 iterations on SciFact, and 1.32

tokens and 496 iterations on MS MARCO). By contrast, the fixed-length DE variant (DE_FIXED_STOP), which always uses the full five-token budget, roughly halves the iteration count (e.g., 5 tokens and 1 256 iterations on SciFact, and 5 tokens and 361 iterations on MS MARCO) while still achieving comparable success rates. Finally, the gradient-guided baseline (GGPP) converges in only fewer gradient steps but likewise consumes the maximum five tokens. Consequently, if minimizing the visible perturbation (suffix length) is the primary goal, DE_SEQ_STOP is the preferable choice; if reducing the number of model queries (iterations) is more important, DE_FIXED_STOP offers a better balance; and when gradient access is available, GGPP delivers the fastest convergence at the cost of the largest token footprint.Further details and the full data underlying the quality shift analysis are provided in Appendix B.

4.3.2 Advantage of Early Stopping. Figure 2 clearly shows that the early-stopping variant (DE_SEQ_STOP) outperforms the vanilla progressive DE (DE_SEQ) at every suffix length. At Top-1 (Fig. 2a), DE_SEQ_STOP reaches a 50% success rate with only 2 tokens, whereas DE_SEQ achieves just 36% at that length and only approaches 51% by 5 tokens. At Top-10 (Fig. 2b), the early-stopping method attains 97% success by 2 tokens and saturates above 99% by 3 tokens, while DE_SEQ needs 3–4 tokens to exceed 90%. These results demonstrate that early stopping not only increases cumulative success for a fixed budget but also identifies the minimal suffix length required—thereby saving unnecessary queries and limiting token perturbations.



(a) Cumulative success (Top-1). **(b) Cumulative success (Top-10).**

Figure 2: Cumulative success rates for `DE_seq` (blue) vs. `DE_seq_stop` (orange) as a function of allowed suffix length on MS MARCO dataset.

4.3.3 Prefix and Suffix mutual Attacks. We additionally ran DE on both prefix and suffix from 1 to 5 tokens, attacking on four datasets to examine whether the two strategies are complementary or not. To consider the best potential, we set the task on targeting rank 1. Table 3 reports the number of queries out of 100 that each strategy succeeds .

Table 3: Prefix–suffix complementarity under SUCCESS@1.

Dataset	Suffix only	Prefix only	Both succeed	Both fail	Comp. rate	Either succeed
FIQA	14	14	17	55	28	45
FEVER	5	7	24	64	12	36
SciFact	11	15	15	59	26	41

We further tested a monotonic suffix length schedule and a naive cosine-similarity objective; neither improved Success@K nor ranking displacement (Appendix D and E). These results show that if we want to achieve more successes, it is more efficient to insert tokens at different positions. Positional diversity in token-level attacks is valuable and gives great performance boost on generating adversarial prompt.

4.4 Adversarial Query Crafting: Detection Bypass and Readability

4.4.1 Evasion Performance against a BERT-Based Detector. The numerical detector scores, along with the corresponding distributions, are deferred to Appendix F. Specifically, Table 13 summarizes the perplexity and BERT-CLS scores, while Fig. 5 and Fig. 6 visualize their distributions respectively, where PPL stands for perplexity. On average, the PPL scores are higher in two datasets and relatively lower in the other two. Notably, the MS MARCO dataset contains an extreme outlier query, which significantly elevates its overall PPL score. In contrast, the BERT-CLS–based attack probability remains consistently similar across datasets, suggesting that, although manual inspection may still reveal differences between original and adversarial queries, these perturbations can deceive certain detection models. Future research should develop targeted attacks against emerging defense mechanisms.

We further evaluate our adversarial examples against the RoBERTa detector [26] to measure the detection rate. In total, we collected 783 samples (483 adversarial and 300 original clean queries) and assessed the detector's ability to distinguish between them. Table 4 summarizes detection metrics under various FPR targets.

Table 4: Detection performance of RoBERTa detector under different threshold.

Target FPR	Actual FPR	Threshold	Precision	Recall	F1
0.5%	0.33%	0.952	0.00%	0.00%	0.00%
1.0%	0.33%	0.952	0.00%	0.00%	0.00%
2.0%	2.00%	0.874	33.33%	0.62%	1.22%

The overall detection performance is poor: at a standard target FPR of 0.5%, the detector achieves an AUROC of 0.2023 and an AUPRC of 0.4665. This indicates that prompt-injection examples generated by DE attacks are nearly indistinguishable from benign inputs since it only required few tokens, therefore demonstrates that our token-level perturbations effectively evade detection even at stringent false-positive rates, providing strong evidence that DE prompt injection attacks can deceive detectors and are difficult to distinguish.

4.4.2 Human Reading Availability Strategy. To enhance human readability without degrading attack success rates, we propose a dynamic candidate-pool construction strategy. For each query, we first encode it with BERT on CPU to obtain its CLS vector and compute cosine similarities against a pre-encoded matrix of document CLS vectors; the top N most similar documents are used solely to form the contrastive pool in our differential-evolution attack. Independently, to construct the token candidate pool, we mask the

Table 5: Answer quality degradation under adversarial retrieval. "Top-10 only" means target chunk elevated into Top-10 but not rank 1. "Fail" = target chunk not inserted into Top-10. "Average" is the weighted per-query best attacked outcome (Top-1 → Top-10 only → Fail) using the empirical rates. Percent changes are relative to each dataset's baseline.

Dataset	Group	Rate	EM	F1	ROUGE-L	BERTScore
SQuAD (500)	**Baseline**	–	**0.322**	**0.388**	**0.396**	**0.900**
	Top-1	11.4%	0.053 (↓83.5%)	0.056 (↓85.6%)	0.056 (↓85.9%)	0.816 (↓9.3%)
	Top-10 only	42.0%	0.162 (↓49.7%)	0.190 (↓51.0%)	0.191 (↓51.8%)	0.860 (↓4.4%)
	Fail	46.6%	0.348 (↑8.1%)	0.415 (↑7.0%)	0.430 (↑8.6%)	0.897 (↓0.3%)
	Average	100%	0.236 (↓26.7%)	0.279 (↓28.1%)	0.287 (↓27.5%)	0.872 (↓3.1%)
NQ-Open (500)	**Baseline**	–	**0.782**	**0.8236**	**0.8616**	**0.9603**
	Top-1	8.4%	0.6190 (↓20.8%)	0.6565 (↓20.3%)	0.6520 (↓24.3%)	0.9218 (↓4.0%)
	Top-10 only	91.4%	0.6696 (↓14.4%)	0.7117 (↓13.6%)	0.7485 (↓13.1%)	0.9365 (↓2.5%)
	Fail	0.2%	1.0000 (↑27.9%)	1.0000 (↑21.4%)	1.0000 (↑16.1%)	1.0000 (↑4.1%)
	Average	100%	0.6660 (↓14.8%)	0.7076 (↓14.1%)	0.7409 (↓14.0%)	0.9354 (↓2.6%)

last TAIL_L tokens of the query, pass the masked sequence through the MLM head, average the predicted softmax probabilities over those TAIL_L positions, and select the top k vocabulary tokens as mask-fill candidates. We perform an ablation over multiple pooling size shown below. Table 6 shows that retrieval effectiveness ($K = 1$ and ΔMRR) remains stable across settings, and Table 14 (Appendix G) shows that pool construction, query optimization, and overall attack time remain low.

Table 6: Rank@1 success rate and MRR under various pool sizes.

Pool size	Fever		FiQA		SciFact	
	$K = 1$	ΔMRR	$K = 1$	ΔMRR	$K = 1$	ΔMRR
500	0.19	0.277	0.24	0.345	0.24	0.348
1 000	0.19	0.277	0.28	0.373	0.26	0.384
2 000	0.22	0.307	0.24	0.358	0.22	0.358
5 000	0.24	0.345	0.33	0.420	0.30	0.411
10 000	0.34	0.424	0.34	0.438	0.33	0.427
20 000	0.38	0.457	0.27	0.377	0.27	0.379
30 522	0.29	0.391	0.32	0.412	0.25	0.351

To quantify readability gains, we report the average MLM negative log-likelihood (NLL), a proxy for fluency, in Appendix H.

4.5 Downstream Answer Quality Degradation Evaluation

We evaluate a two-phase adversarial suffix attack on a dense-retrieval RAG pipeline to measure degradation of downstream answer quality. The attack is staged: first we drive a chosen (irrelevant) target chunk into the Top-10 so that it can influence the generator's context window; second, conditional on that success, we further optimize to promote the target to rank 1. We select two QA benchmarks: SQuAD [5] and NQ-Open (500 training queries). SQuAD supplies short span-style factoid answers drawn from comparatively tight passages, and NQ-Open contains broader and noisier web-sourced evidence, probing robustness under a larger semantic space. For each query we first run a baseline generation using the original Top-10 retrieved chunks. We then optimize a 5-token adversarial suffix (Differential Evolution, up to 120 iterations) to insert an unrelated target chunk into the Top-10 and, if successful, to push it to rank 1. Outcomes are stratified into Top-1 success, Top-10-only success, and Fail , which stands for the target never reaches the Top-10. We then compare answer quality across these outcome strata to quantify semantic degradation.

Table 5 shows persistent degradation once an adversarial chunk occupies retrieval slots. On SQuAD, a substantial portion of attempts did not elevate the target all the way to rank1 because we allotted fewer optimization iterations, yet the successful insertions still drove large semantic quality losses. On Natural Questions we used more iterations, achieved higher insertion success, and again observed pronounced declines in answer fidelity. The pattern indicates that even realistic, heterogeneous questions are vulnerable and that reliance on protecting only the very top rank is inadequate; limiting adversarial occupancy at any position within the narrow retrieval window is essential to preserve downstream answer quality.

5 CONCLUSION

We introduce DeRAG, a novel black-box adversarial attack framework targeting Retrieval-Augmented Generation (RAG) systems via prompt suffix perturbations. DeRAG formulates the attack as a discrete optimization problem and leverages Differential Evolution with custom stopping rules to efficiently discover suffixes that misdirect retrieval to incorrect documents. Experiments show that DeRAG matches or even outperforms gradient-based and other existing methods, while using fewer tokens and inducing smaller embedding shifts across both dense and sparse retrievers, all while maintaining a moderate level of anti-detection ability. We highlight the importance of factors like population diversity and suffix length, and validate early-stopping strategies that cut query costs without reducing effectiveness. We further evaluate the downstream results in real-world scenarios. DeRAG not only reveals critical RAG vulnerabilities but also informs future defenses such as prompt precision, embedding regularization, and anomaly detection—ultimately advancing robust, trustworthy AI.

REFERENCES

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A Human Generated Machine Reading Comprehension Dataset. *arXiv preprint arXiv:1611.09268*, 2018. https://arxiv.org/abs/1611.09268.

[2] Macedo Maia, Siegfried Handschuh, Andre Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. WWW'18 open challenge: Financial opinion mining and question answering. In *Companion of the The Web Conference 2018 (WWW '18 Companion)*, pages 1941–1942, Lyon, France, April 2018. International World Wide Web Conferences Steering Committee.

[3] Daniel Wadden, Rishi Bommasani, Russell Kaplan, Swabha Swayamdipta, and Nanyun Peng. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of EMNLP*, pages —, 2020.

[4] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: A large-scale dataset for fact extraction and VERification. In *Proc. of NAACL-HLT*, pages 809–819, New Orleans, LA, USA, 2018. https://aclanthology.org/N18-1074/.

[5] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proc. of EMNLP*, pages 2383–2392, Austin, TX, USA, 2016. https://aclanthology.org/D16-1264/.

[6] Nandan Thakur, Joel Mackenzie, Hasan Sajjad, Ziyu Yao, and Preslav Nakov. BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models. *arXiv preprint arXiv:2104.08663*, 2021. https://arxiv.org/abs/2104.08663.

[7] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. https://aclanthology.org/Q19-1026/.

[8] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[9] Mohamed Alsayed Elaziz, Ahmed Ali Ismail, Ashraf M. Salem, and Ahmed E. Hassanien. Differential Evolution: A Recent Review Based on State-of-the-Art Works. *Alexandria Engineering Journal*, 60(4):3595–3615, 2021.

[10] Jiarui Li, Ye Yuan, and Zehua Zhang. Enhancing LLM Factual Accuracy with RAG to Counter Hallucinations: A Case Study on Domain-Specific Queries in Private Knowledge-Bases. *arXiv preprint arXiv:2403.10446*, 2024. https://arxiv.org/abs/2403.10446.

[11] Xuying Li, Zhuo Li, Yuji Kosuga, Yasuhiro Yoshida, and Victor Bian. Targeting the Core: A Simple and Effective Method to Attack RAG-based Agents via Direct LLM Manipulation. *arXiv preprint arXiv:2412.04415*, 2024. https://arxiv.org/abs/2412.04415.

[12] Jiaqi Xue, Mengxin Zheng, Yebowen Hu, Fei Liu, Xun Chen, and Qian Lou. BadRAG: Identifying Vulnerabilities in Retrieval Augmented Generation of Large Language Models. *arXiv preprint arXiv:2406.00083*, 2024. https://arxiv.org/abs/2406.00083.

[13] Gabriel Alon and Michael Kamfonas. Detecting Language Model Attacks with Perplexity. *arXiv preprint arXiv:2308.14132*, 2023. https://arxiv.org/abs/2308.14132.

[14] Jinhwa Kim, Ali Derakhshan, and Ian Harris. Robust Safety Classifier Against Jailbreaking Attacks: Adversarial Prompt Shield. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 159–170, Mexico City, Mexico, June 2024. Association for Computational Linguistics. https://aclanthology.org/2024.woah-1.12.

[15] Runqi Sui. CtrlRAG: Black-box Adversarial Attacks Based on Masked Language Models in Retrieval-Augmented Language Generation. *arXiv preprint arXiv:2503.06950*, 2025. https://arxiv.org/abs/2503.06950.

[16] Chen Wu and Ruqing Zhang. PRADA: Practical Black-box Adversarial Attacks against Neural Ranking Models. *ACM Transactions on Information Systems*, 41 (Dec. 16, 2022). 10.1145/3576923.

[17] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One Pixel Attack for Fooling Deep Neural Networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. https://doi.org/10.1109/TEVC.2019.2890858.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018. http://arxiv.org/abs/1810.04805.

[19] Fábio Perez and Ian Ribeiro. Ignore Previous Prompt: Attack Techniques for Language Models. *arXiv preprint arXiv:2211.09527*, 2022. https://arxiv.org/abs/2211.09527.

[20] Xiaogeng Liu, Zhiyuan Yu, Yizhe Zhang, Ning Zhang, and Chaowei Xiao. Automatic and Universal Prompt Injection Attacks Against Large Language Models. *arXiv preprint arXiv:2403.04957*, 2024. https://arxiv.org/abs/2403.04957.

[21] Chong Zhang, Mingyu Jin, Qinkai Yu, Chengzhi Liu, Haochen Xue, and Xiaobo Jin. Goal-guided Generative Prompt Injection Attack on Large Language Models. *arXiv preprint arXiv:2404.07234*, 2024. https://arxiv.org/abs/2404.07234.

[22] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking Black Box Large Language Models in Twenty Queries. *arXiv preprint arXiv:2310.08419*, 2024. https://arxiv.org/abs/2310.08419.

[23] Zhibo Hu, Chen Wang, Yanfeng Shu, Hye-young Paik, and Liming Zhu. Prompt Perturbation in Retrieval-Augmented Generation based Large Language Models. *arXiv preprint arXiv:2402.07179*, 2024. https://arxiv.org/abs/2402.07179.

[24] Andy Zhou and Ron Arel. TEMPEST: Multi-Turn Jailbreaking of Large Language Models with Tree Search. In *Proceedings of the ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. https://openreview.net/forum?id=rDC2UVdB0t.

[25] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *arXiv preprint arXiv:2005.11401*, 2021. https://arxiv.org/abs/2005.11401.
https://arxiv.org/abs/2011.03901.

[26] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and others. Release strategies and the social impacts of language models. In *arXiv preprint arXiv:1908.09203*, pages —, 2019.

[27] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. EvoPrompt: Connecting LLMs with Evolutionary Algorithms Yields Powerful Prompt Optimizers. *arXiv preprint arXiv:2309.08532*, 2025. https://arxiv.org/abs/2309.08532.

[28] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Black-box Adversarial Sample Generation Based on Differential Evolution. *arXiv preprint arXiv:2007.15310*, 2020. https://arxiv.org/abs/2007.15310.

[29] Shizhe Diao, Zhichao Huang, Ruijia Xu, Xuechun Li, Yong Lin, Xiao Zhou, and Tong Zhang. Black-Box Prompt Learning for Pre-trained Language Models. *Transactions on Machine Learning Research*, to appear, 2023. https://openreview.net/forum?id=IvsGP7xRvm.

[30] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative Representational Instruction Tuning. *arXiv preprint arXiv:2402.09906*, 2024. https://arxiv.org/abs/2402.09906.

[31] Luke Merrick, Danmei Xu, Gaurav Nuti, and Daniel Campos. Arctic-Embed: Scalable, Efficient, and Accurate Text Embedding Models. *arXiv preprint arXiv:2405.05374*, 2024. https://arxiv.org/abs/2405.05374.

[32] Zhengmian Hu, Yichao Lu, Xuejun Liao, Jianfeng Gao, and Maxine Eskenazi. Token-Level Adversarial Prompt Detection Based on Perplexity Measures and Contextual Information. *arXiv preprint arXiv:2311.11509*, 2023. https://arxiv.org/abs/2311.11509.

[33] Rishi Jha, Collin Zhang, Vitaly Shmatikov, and John X. Morris. Harnessing the Universal Geometry of Embeddings. *arXiv preprint arXiv:2505.12540*, 2025. https://arxiv.org/abs/2505.12540.

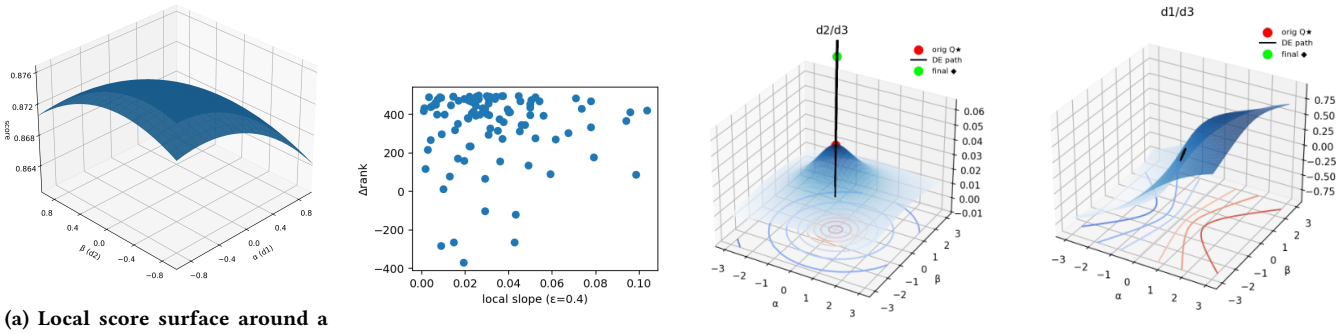
[34] pen9rum, *Rag_attack_DeRag: Differential Evolution Attacks on Retrieval-Augmented Generation*, GitHub repository, https://github.com/pen9rum/Rag_attack_DeRag, accessed May 25, 2025.

[35] Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(1):333–389, 2009. https://doi.org/10.1561/1500000019.

[36] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online, 2020. https://aclanthology.org/2020.emnlp-main.550/.

A EFFECTIVENESS ON DIFFERENTIAL EVOLUTION ATTACKS

Figure 3 shows the results. Over 100 queries under the condition of $|C| = 1000$ passages and twelve Gaussian perturbations, the mean slope is only 0.0328. Figure b plots Δrank against the local slope $\lambda(\mathbf{q})$, yielding a Pearson correlation of $r = 0.016$ (two-tailed $p = 0.871$), proves that isotropic smoothness provides virtually no insight into how effectively an adversary can alter the document's rank. We then project the score of a representative query onto three orthogonal planes using $\mathbf{d}_1$, $\mathbf{d}_2$, and $\mathbf{d}_3$. Figures c and d confirm that the score varies essentially only along $\mathbf{d}_1$, and the DE trajectory (black) climbs the ridge from the red star (original query) to the green diamond (final adversarial query), showing that DE exploits this structure with few tokens. Hence, retrievers remain vulnerable to concise suffix attacks.



(a) Local score surface around a query vector. **(b) Δrank vs. local slope ($\varepsilon = 0.4$)** **(c) $(\mathbf{d}_2, \mathbf{d}_3)$ plane** **(d) $(\mathbf{d}_1, \mathbf{d}_3)$ plane**

Figure 3: (a) Local score surface around a query vector. (b) Isotropic smoothness is nearly uncorrelated with attack efficacy. (c) Local score surface on the $(\mathbf{d}_2, \mathbf{d}_3)$ plane, emphasizing monotonicity along $\mathbf{d}_1$. (d) Local score surface on the $(\mathbf{d}_1, \mathbf{d}_3)$ plane.

B QUALITY SHIFT RESULT

Table 7 compares three attacks (DE_SEQ_STOP, DE_FIXED_STOP, GGPP) across Top-K settings ($K \in \{1, 10, 20\}$) using ΔnDCG and ΔMRR drops and cosine shift $\Delta\cos$ as our evaluation metrics. At $K = 1$, GGPP often attains the largest immediate ranking loss , but shows a larger (often positive) $\Delta\cos$, whereas DE_SEQ_STOP attains comparable ΔnDCG/ΔMRR with a near-zero shift, indicating a much lower semantic perturbation. At $K = 10$ and 20, under more general (larger K) settings, DE_SEQ_STOP surpasses the gradient-guided GGPP (and the fixed variant) by matching or improving ΔnDCG/ΔMRR while keeping $\Delta\cos$ closest to zero. Therefore, DE_SEQ_STOP delivers the best balance—maintaining low semantic movement while producing equal or greater retrieval quality loss—making it the most effective and stealthy of the compared attacks.

Table 7: Quality shift (Δ) comparison of DE_SEQ_STOP, FIXED, and GGPP at $K \in \{1, 10, 20\}$ on dense retriever.

Dataset	Metric	$K = 1$			$K = 10$			$K = 20$		
		Seq	Fixed	GGPP	Seq	Fixed	GGPP	Seq	Fixed	GGPP
SciFact	ΔnDCG	0.188	0.267	0.226	0.196	0.212	0.203	0.198	0.167	0.153
	ΔMRR	0.193	0.212	0.172	0.090	0.115	0.112	0.068	0.060	0.060
	$\Delta\cos$	0.002	0.038	−0.004	−0.008	−0.006	0.031	−0.006	−0.039	0.011
FiQA	ΔnDCG	0.113	0.201	0.147	0.186	0.176	0.201	0.141	0.145	0.158
	ΔMRR	0.114	0.158	0.108	0.092	0.082	0.101	0.043	0.061	0.068
	$\Delta\cos$	−0.003	0.008	−0.019	−0.010	−0.016	0.024	−0.015	−0.012	0.020
FEVER	ΔnDCG	0.181	0.203	0.237	0.110	0.110	0.140	0.181	0.176	0.140
	ΔMRR	0.181	0.168	0.183	0.057	0.049	0.090	0.093	0.098	0.090
	$\Delta\cos$	0.021	0.111	0.090	−0.005	−0.019	0.084	0.040	0.060	0.084
MS MARCO	ΔnDCG	0.722	0.646	0.585	0.597	0.646	0.607	0.474	0.555	0.607
	ΔMRR	0.699	0.597	0.534	0.532	0.597	0.563	0.386	0.494	0.563
	$\Delta\cos$	0.015	−0.027	−0.021	−0.052	−0.027	0.016	−0.048	−0.016	0.016

C ADVERSARIAL SUFFIX EXAMPLES (DIFFERENTIAL EVOLUTION)

C.1 Examples on FiQA Dataset

Table 8: Example DE prefix using user query and passages from the FiQA dataset, including suffix, original searched document, and retrieved (targeted) document.

Ex #	QUERY	Adversarial suffix	Original searched document	Retrieved (targeted) document
ex1	Historical stock prices: Where to find free / low cost data for offline analysis?	##systems williams truss ram exposing	Go to `http://finance.google.com`, search for the stock you want. When you are seeing the stock information, in the top legend you can click "Download to CSV".	It's a good point and I think you answered the question yourself: most IPOs don't have that download option yet. There's a real risk of stale data if you rely only on free sources.
ex2	How can all these countries owe so much money? Why & where did they borrow it from?	##– shannon wed 1995 liquid	"They borrowed it from the people, and typically to finance wars and military spending. For example, Wikipedia suggests that many Eurozone governments issue bonds directly into primary markets."	"Never understood why any country would want to be beholden to complete strangers that you would never hire to begin with. Government debt usually goes to banks, but they resell it into the secondary market."
ex3	Why are auto leases stubbornly strict about visa status and how to work around that?	copyright advocacy affordable applied	"Uh, you want to lease a car through a dealer? That is the worst possible way to obtain a car. Dealers love leases because it locks you into a contract. Try a local credit union—they often just need proof of residency, not citizenship."	"Also, the second amendment. Even if some kind of gun control measure were passed, sport shooting is never going to disappear—most people lease cars based on credit history, not visa status."
ex4	Should I re-allocate my portfolio now or let it balance out over time?	solving ##™ merry ##nett livery	"As you note, your question is inherently opinion-based. That said, if I were in your situation I would sell the stock that has a P/E above 30 and buy more of the index fund that tracks the S&P 500."	"Text blast is the new marketing. I think any company should try it right away. It is cheap and effective. Texting services cost $0.01 per SMS internationally but they boost open rates."
ex5	To pay off a student loan, should I save up a lump sum payoff payment or pay extra each month?	331 cameroon snorted rosenberg	"As a new graduate, aside from the fact that you seem to have the extra $193/mo to pay more towards your loan, we don't know if you'll keep that job. If your interest rate is 6.8%, any extra payment cuts principal every month, so pay extra monthly if you can."	"It wasn't a small hole in your suggestion. You were proposing a major law which already exists. The only anti-intellectual property you need is 'First sale doctrine'. It already covers reselling."

C.2 Examples on SciFact Dataset

Table 9: Example DE prefix using user query and passages from the SciFact dataset, including suffix, original searched document, and retrieved (targeted) document.

Ex #	QUERY	Adversarial suffix	Original searched document	Retrieved (targeted) document
ex1	The density of cytokine receptor bearing cells has no effect on the distance over which cytokines act.	converse behaviour excitedly syllables homeland	Immune cells communicate by exchanging cytokines to achieve a context-appropriate response, but the distances over which those cytokines diffuse remain disputed.	Neutrophil extracellular traps (NETs) are made of processed chromatin bound to granular and selected cytoplasmic protein, and these NETs can trap pathogens outside the cell.
ex2	CCL19 is a ligand for CCR7.	baltic ##chus ##ead wnba	Chemokines orchestrate immune cell trafficking by eliciting either directed or random migration and by activating integrins on leukocytes.	TNF expression of macrophages is under stringent translational control that depends on the p38 MAPK/MK2 pathway and the eIF4E phosphorylation.
ex3	A deficiency of vitamin B6 decreases blood levels of homocysteine.	## bal vocalist ## eager	BACKGROUND Homocysteine is a risk factor for coronary artery disease (CAD), although a causal relation remains to be proven.	Introduction: Among the inflammatory mediators involved in the pathogenesis of obesity, the cell adhesion molecules P-selectin and ICAM-1 are up-regulated.
ex4	Healthy volunteers exhibit rapid and transient increase of cellular ATP after being bolus-injected with fructose.	evangelist tumbling rodeo assignment beard	CONTEXT The mechanisms that drive progression from fatty liver to steatohepatitis and cirrhosis are unknown. In animal models, fructose injection can cause acute ATP depletion in hepatocytes.	We investigated whether or not administered leptin influences anxiety-like behavior in ob/ob mice. Repeated intraperitoneal injection elevated hippocampal CREB phosphorylation.
ex5	Gastric infection with *Helicobacter pylori* decreases risk of gastric cancer in humans.	reporter nicaragua cmll sgt ##idae	OBJECTIVE To investigate the association between gastric cancer and prior infection with *H. pylori*. DESIGN Case-control study in high-incidence region.	Neutrophil extracellular traps (NETs) are made of processed chromatin bound to granular and selected cytoplasmic protein, which may contribute to tissue damage during sepsis.

C.3 Examples on FEVER Dataset

Table 10: Example DE prefix using user query and passages from the FEVER dataset, including suffix, original searched document, and retrieved (targeted) document.

Ex #	QUERY	Adversarial suffix	Original searched document	Retrieved (targeted) document
ex1	Jensen Ackles was nominated for several Daytime Emmy Awards.	##↓ contribute ##ons drowning brave	Jensen Ross Ackles (born March 1, 1978) is an American actor and director. He is known for his roles in television, including Dean Winchester on "Supernatural." He received Daytime Emmy nominations for his work on "Days of Our Lives."	The discography of the English rock band Embrace. The band's highest charting single is "Nature's Law," which reached #2 on the UK Singles Chart in 2006.
ex2	Steve Buscemi the film "Trees Lounge"'s director.	##hul ##apes] ##erine holes	Steven Vincent Buscemi (/bismi/; born December 13, 1957) is an American actor and film director. Buscemi wrote and directed "Trees Lounge" (1996), which he also starred in. The film premiered at the Sundance Film Festival and won the Waldo Salt Screenwriting Award.	Kadapra is a village in Koipuram Panchayath of Pathanamthitta District, Kerala, India. It is known for the ancient Kadapra Shiva Temple and Iringhavala Kavu temple festivals.
ex3	The *Underworld* film series has grossed a total of $539 million.	staples reprinted appendix [unused42] alison	Underworld is a series of action horror films directed by Len Wiseman, Patrick Tatopoulos, Måns Mårlind, and Björn Stein. The first film (2003) grossed $95 million worldwide. The entire franchise has grossed over $539 million globally.	This article is about the particular significance of the year 1717 to Wales and its people. In 1717, the first Welsh-language newspaper was published, "Eurgrawn Cymraeg."
ex4	Aneurin Barnard's birth year is 1987.	chloride gp pretended pp dilapidated	Aneurin Barnard (born 8 May 1987) is a Welsh stage and screen actor. He is best known for roles in "Dunkirk" (2017) and "The Goldfinch" (2019). He won the BAFTA Cymru Award for Best Actor in 2014.	*Tumi Kar?* (2008) is a Bengali film directed by Gobinda Chandra Haldar and produced by Agni Productions. It stars Dipankar Dey and Moushumi Chatterjee.
ex5	Claudia Leitte is on "The Voice Brasil" as a coach and mentor.	ruth evelyn kurdistan treaty granite	Cláudia Cristina Leite Inácio Pedreira (/klawd lejti/; born 10 July 1980) is a Brazilian singer and songwriter. She served as a coach on "The Voice Brasil" from 2013 to 2016 and again in 2019.	Mukesh Dhirubhai Ambani (born 19 April 1957) is an Indian business magnate who is the chairman and managing director of Reliance Industries. He is currently the richest person in India.

C.4 Examples on MS MARCO Dataset

Table 11: Example DE prefix using user query and passages from the MS MARCO dataset, including suffix, original searched document, and retrieved (targeted) document.

Ex #	QUERY	Adversarial suffix	Original searched document	Retrieved (targeted) document
ex1	Collagen, the most abundant protein in the body, plays a role in which function?	rotary sprung awake prosecuted jessica	Collagen is the major structural protein in connective tissue and the most abundant protein in the human body. It is responsible for skin elasticity, bone strength, tendon flexibility, and wound healing.	Lisa Glasberg; Born: Lisa Glasberg June 6, 1956 (age 61); Other names: Lisa G; Alma mater: Hofstra University; Occupation: radio and TV personality at WFAN, Howard 101.
ex2	Where is blood coming from when it enters the tricuspid valve and passes into the right ventricle?	retailer staffed exchanging bread economical	As deoxygenated blood flows into the right atrium, it passes through the tricuspid valve and into the right ventricle, which then pumps it into the pulmonary artery.	I recommend this course. Thanks to: myalftraining.com online course (ALF Core Training), I have gained immense knowledge about Air Leakage Fundamentals.
ex3	Which of these terms is defined as words that are spelled the same but have different meanings and pronunciations?	honestly chen dowry belonging	Homographs are words that are spelled the same but differ in meaning, derivation, or pronunciation (e.g., "lead" the metal vs. "lead" to guide). Homophones are words that sound the same but have different spellings (e.g., "red" vs. "read").	The Reasons You Should Be Sipping Apple Cider Vinegar: I drink apple cider vinegar just like you described, Kim, with lemon and honey mixed in for breakfast to aid digestion.
ex4	What is the rhythmic wavelike muscular contraction of the esophagus that helps work the food down into the stomach called?	waited chateau officials male revealing	Digestive System – The esophagus is a muscular tube connecting from the throat to the stomach. The rhythmic wavelike contraction that propels food is called peristalsis.	Local news and events from Whitefish Bay, WI Patch. Latest headlines: Winter Storm Snowfall Totals, Plus More Snow On The Way.
ex5	What is the name of the condition characterized by disordered eating, lack of menstrual periods, and osteoporosis?	51st ##ße vacant kidnapped devastated	But for some girls, not balancing the needs of their bodies and their sports can have major consequences. Some girls who exercise excessively and restrict calories may develop the "female athlete triad," which includes eating disorders, amenorrhea, and osteoporosis.	Koenigsegg. Bugatti's Veyron. Lamborghini. Ferrari's Pininfarina. The names alone are exotic. Here are the most expensive hypercars in the world.

D INFLUENCE OF SUFFIX LENGTH

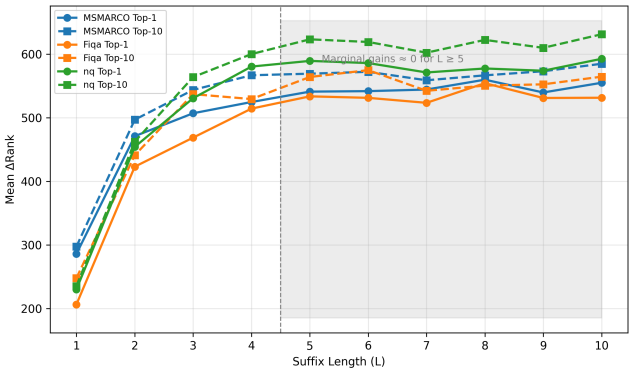
To investigate the influence of suffix length on adversarial effectiveness in retrieval-based systems, we conduct experiments across three benchmark datasets—MS MARCO, FiQA, and NQ [7]—under two ranking objectives: Top-1 and Top-10. Two key metrics are examined in this ablation study: the mean improvement in target document rank (*Mean* Δ*Rank*) and the incremental improvement obtained by increasing the suffix length by one token (*Marginal Gain*). These metrics are visualized in Figure 4: panel (a) plots Mean ΔRank and panel (b) plots Marginal Gain. In this experiment, we assume that the target corpus is the one with rank = 800, targeting the Top-1 or Top-10 result to make sure the maximum difference of the result stays the same and also ensure that there is sufficient variation to assess how the ΔRank efficient frontier diminishes.

Figure 4a shows that for both Top-1 and Top-10 settings, ΔRank rises sharply from $L = 1$ to $L = 4$. Beyond $L = 5$ (shaded region), further tokens yield negligible or negative gains, indicating a clear plateau in adversarial effectiveness.

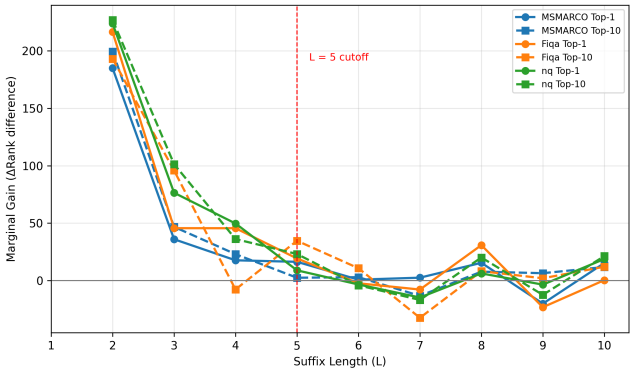
To provide a more granular understanding of this phenomenon, we compute the marginal gain for each suffix length, as illustrated in Figure 4b. Marginal gain is defined as the difference in mean ΔRank between suffix lengths L and $L - 1$, formally:

$$\text{MarginalGain}(L) = \text{Mean}\,\Delta\text{Rank}(L) - \text{Mean}\,\Delta\text{Rank}(L-1). \tag{9}$$

This measure directly quantifies the effectiveness of adding one additional token at each step. All six curves exhibit a steep decline in marginal gain after $L = 2$, reaching near zero by $L = 5$. Although the absolute ΔRank values vary across datasets, the curves share the same shape and exhibit diminishing returns beyond five tokens, implying that, under a fixed iteration budget, attackers should craft concise perturbations rather than adding tokens indiscriminately.



(a) Mean ΔRank vs. suffix length (Top-1 / Top-10).



(b) Marginal ΔRank gain per extra token (Top-1 / Top-10).

Figure 4: (a) Mean ΔRank as suffix length increases across MS MARCO, FiQA, and NQ. (b) Marginal gain in ΔRank per additional suffix token.

E AGGREGATED RESULTS ON LOSS COMPARISON

To evaluate the relative effectiveness of Hinge Loss and Cosine Loss in adversarial tail-patch attacks, we conducted experiments on four benchmarks. For every query, we selected as the adversarial target the relevant document that was ranked exactly 100th under the original cosine-similarity baseline. The goal of the attack was to promote this target document to the top of the ranking specifically, from rank 100 to rank 1 by appending a five-token adversarial suffix to each query. Differential Evolution was used to optimize the suffix, minimizing either a cosine-loss objective or a hinge-based ranking objective. Let the embedding of the adversarially modified query be denoted as $\boldsymbol{u} \in \mathbb{R}^d$, and the embedding of the target document as $\boldsymbol{v} \in \mathbb{R}^d$. Cosine Loss is defined as the negative direction of this similarity:

$$\mathcal{L}_{\cos}(\boldsymbol{u}, \boldsymbol{v}) = -\cos(\boldsymbol{u}, \boldsymbol{v}) = -\frac{\boldsymbol{u}^\top \boldsymbol{v}}{\|\boldsymbol{u}\| \, \|\boldsymbol{v}\|}. \tag{10}$$

While hinge loss is defined above. In optimization, minimizing $\mathcal{L}_{\cos}$ is equivalent to directly maximizing the alignment between the query and the target document in embedding space. However, as shown in As shown in Table 12, the cosine objective yields worse ranking performance than the hinge loss while incurring greater—and undesirable—semantic drift, and still does not reliably place the target document to the top rank.

Table 12: Aggregated Results for Hinge Loss versus Cosine Loss (baseline target rank = 100). All values are averaged over 100 queries per dataset.

Dataset	Cosine Δ cos	Hinge Δ cos	Cosine Δrank	Hinge Δrank	Cosine Iters	Hinge Iters	Cosine Succ. Rate	Hinge Succ. Rate
FEVER	0.0686	−0.0273	59.04	76.26	4425	3615	0.0100	0.2600
FiQA-2018	0.0422	−0.0092	63.92	86.99	4274	3743	0.0200	0.2300
MS MARCO	0.0634	0.0208	77.63	88.25	4346	3499	0.0700	0.2900
SciFact	0.0577	−0.0203	60.67	80.94	4624	3466	0.0200	0.3100

F DISTRIBUTION PLOTS

Table 13: Detector means for the original query (Q) and the attacked query with injection (Q‖S).

Dataset	PPL		CLS prob.	
	Q	Q‖S	Q	Q‖S
FEVER	1.451	1.491	0.407	0.402
FIQA	2.053	1.544	0.393	0.400
MSMARCO	42.57	1.620	0.399	0.399
SciFact	1.380	1.419	0.403	0.401

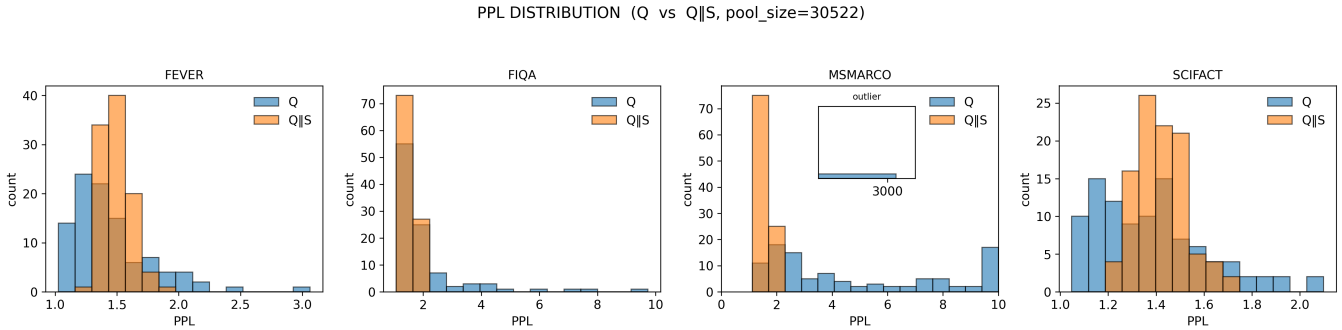


Figure 5: PPL distributions. Blue = original queries (Q); orange = attacked queries (Q‖S, pool size ≈30 k). The inset on the MS-MARCO panel highlights the single extreme outlier (≈3,000).

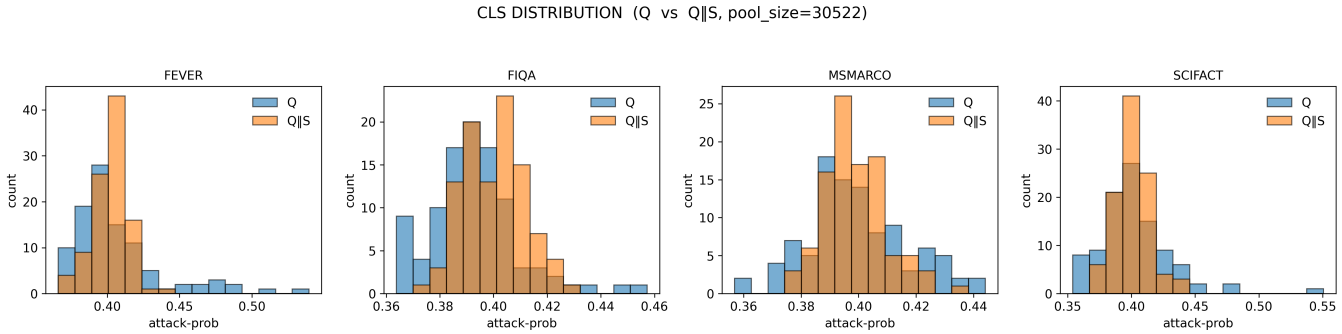


Figure 6: CLS attack-probability distributions for the same 100 query pairs as Fig. 5.

G POOLING TIME ON MLM STRATEGY

Table 14 reports pool construction ("Build") and query optimization ("Query") times. Total attack time is their sum. The results demonstrate the efficiency of constructing a readability-aware candidate pool to improve suffix quality.

Table 14: Pool construction ("Build") and query optimization ("Query") times (seconds). Total attack time is the sum of the two.

Pool Size	FEVER (s)		FiQA (s)		SciFact (s)	
	Build	Query	Build	Query	Build	Query
500	0.0117	44.4184	0.0121	51.0981	0.0106	50.4705
1 000	0.0131	47.2129	0.0128	43.0398	0.0106	48.6922
2 000	0.0132	44.3453	0.0132	46.2533	0.0107	44.8168
5 000	0.0131	45.2823	0.0114	41.4641	0.0117	45.1493
10 000	0.0153	46.0093	0.0113	40.9918	0.0159	42.6680
20 000	0.0155	39.5988	0.0122	46.6468	0.0144	45.9617
30 522	0.0004	45.5934	0.0004	41.4091	0.0004	43.4744

H ADDITIONAL READABILITY ANALYSIS

Table 15 shows the average MLM negative log-likelihood (NLL) of generated suffixes under each pool size, where NLL clearly decreases as the pool shrinks. MLM NLL is often treated as a readability proxy. Table 16 reports Welch's t-test comparing pool_size = 5,000 vs. full (30,522), confirming the reduction is significant across all datasets. Together, these results show the lightweight pooling strategy improves suffix fluency with negligible attack-success impact.

Table 15: Average MLM NLL of generated suffixes under different pool sizes.

Pool size	Fever NLL	FiQA NLL	SciFact NLL
500	7.13 ± 1.26	6.65 ± 1.17	7.19 ± 1.03
1 000	7.19 ± 1.03	7.06 ± 1.13	7.42 ± 0.91
2 000	7.48 ± 1.03	7.48 ± 1.03	7.65 ± 0.94
5 000	7.78 ± 0.86	7.90 ± 1.00	7.78 ± 0.77
10 000	8.17 ± 0.97	8.06 ± 0.79	8.31 ± 0.82
20 000	8.55 ± 0.94	8.57 ± 1.00	8.66 ± 0.86
30 522	8.78 ± 0.85	8.77 ± 0.86	8.84 ± 0.81

Table 16: Welch's t-test comparing MLM NLL for pool_size = 5,000 vs. full pool (30,522).

Dataset	t-value	p-value
Fever	-8.33	1.34×10^{-14}
FiQA	-6.54	5.33×10^{-10}
SciFact	-9.40	1.39×10^{-17}